Energy-aware mapping and scheduling strategies for real-time workflows under reliability constraints

Zhiwei Wu, Li Han, Jing Liu, Yves Robert, Frédéric Vivien

Energy-aware mapping and scheduling strategies for real-time workflows under reliability constraints

Zhiwei Wu, Li Han, Jing Liu, Yves Robert, Frédéric Vivien

RESEARCH REPORT
N° 9469
April 2022
Project-Team Roma

ISSN 0249-6399 ISRN INRIA/RR--9469--FR+ENG

Energy-aware mapping and scheduling strategies for real-time workflows under reliability constraints

Zhiwei Wu *†, Li Han *, Jing Liu *, Yves Robert †‡, Frédéric Vivien †

Project-Team Roma

Research Report n° 9469 — version 3 — initial version April 2022 — revised version November 2022 — 1624 pages

Abstract: This paper focuses on energy minimization for the mapping and scheduling of real-time workflows under reliability constraints. Workflow instances are input periodically to the system. Each instance is composed of several tasks and must complete execution before the arrival of the next instance, and with a prescribed reliability threshold. While the shape of the dependence graph is identical for each instance, task execution times are stochastic and vary from one instance to the next. The reliability threshold is met by executing several replicas for each task. The target platform consists of identical processors equipped with Dynamic Voltage and Frequency Scaling (DVFS) capabilities. A different frequency can be assigned to each task replica to save energy, but it may have negative effect on the deadline and reliability target.
This difficult tri-criteria mapping and scheduling problem (energy, deadline, reliability) has been studied only recently for workflows with arbitrary dependence constraints. We investigate new mapping and scheduling strategies based upon layers in the task graph. These strategies better balance replicas across processors, thereby decreasing the time overlap between different replicas of a given task, and saving energy. We compare these strategies with two state-of-the-art approaches and a reference baseline on a variety of benchmark workflows. Our best heuristics achieve an average energy gain of 60% over the competitors and of 82% over the baseline.

Key-words: real-time workflows, energy-aware, reliability, makespan, mapping, scheduling, tri-criteria optimization

* ECNU, Shanghaï, China
† Univ Lyon, EnsL, UCBL, CNRS, Inria, LIP, F-69342, LYON Cedex 07, France
‡ University of Tennessee Knoxville, USA

RESEARCH CENTRE GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Stratégies de placement et d'ordonnancement sensibles à la consommation énergétique pour graphes de tâches temps-réel avec contraintes de fiabilité

Résumé : Ce travail s'intéresse à la minimisation de la consommation énergétique lors du placement et de l'ordonnancement de graphes de tâches temps-réel soumis à des contraintes de fiabilité. Des instances d'un graphe de tâches sont soumises périodiquement à un système. Chaque instance est composée de plusieurs tâches et son exécution doit être terminée avant l'arrivée de l'instance suivante tout en respectant un niveau de fiabilité donné. Ce niveau de fiabilité est atteint en répliquant un certain nombre de fois chacune des tâches. La plateforme de calcul est constituée de processeurs identiques dont le voltage et la fréquence peuvent être modifiés (système *DVFS*). Chaque réplica de tâche peut se voir attribuer sa propre fréquence.

Ce problème tri-critère de placement et d'ordonnancement (énergie, dates butoir, fiabilité) n'a commencé à être étudié que très récemment avec des dépendances arbitraires. Nous étudions de nouvelles stratégies de placement et d'ordonnancement basées sur une notion de couche du graphe de tâches, et qui équilibrent mieux les réplicas entre les processeurs, ce qui permet de réduire le recouvrement temporel entre les différents réplicas d'une même tâche et, de ce fait, la consommation énergétique. Nous comparons ces stratégies à deux approches concurrentes et à une approche de référence, sur un tout un ensemble de graphes de tâches. Nos meilleures heuristiques obtiennent un gain d'énergie moyen de 60% par rapport aux approches concurrentes et de 82% vis-à-vis de l'approche de référence.

Mots-clés : graphes de tâches temps-réel, consommation énergétique, faibilité, placement, ordonnancement, optimisation multi-critère

1 Introduction

Real-time systems are composed of periodic tasks that are regularly input to a parallel computing platform and must complete execution before their deadlines. In the simpler version of the problem, the tasks are independent. However, many scientific applications from various disciplines are structured as workflows [1]. Informally, a workflow can be seen as the composition of a set of basic operations that have to be performed on a given input data set to produce the expected scientific result. The development of complex middleware with workflow engines [2, 3, 4] has automated workflow management, providing even more appeal to a task-based approach in a variety of scenarios. This leads to scheduling real-time workflows: instances of the same workflow are input periodically to the system and must be completed by their deadline, which is the inter-arrival time of two consecutive instances. All dependence constraints, including communication costs, must be enforced among the tasks, which complicates the derivation of a schedule that meets the deadline. All workflow instances have the same dependence graph but operate on different data sets. Task execution times are stochastic and vary from one instance to the next; typically, it is assumed that they follow some probability distribution with bounded support. Scheduling and mapping decisions are taken based upon the *Worst Case Execution Time (WCET)* of each task, which is the largest value of the distribution support, i.e., the range of non-zero values of the probability density function. Mapping and static schedule tasks based on WCETs is the de-facto standard approach in real-time systems, which is pessimistic but mandatory, to guarantee that deadlines are met in every possible execution. Then at runtime the schedule is updated according to actual execution times of task instances, which obey the distribution and may be significantly smaller than their WCETs.

When scheduling a single workflow instance on a parallel platform, the traditional objective is to minimize the makespan, or total execution time: this is the time at which the last exit task of the workflow completes its execution. With real-time workflows, the objective is similar: one needs to match the deadline, i.e., to guarantee that the makespan does not exceed the deadline bound. Because the problem is already NP-complete with two processors, independent tasks (no dependence) and no communication cost[1], list-scheduling heuristics have been introduced; the main idea is to sort the tasks according to their bottom levels, and to greedily map them onto the first available processor when they become ready; here the bottom level of a task is the length of the longest path from that task to an exit task in the dependence graph of the workflow. The bottom level was originally computed without taking communication costs into account. However, for many scientific workflows, communication costs cannot be neglected, which has motivated the introduction of HEFT [7], a list-scheduling heuristic that computes bottom levels by (pessimistically) including all communication costs in the dependence paths. HEFT is able to deal with heterogeneous platforms by averaging task weights and communication costs, and it has become a reference approach since its inception 20 years ago.

The advent of green computing has motivated to revisit the scheduling of scientific workflows beyond pure time constraints, taking energy consumption into account. Modern processors are equipped with Dynamic Voltage and Frequency Scaling (DVFS) and can run at different frequencies. Running at a lower frequency decreases the dynamic energy that is consumed during the computation but also increases execution time, as well as the consumed static energy (these power components are detailed in Section 2.5). This leads to achieving complicated trade-offs. In addition, it has been widely proved that DVFS has negative effects on transient failure rates. As a result, it is important to take reliability degradation into consideration while managing energy by DVFS. Transient errors [8] are usually manifested as bit-flips and due to radiation,

[1] An easy reduction from 2-PARTITION [5]. See also [6] for more details on the complexity of multiprocessor scheduling.

minimum voltage, or thermal cycling, etc. They cause a task execution to fail but without completely losing the data present in the processor memory, which makes them harder to detect than permanent faults. Several replicas of the same task instance must be executed in order to guarantee a prescribed level of reliability. In fact, reliability has long been the main objective of real-time systems, because these systems deal with critical applications where erroneous or incomplete results could lead to catastrophic scenarios [9]. There are several approaches to enforce a prescribed reliability threshold. Re-execution after a failed execution attempt is one possibility, but it induces additional delays. Hence, the standard approach is to use *replication*: for each task, the execution of several replicas is scheduled, as many as needed to meet the reliability criterion. When DVFS is available, the problem gets more difficult, because one can choose to use, say, either two replicas at a high frequency, or three replicas at a lower one. Moreover, different frequencies can be assigned to different replicas of the same task. To further complicate matters, it is known that the transient error rate increases when scaling down frequency using DVFS (see [10] and the discussion in Section 2.3), so saving energy might lead to more errors. Similarly, the introduction of replicas, while mandatory to guarantee a reliability threshold, is also increasing energy consumption. Clearly, in order to limit this additional consumption, it is always beneficial to kill the other replicas of a task instance after one replica has succeeded; but several replicas may well have been executing concurrently until one of them succeeded, and the energy spent before interrupting them is wasted.

Altogether, the problem of mapping and scheduling real-time workflows on a parallel platform whose processors are equipped with DVFS, has become a challenging tri-criteria optimization problem: energy, deadline (or makespan bound) and reliability are conflicting objectives, and sophisticated algorithms must be designed. To the best of our knowledge, only two recent papers have tackled this problem [11, 12], albeit with additional restrictions (see Section 4.1 for details). We present a novel approach that outperforms all previously published solutions for an extended set of simulation scenarios. Given a deadline constraint and a reliability threshold, we first decide for each task how many replicas to use, and at which frequency to execute them. Next we use a new mapping heuristic based upon a notion of layers in the dependence graph; we map several (ready) tasks simultaneously rather than only the highest priority task; the motivation of using layers is to decrease the overlap between all the replicas of a given task, thereby providing more opportunities to save energy. Finally, we dynamically update the schedule to further reduce replica overlaps without violating deadline and reliability constraints.

The three main contributions of the paper are the following:

1. The formulation of the tri-criteria optimization problem, i.e., minimizing energy cost while ensuring deadline and reliability constraints, and the design of several mapping and scheduling heuristics to solve this problem without any restriction; to the best of our knowledge, this work is the first that can accommodate more than one or two replicas per task, hence that can deal with arbitrary reliability thresholds;

2. A set of heuristics which aggressively attempt to minimize the overlap between the replicas of the same task instance, relying upon the notion of task layers, upon the distinction between primary and secondary replicas, and upon the reclaim of the potential slack in the schedule;

3. An experimental evaluation based on a comprehensive set of simulation scenarios, showing that our best heuristics achieve a significant energy gain compared to the state-of-the-art methods over the whole spectrum of application and platform parameters.

The rest of the paper is organized as follows. Section 2 provides a detailed description of the optimization problem under study. The mapping and scheduling heuristics are described in

Section 3. Section 4 is devoted to a comprehensive experimental comparison of the heuristics. Section 5 presents related work. Finally, Section 6 gives concluding remarks and hints for future work.

Table 1: Key Notations

Notation	Explanation
n, M	number of tasks and of processors
T_i	i-th task of application G
w_i	WCET for task T_i under f_{max}
w_{i,f_j}	WCET for task T_i at frequency f_j
$c_{i,j}$	communication cost from T_i to T_j
$succ(T_i)$	set of successor tasks of T_i in G
$prec(T_i)$	set of tasks preceding T_i in G
$\mathcal{D}(G)$	deadline for application G
$\mathcal{D}(T_i)$	deadline for task T_i
$\mathcal{P}$	set of processors
$rep\_num(T_i)$	actual total number of replicas for T_i
r_i^α	α-th replica of T_i
f_j	the j-th available frequency
$f(r_i^\alpha)$	frequency of replica r_i^α
$\mathcal{R}(G)$	reliability threshold for application G
$\mathcal{R}(T_i)$	reliability threshold for T_i
$R(r_i^\alpha, f_j)$	reliability of r_i^α under f_j
$R_i(f_j)$	reliability of T_i under f_j
k_{i,f_j}	minimal number of replicas required to run T_i with $f(r_i^1) = f_j$ while meeting $\mathcal{R}(T_i)$
$ST(r_i^\alpha)$	start time of r_i^α
$FT(r_i^\alpha)$	completion time of r_i^α
$FT(T_i)$	completion time of T_i
$FT(G)$	completion time of application G
$E(T_i, f_j, k_{i,f_j})$	energy cost for T_i with k_{i,f_j} replicas under f_j

2 Model

The inputs to the optimization problem are a workflow (i.e., a task graph), a set of homogeneous processors, a reliability target and a global deadline. Key notations are summarized in Table 1. Our new scheduling algorithms apply to a very general framework, which we describe in this section. A particular instantiation of the model for the simulations is outlined in Section 4.2.

2.1 Application

The workflow application is modeled as a task graph G, where $G = (V, E)$ is a Directed Acyclic Graph (DAG) composed of a set of n tasks $V = \{T_1, T_2, ..., T_n\}$. An edge from task T_i to task T_j

in set E represents a task dependency. Each task T_i has a weight w_i, which denotes its worst-case execution time (WCET) under the maximum available frequency f_{max}. Task instances (also called *jobs*) typically complete their execution earlier than their estimated WCETs: execution times are assumed to be data-dependent and randomly sampled from some probability distribution whose support is upper bounded by the WCET [13]. Note that without a WCET, i.e., with probability distributions that can take arbitrarily large values, it is not possible to guarantee any deadline. Each edge between two tasks T_i and T_j is also weighted by the size $c_{i,j}$ of the output data produced by T_i that is needed as input to T_j. $prec(T_i)$ and $succ(T_i)$ represent the set of direct predecessors and successors of task T_i respectively. A task T_i can only start its execution when all of its direct predecessors have finished their execution and all intermediate data has been transferred.

2.2 Processors

The platform consists of M homogeneous processors, with the same set F of frequencies ranging from f_{min} to f_{max}. Without loss of generality, we normalize the frequencies to enforce $f_{max} = 1$. The interplay between task execution time and frequency is complicated [14]. To be fully general, we consider arbitrary WCETs that decrease when frequency increases. Specifically, for task T_i at frequency f_j, the WCET is w_{i,f_j}, and we assume that $w_{i,f_j} > w_{i,f_k}$ if $f_j < f_k$. Then the actual execution time of a task instance of task T_i at frequency f_j is drawn from a distribution whose support is upper bounded by w_{i,f_j}. For the simulations in Section 4, we use four different set of frequencies taken from actual microprocessor values, we consider Amdahl speedup functions for the WCETs and we draw actual execution times from Uniform and Truncated Normal distributions.

2.3 Error model

We consider transient errors [8], modeled by an Exponential distribution whose average arrival rate depends upon the frequency used. The interplay between error rate and frequency is complicated too. In fact, the impact of DVFS on the error rate is discussed extensively in the literature. On the one hand, lowering the voltage/frequency is believed to have an adverse effect on the system reliability [15, 16, 17, 18, 19]: this is because decreasing the nominal voltage will increase the vulnerability to longer delays in rippling logic along computational paths, hence lowering the circuit's critical charge (i.e., the minimum charge required to cause an error in the circuit). On the other hand, increasing the voltage beyond the nominal value (overclocking) will also increase the error rate due to overheating [20, 21, 22, 23]. In the general framework, we deal with frequency values in the range $[f_{min}, f_{max}]$, and can safely assume that the lower the frequency, the higher the error rate. In other words, f_{max} is the nominal frequency recommended by the chip manufacturer, and we avoid overclocking to save energy. We let $\lambda(f_j)$ be the instantaneous error rate at frequency f_j and assume that $\lambda(f_j) > \lambda(f_k)$ if $f_j < f_k$. For the simulations in Section 4, we instantiate the values of the $\lambda(f_j)$ from the formula used in [15, 16, 18, 19].

At the end of the execution of a task instance, there is an *acceptance test* [24] to check the occurrence of soft errors induced by the transient errors. If the execution result of a task instance satisfies its predefined acceptance criteria, then it is considered to be successfully executed [25, 26]. As in all relevant literature, it is assumed that acceptance tests are 100% accurate[2], and that the duration of the test is included within the task WCET (see [27] and references therein). The *reliability* of a task instance is the probability of executing it successfully, in the presence of

[2]One could account for the possibility of the acceptance test to fail by lowering the reliability of a replica by some amount, but this goes beyond the standard model.

errors. Multiple replicas of a task instance will be executed to mitigate the impact of transient errors. Note that two replicas of a given task instance will have the same execution time if run at the same frequency, because they operate on the same data. Because we do not know the actual execution time of tasks before they complete, to estimate the reliability of any task instance replica we conservatively assume that its actual execution time is equal to the task WCET. Therefore, a replica of an instance of task T_i running at frequency f_j has reliability $\exp^{-\lambda(f_j)w_{i,f_j}}$.

2.4 Reliability threshold and deadline

There are two constraints for the workflow G in the optimization problem. Firstly, $\mathcal{D}(G)$ represents the global deadline: the execution of exit tasks must be finished before $\mathcal{D}(G)$. Let $FT(T_i)$ denote the finish time of the task T_i. Then the total execution time of the application is $FT(G) = \underset{T_i \in V}{max}\ FT(T_i)$. If $FT(G) \leq \mathcal{D}(G)$, the deadline is met. The calculation of $FT(T_i)$ will be detailed in Section 3.

Secondly, $\mathcal{R}(G)$ denotes the reliability threshold. Given that the reliability of the application is the product of the reliability of each task, we have $\mathcal{R}(G) = \Pi_{i=1}^{n}\mathcal{R}(T_i)$, where $\mathcal{R}(T_i)$ is the reliability threshold of task T_i. In some settings, the set of all $\mathcal{R}(T_i)$ values is given as input, while in some other settings these values are computed from $\mathcal{R}(G)$. Now, given the reliability threshold $\mathcal{R}(T_i)$ for each task T_i, the question is to determine the number of replicas to use, and at which frequency to execute them, so that the threshold $\mathcal{R}(T_i)$ is met. Following the literature again, we assume that the first replica of T_i runs at some frequency f_j to be determined by the mapping and scheduling process, and that all the other replicas run at frequency f_{max}. This simplifying assumption is not needed for our approach but it avoids the exponential cost of searching for all possible combinations. To fix notations, let k_{i,f_j} be the total number of replicas needed for task T_i when the first replica operates at frequency f_j, and let $f(i,j,\alpha)$ be the frequency of replica number α when the first replica runs at f_j, so that $f(i,j,1) = f_j$ and $f(i,j,\alpha) = f_{max}$ for $2 \leq \alpha \leq k_{i,f_j}$. The reliability of replica number α of task T_i is then $R(r_i^\alpha, f_j) = \exp^{-\lambda(f(i,j,\alpha))w_{i,f(i,j,\alpha)}}$.

Since a task instance fails only if all its replicas fail, we achieve a reliability $R_i(f_j)$ for task T_i where

$$R_i(f_j) = 1 - \Pi_{\alpha=1}^{k_{i,j}}(1 - R(r_i^\alpha, f_j)) \tag{1}$$

and we need to enforce that $R_i(f_j) \geq \mathcal{R}(T_i)$. In the scheduling and mapping phase, we let $f(r_i^\alpha)$ denote the frequency actually used for replica number α of task T_i.

2.5 Energy model

The energy cost of a replica of task T_i at frequency f_j is estimated as the power times its worst-case execution time w_{i,f_j}, which is an upper bound for the actual execution. As for the power $P(f_j)$ at frequency f_j, we use

$$P(f_j) = P_{static} + P_{dyn}(f_j) = P_{static} + (P_{indep} + C \times f_j^3)$$

where P_{static} (the static power), P_{indep} (the frequency independent part of dynamic power) and C (the effective switching capacitance) are system-dependent constants. The energy cost $E(T_i, f_j, 1)$ for the first replica of task T_i at f_j is then: $E(T_i, f_j, 1) = P(f_j) \times w_{i,f_j}$. The final

energy cost with k_{i,f_j} replicas is estimated as:

$$E(T_i, f_j, k_{i,f_j}) = E(T_i, f_j, 1) + (1 - R(r_i^1, f_j)) \times \sum_{2}^{k_{i,f_j}} E(T_i, f_{max}, 1) \quad (2)$$

In Equation (2), we assume that there is no overlap between the primary replica (the first replica starting its execution) and the other replicas of a task, while secondary replicas fully overlap. Thus, the estimated energy consumption for a task is the energy consumed by the primary replica plus (in case the primary fails) the energy consumed by all the secondary replicas. Furthermore, each processor always consumes static power when idle (this consumption can be eliminated only by a complete shutdown). Hence, we account for static power whenever the scheduling strategies described below keep some processors idle.

2.6 Optimization Objective

The objective is to determine a set of replicas for each task and their execution frequencies, and to build a static schedule, where the replicas of each task are mapped onto the processors, so that the expected energy consumption is minimized. The constraints are: enforcing all dependencies in the task graph, matching the workflow deadline $\mathcal{D}(G)$, and meeting the reliability threshold $\mathcal{R}(T_i)$ for each task T_i. As already mentioned in the introduction, the expected energy consumption is an average made over all possible execution times randomly drawn from their distributions, and over all failure scenarios (with every component weighted by its probability to occur). An analytical formula is out of reach, and we use Monte-Carlo sampling in the experiments. However, we stress the following two points:

- To guide the design of the heuristics, we use a simplified objective function as described in Equation (2); more precisely, we use WCETs instead of (yet unknown) actual execution times, and we estimate the energy of a task as the sum of the energy of its primary replica plus all its secondary replicas in case the primary replica failed.
- To further complicate matters, the static schedule is dynamically modified on the fly to take into account the actual execution times rather than the WCETs. Also, as soon as one replica of a given task instance completes its execution successfully, all the other replicas of that task instance become redundant and are terminated instantaneously. The scheduler receives a signal from the successful replica and is responsible for both interrupting the other replicas that are currently executing and discarding any additional replica that has not started yet [27].

3 Scheduling heuristics

In this section, we describe the heuristics we introduce to solve the tri-criteria optimization problem. We start by outlining some design guidelines, namely, how to prioritize tasks and compute a deadline for each of them (Section 3.1), how to set frequencies to task replicas (Section 3.2), how to map tasks (Section 3.3) and how to partition them into layers (Section 3.4). Then we introduce our basic static scheduling algorithm in Section 3.5, followed by several energy-saving heuristics using DVFS in Section 3.6. Finally the runtime optimization is described in Section 3.7.

3.1 Task deadlines and ordering

From the workflow deadline $\mathcal{D}(G)$ we derive a deadline $\mathcal{D}(T_i)$ for each task T_i. When deriving task deadlines, we conservatively assume that all communications take place (which is similar to

HEFT assumptions [7]). Consequently, if each task satisfies its deadline, the schedule is valid. We then have:

$$\mathcal{D}(T_i) = \begin{cases} \mathcal{D}(G) \text{ if } succ(T_i) = \emptyset \\ \min\limits_{T_j \in succ(T_i)} \mathcal{D}(T_j) - w_j - c_{i,j} \text{ otherwise} \end{cases} \quad (3)$$

List-based schedules require tasks to be prioritized. We order tasks by non-increasing bottom-levels. The bottom-level $bl(T_i)$ of a task T_i is the length of a critical path starting from T_i if all communications take place:

$$bl(T_i) = w_i + \max_{T_j \in succ(T_i)} (c_{i,j} + bl(T_j)) \quad (4)$$

3.2 Replicas

For saving energy, we use DVFS and decrease the frequency of replicas. We decide that only the frequency of primary replicas may be modified (and, thus, may be lower than f_{max}), while all secondary replicas are executed at f_{max}. Running all secondary replicas at f_{max} decreases the pressure on the schedule. This assumption should not significantly impact the energy consumption because we expect that most secondary replicas will not be executed in actual failure-free execution scenarios.

As we decrease the frequency of a primary replica, we may need to add an additional secondary replica to satisfy the reliability threshold. However, even if we decrease the frequency of the primary replica from f_{max} to f_{min}, at most one additional replica will be required. Indeed, let us assume that we need r replicas to satisfy the reliability threshold for task T_i when all replicas (including the primary) are executed at f_{max}. Then a primary replica at f_{min} and r secondary replicas at f_{max} satisfy the reliability threshold as $\mathcal{R}(T_i) \leq 1 - \Pi_{\alpha=1}^{r}(1 - R(r_i^\alpha, f_{max})) < 1 - (1 - R(r_i^1, f_{min})) \times \Pi_{\alpha=1}^{r}(1 - R(r_i^\alpha, f_{max}))$.

3.3 Task Mapping

To schedule replicas we follow an earliest starting time policy, i.e., we map each replica on a processor that will start its execution at the earliest (breaking ties arbitrarily). In the static scheduling phase, each task should wait for the completion of all the replicas of all of its predecessors before starting its execution. This is because we do not know which replicas of the predecessors will be successful, and we must consider the worst case.

We now describe how to compute the start time ST and the finish time FT of each replica. Let $EST(r, p)$ denote the earliest time at which the replica r could start on processor p, and let $P(r)$ denote the processor that will execute r. By definition of the earliest starting time policy, we have

$$P(r) = \arg\min_{p \in \mathcal{P}} EST(r, p) \quad (5)$$

and

$$ST(r) = EST(r, P(r)), \quad (6)$$

where

$$EST(r_i^\alpha, p) = \max \left\{ \begin{array}{c} ready(p), \\ \max\limits_{\substack{T_j \in prec(T_i) \\ \beta \in [1, rep\_num(T_j)]}} FT(r_j^\beta) + c(r_i^\alpha, r_j^\beta) \end{array} \right\}$$

with

$$c(r_i^\alpha, r_j^\beta) = \begin{cases} 0 \text{ if } P(r_i^\alpha) = P(r_j^\beta) \\ c_{j,i} \text{ otherwise} \end{cases}$$

Here $ready(p)$ denotes the time at which processor p is available, according to the mapping and scheduling decisions previously taken. $rep\_num(T)$ denotes the number of replicas for task T. We will later detail its computation.

We map each replica while ensuring that no two replicas of the same task are assigned on the same processor. This is implicit in Equation (5): the arg min is computed over all processors on which a replica of the same task as r has not already been mapped. Finally, we derive the completion time of each replica as

$$FT(r_i^\alpha) = ST(r_i^\alpha) + w_{i,f(r_i^\alpha)} \tag{7}$$

3.4 Task layers

Tasks are partitioned into *layers* and we will use a layer by layer approach to map and schedule replicas. A task layer is a set of tasks which are mutually independent. Because there is no dependence between any two tasks belonging to the same layer, we can schedule all their replicas in any order. For any task T, $L(T)$ is the integer index denoting its layer. The layer index of exit tasks is equal to 1, and layers are numbered in reverse topological order. Formally:

$$L(T_i) = \begin{cases} 1 + \max\limits_{T_j \in succ(T_i)} L(T_j) \text{ if } succ(T_i) \neq \emptyset, \\ 1 \text{ if } succ(T_i) = \emptyset \end{cases} \tag{8}$$

With the layer by layer approach, we first map one replica of each task in the layer L considered, and we tag them as the *primary* replicas of the tasks in L. All (possibly) remaining replicas for the tasks in L will be *secondary* replicas. We then map a first secondary replica of each task in L, then a second secondary replica, and so on. The aim of this approach is to reduce the overlapping of replicas of the same task.

3.5 The BasicLayer algorithm

Initially, the task graph G is scheduled with the BasicLayer algorithm (Algorithm 1). The aim of BasicLayer is only to build a valid schedule, which will later be optimized. Therefore, BasicLayer does not use DVFS, which means all tasks are mapped with frequency f_{max}. BasicLayer first computes the bottom-level, deadline, and layer index of each task (Lines 2 to 4). Then BasicLayer sorts tasks by their bottom-levels (Line 5) and builds a set of tasks that are in the same layer (Lines 7 to 10). We get the layer identifier from the top task of *TaskList* (Line 7), then move the top task and other tasks with the same layer identifier to *TasksInLayer* (Lines 8 to 10). In line 11, BasicLayer tries to map tasks in this set first using a layer by layer approach (Algorithm 2). If Algorithm 2 fails, BasicLayer maps those tasks using a task by task approach (Algorithm 3 called on line 13). If Algorithm 3 succeeds, BasicLayer moves to the scheduling of the next layer. Otherwise, when even the task-by-task approach failed, the whole schedule is cleared (Line 15), and the whole graph is rescheduled using only the task-by-task approach (Line 16). If this last attempt also fails, we report a failure of the BasicLayer algorithm, meaning that we are not able to find a feasible schedule of G.

Algorithm 2 is an algorithm to map and schedule the replicas of a given layer in a layer-by-layer approach. In Line 2, we first compute the number of replicas for each task in *TasksInLayer* under the assumption that the frequency of each primary replica is f_{max} (recall that secondary replicas are always executed at f_{max}). Then we map the primary replica of each task (Line 3). Note that we use r_i^1 to denote the primary replica of task T_i, and r_i^α to represent any secondary replica (when $\alpha > 1$). After mapping all primary replicas in the layer, we map all secondary replicas in a layer-by-layer manner, which means we map a first secondary replica for each task,

Algorithm 1: The BasicLayer algorithm

Input: A graph $G = (V, E)$, the set F of possible frequencies, the reliability threshold $\mathcal{R}(T_i)$ for each task T_i, the deadline $\mathcal{D}(G)$

```
for T_i ∈ V do
    Compute bl(T_i) using Equation (4)
    Compute D(T_i) using Equation (3)
    Compute L(T_i) using Equation (8)
Sort the tasks in a list TaskList by non increasing bottom-levels
while TaskList ≠ ∅ do
    LayerIdentifier ← L(Head(TaskList))
    TasksInLayer ← {RemoveHead(TaskList)}
    while L(Head(TaskList)) == LayerIdentifier do
        TasksInLayer ← TasksInLayer ∪ {RemoveHead(TaskList)}
    map the tasks in TasksInLayer using Algorithm 2
    if not feasible then
        map the tasks in TasksInLayer using Algorithm 3
        if not feasible then
            clear the whole schedule
            reschedule application G, while only using Algorithm 3 to map each task
            if not feasible then
                return failure to build a schedule
            else
                return schedule
return schedule
```

Algorithm 2: Replica mapping (layer by layer)

Input: The set of tasks *TasksInLayer*, the reliability threshold $\mathcal{R}(T_i)$ and the deadline $\mathcal{D}(T_i)$ for each task T_i in *TasksInLayer*, the schedule built so far

1 **for** $T_i \in$ *TasksInLayer* **do**
2 $\quad rep\_num(T_i) \leftarrow k_{i,f_{max}}$
3 $\quad$ Compute $ST(r_i^1)$ using Equation (6)
4 $max\_rep\_num \leftarrow \max_{T_i \in TasksInLayer} rep\_num(T_i)$
5 **for** $k = 2$ *to* max_rep_num **do**
6 $\quad$ **for** $T_i \in$ *TasksInLayer* **do**
7 $\quad\quad$ **if** $rep\_num(T_i) \leq k$ **then** Compute $ST(r_i^k)$ using Equation (6)
8 **for** $T_i \in$ *TasksInLayer* **do**
9 $\quad FT(T_i) \leftarrow \max_{1 \leq k \leq rep\_num(T_i)} FT(r_i^k)$
10 $\quad$ **if** $FT(T_i) > \mathcal{D}(T_i)$ **then** return *not feasible*
11 return the schedule built so far

Algorithm 3: Replica mapping (task by task)

Input: The set of tasks *TasksInLayer*, the reliability threshold $\mathcal{R}(T_i)$ and the deadline $\mathcal{D}(T_i)$ for each task T_i in *TasksInLayer*, the schedule built so far

1 **for** $T_i \in$ *TasksInLayer* **do**
2 $\quad rep\_num(T_i) \leftarrow k_{i,f_{max}}$
3 $\quad$ **for** $1 \leq k \leq rep\_num(T_i)$ **do** Compute $ST(r_i^k)$ using Equation (6)
4 $\quad FT(T_i) \leftarrow \max_{1 \leq k \leq rep\_num(T_i)} FT(r_i^k)$
5 $\quad$ **if** $FT(T_i) > \mathcal{D}(T_i)$ **then** return *not feasible*
6 return the schedule built so far

then a second secondary replica, and so on (Lines 5 to 7). Finally, we compute the completion time of each task (Line 9) and check whether any task exceeds its deadline (Line 10). If this is the case, we return *not feasible*, otherwise we return the partial schedule built so far.

For the task-by-task mapping of replicas (Algorithm 3), we iteratively schedule each task in *TasksInLayer* according to the task priority order. For any given task, we first compute its number of replicas with the same assumption as for Algorithm 2 (Line 2). Then we map its primary replica and each of its secondary replicas (Line 3). Finally, we compute its completion time (Line 4), and check whether it exceeds its deadline (Line 5). If this is the case, we return *not feasible*, otherwise, we return the partial schedule built so far.

3.6 Energy-saving heuristics

Recall that BasicLayer does not use DVFS to save energy. Therefore, after successfully building an initial solution where all replicas are run at f_{max}, we use some heuristics to lower the frequency of primary replicas in order to save energy. However, as already stated in Section 3.2, when the frequency of a primary replica decreases, we may need to add one additional replica. Adding some extra replicas may prevent the workflow to be scheduled under the deadline and reliability constraints. Hence, it is very important to choose which tasks can have an additional replica, and

which cannot. We propose several heuristics to choose which tasks will be granted an additional replica while ensuring the application can still be scheduled:

- **MinRepNumber**: No task is allowed to have an additional replica (this is the baseline).
- **TaskSize**: In each layer, tasks are sorted by non-increasing WCETs. This is because tasks with longer execution time consume more energy. Higher priority will give more freedom to those tasks to decrease their frequencies as low as possible. Following this order, we consider the tasks one by one. Each task can have an additional replica provided that a valid schedule can still be built.
- **LayerSize**: We define the WCET of a layer as the sum of the WCETs of its constituting tasks. LayerSize sorts layers by non-increasing WCETs and processes them in this order. When processing a layer L, it checks whether all tasks in L can be granted an additional replica. If so, one replica is added for each task in L, otherwise none of the tasks in L is allowed an additional replica.
- **TopoLayerSize**: This heuristic works as LayerSize except that after processing a layer L, it goes directly to processing the next layer in the sorted list whose layer index is larger than that of L. In other words, LayerSize does not process all layers but only a topologically ordered subset of them.

The next heuristic proceeds differently: rather than starting with BasicLayer, it tries to use DVFS from the start in an aggressive way:

- **OptFrequency**: For each layer, OptFrequency maps and schedules each primary replica using the frequency minimizing the energy consumption as expressed by Equation (2). If the global deadline is not satisfied, the solution of BasicLayer is used as is.

After deciding which tasks can have an additional replica, we use Algorithm 4 to adjust the frequency of primary replicas and the scheduling of all replicas. The aim of the latter type of modification is to reduce the overlap between the execution of a primary replica and that of the secondary replicas of the same task. In the BasicLayer algorithm, we map each replica as early as possible. Hence, it may contain some slack with respect to deadlines at the end of the schedule. We use this slack to postpone the execution of some replicas or to adjust the frequency of some primary replicas. Firstly, we consider the tasks in an order reverse from that of BasicLayer (Line 1), which means considering tasks starting from the end of the schedule. Then, we build at Line 5 the layer containing the first task, exactly as we did in Lines 7–10 of Algorithm 1. We delay the start of each secondary replica as much as possible. For a secondary replica (Lines 7 to 14), we compute its latest possible completion time according to its deadline and to the earliest start time of its successors (Line 8). We then optimize the schedule to minimize the overlap between the different secondary replicas of the same task (Line 9), while forbidding any of them to start earlier than in the original schedule. Then we delay the start times of secondary replicas according to the new computed completion times (Lines 11–14). To be specific, we compare the new completion times with original ones (Line 11): set completion time to the larger value (Line 12), and if this value is the new computed one, then set signal *has_changed* (Line 13) and delay the start time of secondary replicas (Line 14). For a primary replica, we also compute its latest possible completion time (Line 16). Then we try to reduce its frequency as much as possible as allowed by the available number of replicas (Line 18). Then we delay its start time according to the new computed completion time and frequency (Lines 19 and 20). Note that some tasks have already been granted an additional replica by some of the heuristics described

Algorithm 4: Schedule optimization

Input: The base schedule S, the Graph G = (V,E)

1 Sort the tasks in a list *TaskList* by non-decreasing bottom-levels
2 $has\_changed \leftarrow true$
3 **while** $has\_changed$ **do**
4 $has\_changed \leftarrow false$
5 $TasksInLayer \leftarrow ExtractFirstLayer(TaskList)$
6 **for** $T_i \in TasksInLayer$ **do**
7 **for** $2 \leq k \leq rep\_num(T_i)$ **do**
8 $newFT(r_i^k) \leftarrow \min\left\{\mathcal{D}(T_i), \min_{T_j \in succ(T_i)} ST(T_j) - c_{i,j}\right\}$
9 Minimize the overlap between replicas of T_i
10 **for** $2 \leq k \leq rep\_num(T_i)$ **do**
11 **if** $newFT(r_i^k) > FT(r_i^k)$ **then**
12 $FT(r_i^k) \leftarrow newFT(r_i^k)$
13 $has\_changed \leftarrow true$
14 $ST(r_i^k) \leftarrow FT(r_i^k) - w_{i,f_{max}}$
15 **for** $T_i \in TasksInLayer$ **do**
16 $newFT(r_i^1) \leftarrow \min\left\{\mathcal{D}(T_i), \min_{2 \leq k \leq rep\_num(T_i)} ST(r_i^k)\right\}$
17 **if** $FT(r_i^1) < newFT(r_i^1)$ **then**
18 decrease r_i^1 frequency as much as possible while satisfying $FT(r_i^1) \leq newFT(r_i^1)$ and $k_{i,f(r_i^1)} \leq rep\_num(T_i)$
19 $FT(r_i^1) \leftarrow newFT(r_i^1)$
20 $ST(r_i^1) \leftarrow FT(r_i^1) - w_{i,f(r_i^1)}$
21 $has\_changed \leftarrow true$
22 **if** $rep\_num(T_i) > k_{i,f(r_i^1)}$ **then** delete one secondary replica of T_i

above. Hence, these tasks can decrease the frequency of their primary replica to f_{min} (the slack permitting). If a task T_i has received an additional replica, but the frequency of its primary replica frequency does not require one, we delete the additional replica (Line 22). We repeat the whole process as long as we can improve the schedule.

The time complexity of Algorithm 1 is $O(n^2)$, while that of Algorithm 4 is $O(n^2 \times |\mathcal{P}|)$ –recall that n is the number of tasks, and $|\mathcal{P}|$ is the number of processors. This low complexity shows that our approach has a great potential for dealing with a large number of tasks (see also reports on actual execution times in Section 4.4).

3.7 Runtime optimization

In all executions of the heuristics, the successful execution of a replica leads to the immediate cancellation of all other replicas of the same task instance. This is a crucial source of energy saving. Furthermore, the actual execution times of replicas are usually shorter than their WCETs. Therefore, when one replica completes, we have some additional slack to adjust the schedule of other (unfinished) replicas, in order to save energy.

In the actual execution, when one replica completes, we adjust the scheduling of the first

unfinished replica of each processor. For each unfinished replica, if there exists another replica of the same task instance that has successfully completed, it is immediately canceled. Otherwise, this replica is processed as early as possible, while satisfying precedence constraints and while not increasing its overlap with other unfinished replicas of the same task instance.

4 Performance evaluation

In this section, we present simulation results to evaluate the performance of our strategies compared to two state-of-the-art competitor approaches, OEA and MILP. In Section 4.1, we introduce both competitors and explain how we have extended them; we also present a baseline approach, QFEC, which we use for reference. In Section 4.2, we describe the parameters and settings used for the experimental campaign and in Section 4.3 we discuss the limitations of MILP. Finally, we present the results in Section 4.4.

4.1 Comparing Methods

We compare our five heuristics (MinRepNumber, TaskSize, LayerSize, TopoLayerSize, and OptFrequency) with one baseline reference, QFEC [28], and two state-of-the-art approaches, OEA [11] and MILP [12].

Baseline QFEC Some related heuristics have been designed in the book [28]. Chapter 2 of [28] introduces heuristics for the bi-criteria problem (energy, makespan) while chapter 3 deals with another bi-criteria problem (reliability, makespan). Both chapters target heterogeneous platforms. In this work, we have extended the heuristic QFEC of chapter 3 in [28] to address the tri-criteria problem and used it as a baseline for performance comparison. In our extension of QFEC, all replicas are assigned the highest frequency, and their number is computed so as to match the reliability threshold. The scheduling and mapping obey the HEFT priority and assignment rules [7]. This QFEC extension is used as a reference to report the energy savings of the other heuristics.

OEA The OEA heuristic [11] minimizes the energy consumption while satisfying the reliability requirement, but only considers low reliability thresholds for which additional replicas are not needed: using only one instance of each task is assumed to be enough to match the reliability threshold. The general OEA heuristic targets heterogeneous platforms; for homogeneous platforms, the same frequency is assigned to all tasks. A major difference is that the mapping of tasks to processors is assumed to be given in OEA; computing a mapping is achieved by the ODS heuristic [11], which differs from HEFT in that priority is given to tasks with high out-degree. The intuition is that the scheduling of a task is more urgent if it has many successors. We have extended OEA with replicas to match arbitrary reliability thresholds as follows: all primary replicas are assigned the same frequency f, and all secondary replicas are assigned f_{max}; we iteratively find the smallest primary replica frequency f (and the corresponding replica number) that can meet the reliability and deadline constraints using HEFT to allocate each replica: this leads to the Heft-OEA heuristic. We have also designed a similar extension to deal with replicas, but keeping the original ODS scheduling instead of using HEFT: this leads to the ODS-OEA heuristic. After these phases, we use Algorithm 4 to decrease the overlap between primary and secondary replicas. Altogether, we have designed two optimized extended version of OEA, one with traditional HEFT scheduling (Heft-OEA) and one with the ODS priority scheduling (ODS-OEA).

MILP The MILP heuristic [12] presents a Mixed Integer Linear Program (MILP) to deal with the same tri-criteria problem as in this paper (in the variant which considers task-level DVFS giving rise to the RAFTM-TL linear program; the other variants being less general). Hence, this is the closest related work to our approach. However, MILP does not take communication costs into account and is limited by design to have at most two replicas per task; hence, it cannot match high reliability thresholds that require three or more replicas.

We first comment on the objective function of MILP. Then we show how to modify MILP to obtain a new version, called ExtMILP, which does not have its limitations. The objective of MILP is to minimize the energy consumption. The formula used to estimate the energy consumption (Equation (10) of [12]) assumes that if a task has two replicas, then both are fully executed. This is a conservative, and thus pessimistic, assumption: if the primary replica completes successfully, and the two replicas do not overlap, the secondary replica does not need to be launched and its energy consumption is then null. Our own objective function in Equation (2) accounts for the possibility that a primary replica completes successfully, and is optimistic since it assumes that the secondary replicas of a task do not overlap with the primary replica. However, we fix the frequency of secondary replicas at f_{max} (to simplify the problem) while MILP is free to chose any frequency.

To take communications into account in MILP, we need to modify Equation (8) of [12] which enforces task dependencies. To the right-hand side of this equation we add the term

$$c_{i,j}(1 - w_{ij} - w_{ji}).$$

where w_{ij} is a binary variable equal to 1 if and only if the i-th task is executed before the j-th task and on the same processor. Indeed, if tasks i and j are both mapped on the same processor no communication cost should be paid and exactly one of the two binary variables w_{ij} and w_{ji} is equal to 1 while the other one is 0. Otherwise $w_{ij} = w_{ji} = 0$ and the communication cost must be paid.

We now show how to extend Equation (2) of [12] to enable a number N of secondary replicas for each task. Let σ_{kN+i} be the binary variable indicating whether there is a need for $k+1$ replicas for task i (hence, k secondary replicas). We have the constraint: $\sigma_{(k+1)N+i} \leq \sigma_{kN+i}$, because there is no need for a $(k+1)$-th secondary replica if there is no need for a k-th one. Let R_i^k be the reliability obtained using the primary replica and the first k secondary replicas for task i. There is a need for a $(k+1)$-th secondary replica only if k secondary replicas do not provide the required level of reliability. Hence, $R_i^{th} > R_i^k$ implies $\sigma_{(k+1)N+i} = 1$. Therefore, the equation generalizing Equation (2) of [12] is:

$$\delta_i - (1 + \delta_i)\sigma_{(k+1)N+i} \leq R_i^k - R_i^{th} \leq 1 - \sigma_{(k+1)N+i}.$$

These inequalities are not linear because of the definition of R_i^k:

$$R_i^k = (1 - \sigma_{kN+i})R_i^{k-1} + \sigma_{kN+i}\left(1 - \prod_{j=0}^{k}(1 - r_{jN+i})\right)$$

where r_{jN+i} is the reliability of the j-th replica, i.e., using the notations of [12]:

$$r_{jN+i} = \sum_{l \in \mathcal{L}} s_{jN+i} l e^{-\phi_i(f_l)}.$$

Hence, the formula of R_i^k is a polynomial of degree $k+1$ in the binary variables s_{jN+i}, which can easily be linearized using classical methods [29].

Table 2: Comparison of the capabilities of the different heuristics: maximum allowed number of replicas per task; whether DVFS is used or all replicas are executed at f_{max}; whether communication costs are taken into account.

	MILP	ExtMILP	OEA	Heft-OEA and ODS-OEA	QFEC	Our methods
Number of replicas	≤ 2	arbitrary	1	arbitrary	arbitrary	arbitrary
DVFS	Yes	Yes	Yes	Yes	No	Yes
Communication cost	No	Yes	Yes	Yes	Yes	Yes

Comparison fairness We point out that all energy-aware heuristics discussed in this section (Heft-OEA, ODS-OEA, ExtMILP) were originally designed to solve problem instances with low reliability thresholds: there is a single replica per task in OEA, and only two replicas per task in MILP. The heuristics introduced in this work are the first ones that can solve arbitrary instances of the tri-criteria problem. We have extended OEA to deal with replicas and MILP to use an arbitrary number of replicas and to take communication costs into account. We have optimized their runtime executions in the same way as our own heuristics. Table 2 details the different capabilities of the methods. We will see in Section 4.4 that solving ExtMILP with only two replicas is already very slow. Therefore we report results for ExtMILP for low reliability scenarios requiring, most of the times, only two replicas.

4.2 Experimental methodology

We validate our methods on a comprehensive set of parameters that we detail below. In the simulations, we set $M = 8$ cores as default. Interested readers could run experiments with different settings using our discrete simulator which is publicly available online [30].

Fault rates. Let λ_0 denote the fault rate at frequency f_{max}. Then the fault rate at frequency f_i is given by the formula used in [15, 16, 18, 19]: $\lambda(f_i) = \lambda_0 \times \exp^{\frac{d(1-f_i)}{1-f_{min}}}$, where d is the sensitivity factor; d is a measure of how quickly the transient fault rate increases when the system supply voltage and frequency are scaled. Following [27], we assume that the transient error arrival rate at f_{max} is $\lambda_0 = 10^{-6}$ and the system sensitivity factor is $d = 4$.

Frequency sets. We have 4 sets of frequencies. Due to space limitation, the plots in Section 4.4 are for a 5-level frequency set which is taken from a real microprocessor [31] and whose (normalized) values are $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$. We also validate our methods on two other real frequency sets, $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$ and $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$ [32, 14], and one synthetic set ranging from 0.1 to 1 by step of 0.1: $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$. See [33] for the complete set of experiments.

WCETs and task execution times. To be close to real application cases, we let WCETs obey a randomized speedup function similar to Amdahl's law: specifically for task T_i at frequency f_j, the WCET is $w_{i,f_j} = s_i w_i + (1 - s_i)\frac{w_i}{f_j}$, where the sequential fraction s_i is drawn uniformly and randomly in the interval $[0.1, 0.3]$. Actual task execution times for each task may have different probability distributions. In the experiments, we consider that the actual execution time is between a best case (BC) and a worst case (WC) execution time and is determined by a normal distribution or by a uniform distribution. The mean and the standard deviation of the normal distribution are set to $(BC + WC)/2$ and $(BC - WC)/6$, respectively, as in [27]. Moreover, we varied the BC/WC ratio from 0.1 to 0.9 by step of 0.1 to model different workload variability settings.

Power. The static power and the frequency-independent part of the dynamic power are set to 5% and 15% respectively, of the maximum frequency-dependent power consumption with $C = 1$.

Workflows. We consider 12 workflows, including 9 workflows (with 300 tasks) from real-world applications, namely Genome, BLAST, BWA, Cycles, Epigenomics, Montage, Seismology, SoyKB and SRASearch generated by the Pegasus Workflow Generator (PWG) [34, 35, 36], as well as the 3 most classical factorization algorithms of a $k \times k$ tiled matrix (LU, QR and Cholesky) [37]. For each factorization, we perform experiments with $k = 15$ with up to 1240 tasks. The number of vertices in the DAG depends on k as follows: the Cholesky DAG has $\frac{1}{3}k^3 + O(k^2)$ tasks, while the LU and QR DAGs have $\frac{2}{3}k^3 + O(k^2)$ tasks. The task weights (WCETs) and file sizes of Pegasus workflows are generated by PWG, and those for the numerical kernels correspond to the number of floating-point operations and communication volumes according to [38]. The actual execution time (at f_{max}) of each task instance of a task T_i is randomly generated from a uniform distribution or a normal distribution see details in Section 4.4.

Reliability threshold. We first compute the reliability of the whole workflow when there is a single replica per task running at f_{max}: $R(G) = \prod_{i=1}^{n} \mathcal{R}_i(f_{max})$. This will give us a corresponding probability of failure $F = 1 - R(G)$. Then we consider three reliability thresholds $\mathcal{R}(G)$ chosen as $1 - F$, $1 - \frac{F}{10}$ and $1 - \frac{F}{100}$. Thus, the reliability thresholds vary from one workflow to another. To compute the reliability threshold $\mathcal{R}(T_i)$ for each task T_i, we let $\mathcal{R}(T_i) = \sqrt[n]{\mathcal{R}(G)}$, where n is the number of tasks in the workflow.

Communication-to-Computation Ratio (CCR). An important factor that influences the performance of scheduling strategies is the data-intensiveness of the application. We define the Communication-to-Computation Ratio (CCR) as the time τ needed to store all the files handled by a workflow (input, output, and intermediate files) divided by the time T needed to perform all the computations of that workflow on a single processor. Here τ is simply the sum S of all communication sizes (in bytes) divided by the bandwidth b, and T is the sum of all WCETs (in seconds). In other words, the CCR is

$$c = \frac{S}{b}\frac{1}{T} \tag{9}$$

We consider 3 CCR values namely $c = 1$, $c = 0.1$ and $c = 0.01$. Given the value c of the CCR, we generate the value of the communication time $c_{i,j}$ as the size $s_{i,j}$ of the file communicated from T_i to T_j divided by the bandwidth $b = \frac{S}{cT}$ computed from Equation (9). When not specified otherwise, we report performance for the case $c = 1$. Again, the complete set of experimental results is available at [33].

Deadlines. We set five deadlines for each workflow: tightest, tight, medium, relatively loose, and loose. To calculate the tightest deadline d_1, we take the makespan when scheduling the workflow with HEFT with a single replica per task running at f_{max}. Then the loose deadline is $d_5 = 10 \times d_1$ and the other deadlines are at $\frac{1}{4}$, $\frac{1}{2}$, and $\frac{3}{4}$ of the interval (d_1, d_5) respectively.

4.3 Limitations of MILP

The time complexity of the MILP method is quite large: in [12], the authors of MILP report execution times for MILP ranging from 3 seconds to more than 300 seconds, with a median at 150 seconds, for a task graph only containing a total of 20 replicas (hence, for a workflow containing at most 10 tasks when each task has a second replica). They do not report execution times for workflows containing a total of 30 tasks and replicas (the largest workflows considered in [12]). We have extended MILP to ExtMILP as detailed in Section 4.1. In the simulations involving ExtMILP, there are three differences from the general setting: (i) we only consider the 8 workflows from the Pegasus suite that could generate graphs with 10 to 70 tasks (for instance, the Montage workflow has a minimal task number of 133 and is not considered); (ii) we consider three relatively low reliability thresholds 0.9, 0.93, and 0.96, that is, reliability thresholds which are chosen sufficiently large to have a good chance to be achievable with only two replicas per

Table 3: Type of the solution found by MILP with a reliability threshold of 0.96.

		deadline				
	number of tasks	tightest	tight	medium	relatively loose	loose
BLAST	20	Suboptimal	Suboptimal	Optimal	Optimal	Optimal
	40	No Result	No Result	Suboptimal	Suboptimal	Suboptimal
	60	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
BWA	20	Suboptimal	Suboptimal	Optimal	Optimal	Optimal
	40	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	60	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
Cycles	20	Optimal	Optimal	Optimal	Optimal	Optimal
	40	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	60	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
Epigenomics	20	Optimal	Optimal	Optimal	Optimal	Optimal
	40	Optimal	Optimal	Optimal	Optimal	Optimal
	60	No Result	Suboptimal	No Result	Suboptimal	Suboptimal
Genome	20	Suboptimal	Suboptimal	Suboptimal	Optimal	Optimal
	40	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	60	No Result	Suboptimal	No Result	No Result	Suboptimal
Seismology	20	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Optimal
	40	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	60	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Suboptimal
SoyKB	20	Optimal	Optimal	Optimal	Optimal	Optimal
	40	Optimal	Optimal	Optimal	Optimal	Optimal
	60 (3 replicas per task)	No Result	No Result	No Result	No Result	No Result
SRASearch	20	Suboptimal	Optimal	Optimal	Optimal	Optimal
	40	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	60	No Result	No Result	No Result	No Result	Suboptimal

task; and (iii) we set a timeout of 600 seconds for ExtMILP (if, after 600 seconds ExtMILP has not been able to find an optimal solution, it will report the best feasible solution found so far, if it was able to find at least one). By construction of ExtMILP, the maximum number of replica per task is hard-coded in the linear program. However, the larger the number of replicas, the larger the running time of ExtMILP. Therefore, in the evaluation we first run ExtMILP with a maximum of two replicas per task. If ExtMILP answers that no solution exists, it is then run with a maximum of three replicas per task, and so on, until it finds a solution or the allowed time is exceeded.

4.4 Results

We designed a discrete event simulator which is publicly available online [30] with all experimental data. Interested readers could run experiments with different settings, e.g., different numbers of cores, different workflow sizes, etc. Due to space limitations we only present a subset of our results in the core of the report: all results are available in the appendix. The performance reported on Figure 1 and on Figures 4 through 7 is based on the first frequency set and a generation of actual execution times that uses a uniform distribution and a BC/WC ratio equal to 0.8. On Figure 1 and on Figures 4 through 7, the x-axis is the deadline of the application (from tightest to loosest), and the y-axis displays the ratio of the energy cost of the studied heuristic with respect to the energy cost of QFEC; hence, the lower the better. All energy costs are evaluated using the actual execution times of tasks (and not their WCETs).

On graphs which display box plots, each box ranges from the 25th percentile to the 75th, and the horizontal line inside a box marks the average performance. The whiskers explicit the whole range of achieved performance from the best to the worst one.

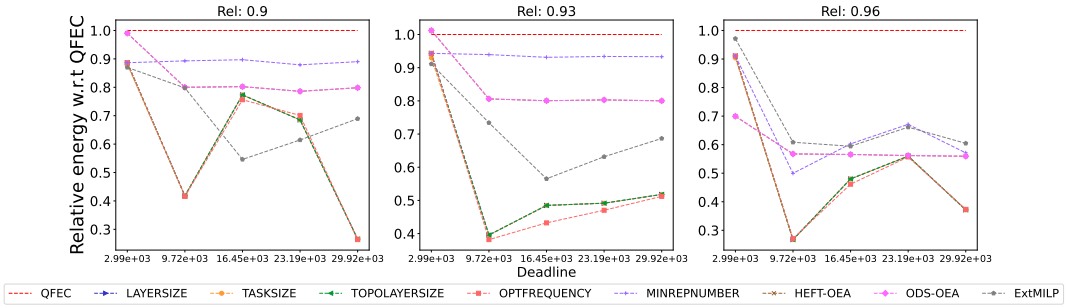

Figure 1: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks) under the frequency set f_1, a CCR of 1.0, and when actual execution times are generated with a uniform distribution and a BC/WC ratio of 0.8.

ExtMILP experiments In Table 3 we report the type of the solution found by ExtMILP: "Optimal" means that ExtMILP found an optimal solution; "Suboptimal" means that ExtMILP only found a feasible solution within the allocated 600 seconds, but not an optimal one; "No Result" means that ExtMILP was not able to find a single feasible solution in 600 seconds. Table 3 shows that as the number of tasks increases, ExtMILP requires more time to obtain the optimal solution. Hence, the quality of the solution found by ExtMILP decreases as the number of tasks increases, moving from an optimal solution, to a feasible solution (without any guarantee on its quality), to no solution at all. Therefore, in our simulations including ExtMILP we set the task size of the considered workflows from 10 to 70.

Figure 1 displays the relative performance of ExtMILP and of the other heuristics for the SRASearch workflow with 20 tasks. This figure shows that when the deadline becomes larger (and thus looser), our methods (e.g., TaskSize, LayerSize) perform (generally) better than ExtMILP. Indeed, with larger deadlines, there is more scheduling freedom and it becomes possible to decrease the overlap between the replicas of the same job. Such a strategy leads to a significant reduction of the energy consumption and explains why our heuristics outperform ExtMILP. Furthermore, when the reliability increases our heuristics outperform ExtMILP for tighter deadlines.

One can check in Figure 2 that this conclusion about the impact of the tightness of deadlines is general. For the tightest deadlines, ExtMILP achieves a performance similar to those of the best competitors. However, for looser deadlines our methods are significantly outperforming it.

Figure 3 aggregates the statistics over the performance of all heuristics over all parameter settings as a function of the considered workflow. ExtMILP achieves good to very good performance for three of these workflows (Genome, Cycles and Seismology), passable performance for one of them (BWA), and very bad performance for the remaining 4 workflows (BLAST, Epigenomics, SoyKB, SRASearch). This confirms that even for small workflows ExtMILP is not the heuristic of choice.

In fact, ExtMILP achieves the best performance only when the deadline is very tight or the reliability threshold is very low. This may surprise the reader as ExtMILP is designed to obtain an *optimal* solution. This is due to the choice of the objective function of ExtMILP. As we explained in Section 4.1, ExtMILP uses a pessimistic approximation to evaluate the energy consumption of a solution while our heuristics work with an optimistic one. The fundamental problem is the complexity of the evaluation of the energy consumption of a solution, mainly because of the potential overlapping of the different replicas of a single task.

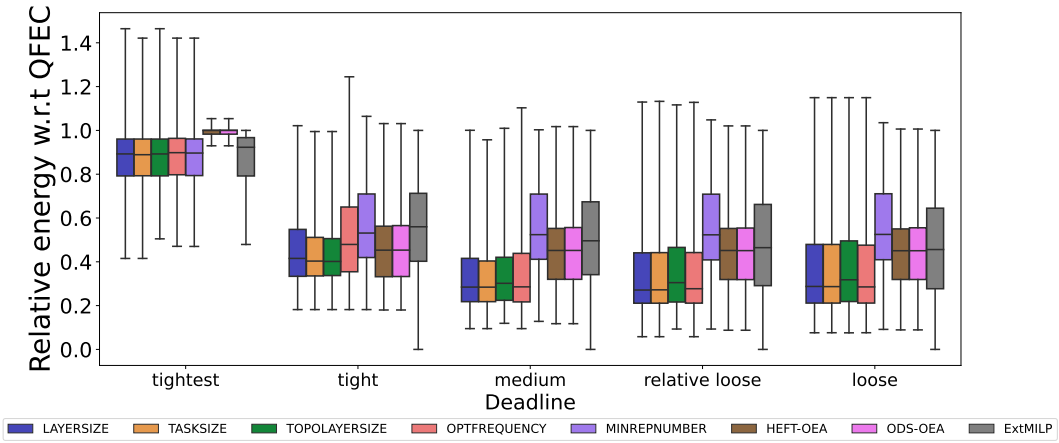

Figure 2: Comparing the performance of ExtMILP and of the other heuristics over all parameter settings as a function of the deadline tightness.

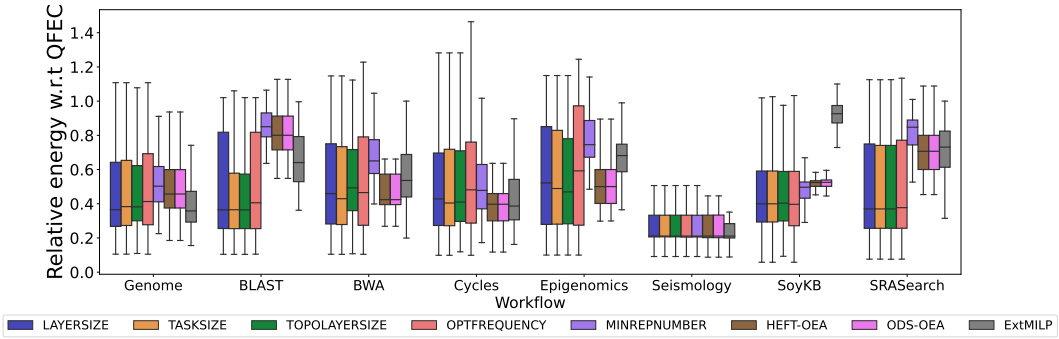

Figure 3: Comparing the performance of ExtMILP and of the other heuristics over all parameter settings as a function of the workflow.

General experiments: study of particular instances We start by considering a set of well-chosen experiments (Figures 4 to 7) to assess some of the features of the proposed solutions. We move to the whole set of experiments right after this.

Figure 4 reports performance when the reliability is low enough so that each task only requires a single replica. HEFT-OEA and ODS-OEA, which were designed for this peculiar case, are nonetheless outperformed by our heuristics, and especially by LAYERSIZE and TASKSIZE (in this graph HEFT-OEA is hidden behind ODS-OEA most of the time). Hence, while our new heuristics were specifically designed to work with any number of replicas, they are already very efficient with a single replica per task.

A major feature of our heuristics is their layer-by-layer approach. We assess the pertinence of this design choice with Figure 5. There, we compare the performance of heuristic TASKSIZE in its layer-by-layer approach and in a task-by-task version. This figure shows that a layer-by-layer approach reduces energy consumption when the deadline is tight. When the deadline is loose, there is enough scheduling freedom for even a task-by-task approach to avoid the overlapping of primary and secondary replicas. Hence, a layer-by-layer approach cannot lead to additional gains in such a case.

At runtime, we implement the optimization described in Section 3.7. We assess with Figure 6 the impact of this optimization. Once again, when the deadline is tight, this type of optimization can lead to additional gains. The runtime optimization never leads to worse performance, whatever the workflow and the parameter settings [33]; therefore, it should always be used.

Figure 7 reports the performance of the heuristics when the number of cores M varies. We observe on this figure that the number of cores has no significant impact on the relative performance of our heuristics with respect to the competitors, HEFT-OEA and ODS-OEA. Again, interested readers can run additional experiments with various settings using [30].

General experiments: synthesis over all parameter settings Figure 8 presents a synthesis of the performance achieved by each heuristic for each workflow, over the whole set of experiments (for the different frequency sets, reliability thresholds, deadlines, communication-to-computation ratio, maximum BC/WC ratio and the two different distributions used to generate the actual execution times).

It is obvious that HEFT-OEA and ODS-OEA achieve rather poor performance overall and for each single workflow, even though they already lead to significant gains with respect to QFEC. This proves the benefit of using DVFS for energy minimization, even if lowering the processor voltage leads to an increase in the fault rate.

MINREPNUMBER achieves some rather good performance for some of the workflows (typically the three factorization algorithms: LU, QR and Cholesky) and poor performance for others (BLAST and SRASearch being the worst workflows for MINREPNUMBER).

OPTFREQUENCY achieves better performance than MINREPNUMBER except for the three factorization algorithms. It even achieves the best performance overall for SoyKB but also has difficulties with the BLAST and SRASearch workflows.

Among the three remaining heuristics, TASKSIZE achieves slightly lower average performance than TASKSIZE and TOPOLAYERSIZE for workflows Cycles, Epigenomics, Montage, SRASearch and, less significantly, for BWA, QR and Cholesky. Its average is slightly outperformed for Genome and LU. Even if it achieves worse performance than OPTFREQUENCY for SoyKB, the difference of average performance is quite small. On the other hand, TASKSIZE achieves significantly better average performance than any other heuristic for SRASearch: its 75-th percentile is below the average of any other heuristic. Altogether, TASKSIZE achieves robust performance overall with very few weak points, and appears as a heuristic of choice. A peculiar case is that of workflow BWA: the 25th percentile and average performance of TASKSIZE is better than any

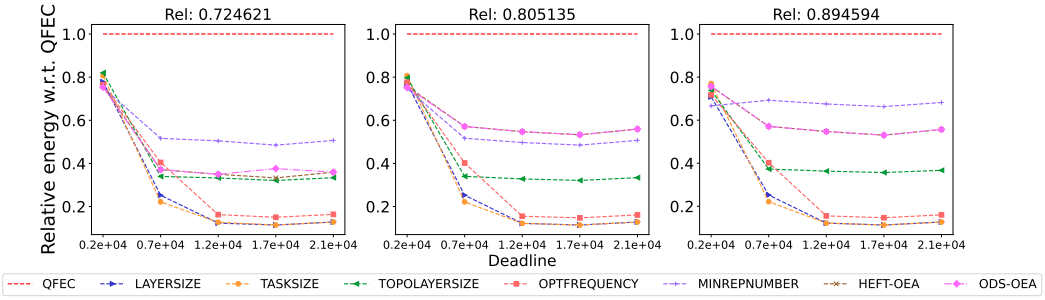

Figure 4: Assessing performance with the Montage workflow when the reliability threshold is very low (a single replica per task is needed) under the frequency set f_1, a CCR of 1.0, and when actual execution times are generated with a uniform distribution and a BC/WC ratio of 0.8.

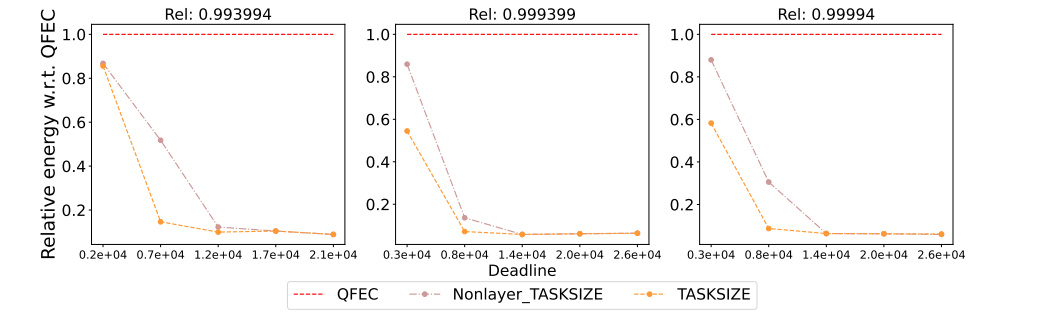

Figure 5: Assessing the impact of the layer-by-layer approach on the Montage workflow under the frequency set f_1, a CCR of 1.0, and when actual execution times are generated with a uniform distribution and a BC/WC ratio of 0.8.

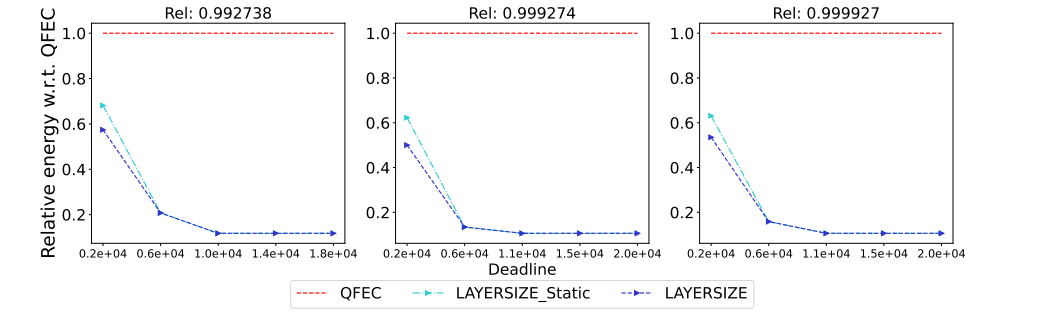

Figure 6: Assessing the impact of the runtime optimization on the Cycles workflow under the frequency set f_1, a CCR of 1.0, and when actual execution times are generated with a uniform distribution and a BC/WC ratio of 0.8.

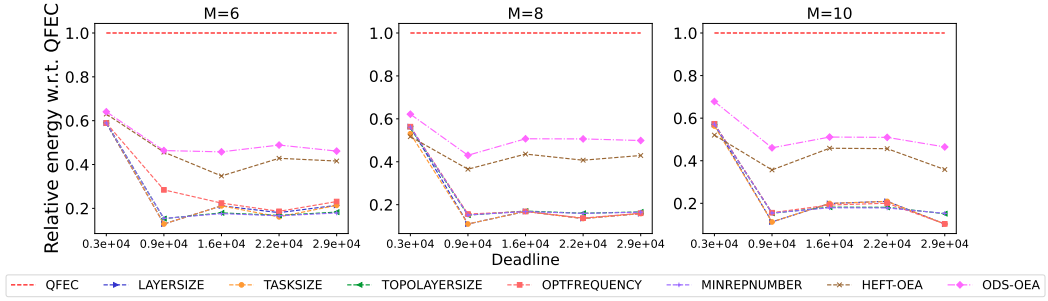

Figure 7: Assessing the impact of the number of processors M on the Montage workflow under the frequency set f_1, a CCR of 1.0, a reliability threshold of 0.999399, and when actual execution times are generated with a uniform distribution and a BC/WC ratio of 0.8.

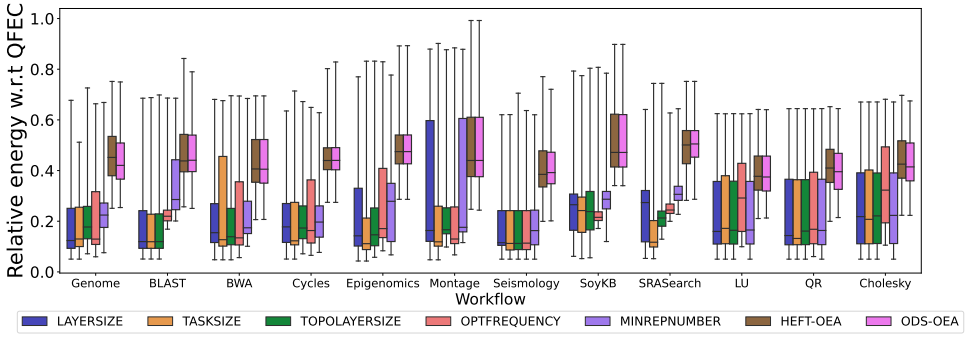

Figure 8: Average performance of the different heuristics for each workflow over all the parameter settings.

other heuristics but its 75th percentile is extremely large; therefore, even if TaskSize achieves very good average performance on that workflow it often achieves bad performance on it.

On average, LayerSize and TaskSize have an average energy gain of 60% over the competitors (Heft-OEA and ODS-OEA) and save 82% of the energy consumed by QFEC. Figure 9 presents a synthesis of the performance achieved by each heuristic, over the whole set of experiments, depending on the tightness of the deadline. It shows that when the deadline is the tightest, all heuristics achieve similar performance, even if the performance of ODS-OEA and Heft-OEA is slightly worse than that of the others. But as deadlines grow larger, the differences between heuristics grow larger too. Our heuristics perform far better, and they can save up to 95% of QFEC cost.

Table 4 presents statistics on the energy performance of the different heuristics over all the parameter settings. The performance reported is the energy used by each heuristic expressed as a fraction of the energy used by QFEC. We report the geometric mean (rather than the classical arithmetic mean) because we aggregate ratios and not raw values. Table 4 shows that TaskSize achieves the minimum mean energy cost. Its average performance is slightly better than those of LayerSize and TopoLayerSize. OptFrequency achieves slightly worse performance, and then MinRepNumber. The performance of both competitors (Heft-OEA and ODS-OEA) is

Table 4: Statistics over all parameter settings on the amount of energy used by each heuristic expressed as a ratio over the energy used by QFEC (hence, the lower the better). Best, worst and geometric average ratios are reported.

	LayerSize	TaskSize	TopoLayerSize	OptFrequency	MinRepNumber	Heft-OEA	ODS-OEA
Best	0.0430	0.0429	0.0478	0.0502	0.0502	0.1994	0.2015
Worst	0.8794	0.9018	0.8770	0.8843	0.8786	0.9923	0.9923
Mean	0.1948	**0.1721**	0.1962	0.2181	0.2373	0.4410	0.4390

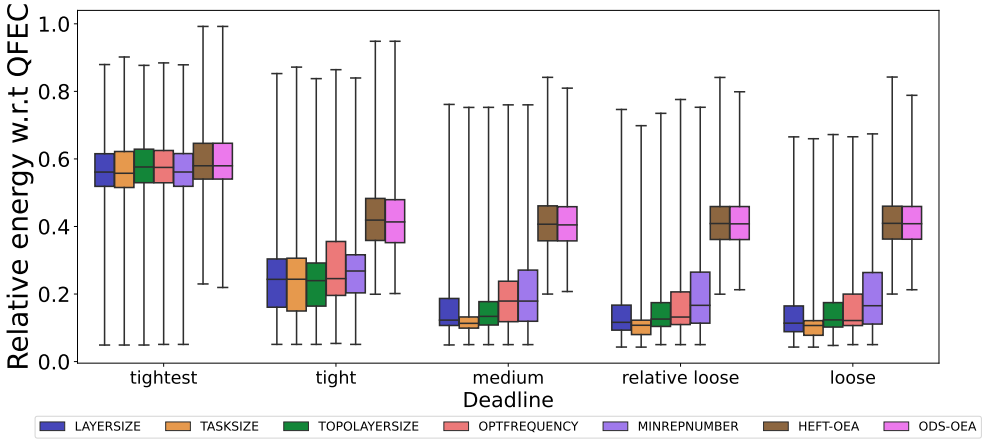

Figure 9: Average performance of the different heuristics for different deadlines over all the parameter settings.

significantly worse. These overall statistics confirm the previous analyses.

Because QFEC does not use DVFS, the fact that there are heuristics which achieve lower energy usage than QFEC shows that using DVFS enables to decrease energy consumption even if lowering processor voltage increases the fault rate which may lead to additional replicas.

Finally, we timed heuristics TaskSize and LayerSize on a subset of the real-world applications workflows (BWA, Cycles, Epigenomics, Genome), and on the QR factorization algorithm. For the real-world workflows, the running time of both TaskSize and LayerSize was under 0.5 second for workflows comprising 1,000 tasks, and under 31 seconds for 10,000 tasks (with an average at 11 seconds). For QR, their running time was under 0.9s for 1240 tasks, and under 5.2 seconds for 2870 tasks. Hence, TaskSize and LayerSize are both able to process large workflows in a fairly reasonable time. It should be noted that the fastest heuristic depends both upon the workflow and upon its size.

5 Related work

A lot of works focus on this tri-criteria optimization problem (reliability, deadlines, energy), but for independent task sets. Reference [27] proposes both static and dynamic methods to reduce overlapping between replicas, thus minimizing energy consumption. Reference [39] improves the approach of [27] through three steps, and proposes several new scheduling heuristics, which have

the best performance under a wide range of scenarios. Reference [40] considers an energy-budget-aware reliability management (enBudRM) method for multi-core embedded systems featuring hybrid energy source, while [41] proposes heuristic-based scheduling techniques to minimize the energy consumption of task executions when errors are absent, and preserves feasibility under the worst case of error occurrences. Reference [42] considers deadline-partition scheduling algorithm for periodic tasks on heterogeneous platforms. Reference [43] presents a low-overhead strategy to reduce energy consumption while enforcing deadlines on heterogeneous platforms. References [44, 45, 46, 47] introduce low-overhead multi-hierarchical heuristics to reduce energy consumption while satisfying temperature constraints for real-time systems.

There are other works that consider managing energy using DVFS for real-time workflows. Reference [48] considers precise scheduling of tasks in a mixed-criticality model. Reference [49] aims to schedule sporadic parallel tasks with deadline constraints while saving energy and proposes a real-time scheduler to solve this problem. Reference [50] considers energy-aware duplication scheduling algorithms for a parallel application on homogeneous systems. Reference [51] presents energy-conscious scheduling to implement joint reduction between the schedule length and energy consumption of a parallel application on heterogeneous systems. Reference [52] investigates the problem of reducing energy consumption with a schedule length constraint for a parallel application on heterogeneous systems by reclaiming the slack time for each task on its fixed assigned processor. Reference [53] also examines the same problem as that reported in [52] by switching off inefficient processors to reduce static energy consumption based on slack time reclamation. The MaxRe [54] and RR [55] algorithms aim at reducing resource consumption while satisfying the reliability goal of a parallel application on heterogeneous systems. However, they only aim to reduce resource consumption cost, which refers to the resource usage of processors when tasks are running.

The works listed above do not consider reliability, but it has been widely proved that DVFS has negative effects on transient failure rates. As a result, it is important to take reliability degradation into consideration while managing energy by DVFS. Reference [56] proposes reliability-aware power management schemes to save energy while guaranteeing a certain level of system reliability on homogeneous multiprocessors. Reference [57] introduces a power-efficient reliability management method through dynamic redundancy and voltage scaling under variations on homogeneous manycore processors. Reference [58] presents an N-modular redundancy (NMR) technique to achieve high reliability with low energy overhead for hard real-time applications on homogeneous multicore processors.

The tri-criteria problem is also studied in [59] but again with major differences: mapping of the tasks onto the processors is assumed to be given, as in [11], and reliability is achieved through re-execution, not replication. Similarly, reference [60] studies the tri-criteria optimization problem with a given mapping on heterogeneous architectures, assuming that the user specifies the maximum number of failures per processor tolerated to satisfy the reliability constraint. Reference [61] addresses the tri-criteria optimization problem by choosing some tasks that have to be re-executed to match the reliability constraint. However, they restrict to the scheduling problem on one single processor, and they consider only the energy consumption of the first execution of a task (best-case scenario) when re-execution is done. Reference [56] studied energy minimization problem for real-time workflows on homogeneous platforms, while reference [62] worked on the same problem on heterogeneous platforms. Both works save energy by scaling down frequencies of selected tasks, and preserve reliability by shared recovery. But they can only guarantee the original reliability level, which means all tasks running at f_{max} with no replica. Finally, reference [63] has proposed an off-line tri-criteria scheduling heuristic (TSH), which uses active replication to minimize the makespan, with a threshold on the global failure rate and the maximum power consumption. TSH is an improved critical-path list scheduling heuristic that

takes into account power and reliability before deciding which task to assign and to duplicate onto the next free processors. The complexity of this heuristic is unfortunately exponential in the number of processors. To the best of our knowledge, the only existing solutions based on replication that applies to workflows of arbitrary shape are OEA [11] and MILP [12], which we described in Section 4.1. As already stated, MILP allows at most two replicas per task and the original version of OEA has only one, which limits their solutions to low reliability thresholds only.

Finally, in all the works above, the mapping procedure always proceeds as a list-scheduling heuristic, where ready tasks are sorted according to some priority in a waiting queue and mapped in this order. The idea of mapping a chunk of ready tasks rather than only the task with the highest priority is used in [64] to achieve a better load-balancing at each decision step when targeting an heterogeneous platform. We re-use the idea but focusing on graph layers rather than arbitrary chunks, and with a different objective, namely to avoid overlap between replicas.

6 Conclusion

In this paper, we have introduced scheduling and mapping heuristics for real-time workflows, with a general tri-criteria objective (deadline, reliability, energy). A key design element is our novel layer-by-layer approach, which allows to considerably limit the overlap between the replicas of a task, thereby dramatically saving energy. To the best of our knowledge, these heuristics are the first to solve the general problem with an arbitrary number of replicas. We have assessed the performance of our heuristics via an extensive set of experimental scenarios, and we have shown that they significantly outperform the best available competitors from the literature. Specifically, our best heuristics LayerSize and TaskSize have an energy gain of 60% over the Heft-OEA and ODS-OEA competitors and save 82% of the energy consumed by QFEC. We also report partial comparisons with ExtMILP, which does not take communication times into account. On the contrary, our heuristics can deal with much larger workflows, account for communication times, and include as many replicas as needed to meet high reliability thresholds.

Future work will investigate whether complementary resilience techniques such as introducing intermediate error detectors and checkpointing can help decrease the number of replicas, and thereby the total energy consumption.

References

[1] S. Bharathi, A. Chervenak, E. Deelman, G. Mehta, M.-H. Su, K. Vahi, Characterization of scientific workflows, in: SC'08 Workshop: The 3rd Workshop on Workflows in Support of Large-scale Science (WORKS08) web site, ACM/IEEE, Austin, TX, 2008.

[2] E. Caron, F. Desprez, T. Glatard, M. Ketan, J. Montagnat, D. Reimert, Workflow-based comparison of two distributed computing infrastructures, in: Workflows in Support of Large-Scale Science (WORKS10), In Conjunction with Supercomputing 10 (SC'10), IEEE, New Orleans, 2010, hal-00677820.
URL `https://hal.inria.fr/hal-00677820`

[3] P. Couvares, T. Kosar, A. Roy, J. Weber, K. Wenger, Workflow Management in Condor, in: I. Taylor, E. Deelman, D. Gannon, M. Shields (Eds.), Workflows for e-Science, Springer, 2007, pp. 357–375.

[4] E. Deelman, G. Singh, M.-H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J. Jacob, D. Katz, Pegasus: a Framework for Mapping Complex Scientific Workflows onto Distributed Systems, Scientific Programming Journal 13 (3) (2005) 219–237.

[5] M. R. Garey, D. S. Johnson, Computers and Intractability, a Guide to the Theory of NP-Completeness, W.H. Freeman and Company, 1979.

[6] H. Casanova, A. Legrand, Y. Robert, Parallel Algorithms, Chapman and Hall/CRC Press, 2008.

[7] H. Topcuoglu, S. Hariri, M. Y. Wu, Performance-effective and low-complexity task scheduling for heterogeneous computing, IEEE Trans. Parallel Distributed Systems 13 (3) (2002) 260–274.

[8] C. Weaver, J. Emer, S. S. Mukherjee, S. K. Reinhardt, Techniques to reduce the soft error rate of a high-performance microprocessor, ACM SIGARCH Computer Architecture News 32 (2) (2004) 264.

[9] A. Burns, A. Wellings, Real-Time Systems and Programming Languages: Ada, Real-Time Java and C/Real-Time POSIX, 4th Edition, Addison-Wesley, 2009.

[10] A. Dixit, A. Wood, The impact of new technology on soft error rates, in: 2011 International Reliability Physics Symposium, IEEE, 2011, pp. 5B–4.

[11] J. Huang, R. Li, X. Jiao, Y. Jiang, W. Chang, Dynamic DAG Scheduling on Multiprocessor Systems: Reliability, Energy, and Makespan, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems 39 (11) (2020) 3336–3347.

[12] M. Cui, A. Kritikakou, L. Mo, E. Casseau, Fault-tolerant mapping of real-time parallel applications under multiple dvfs schemes, in: IEEE 27th Real-Time and Embedded Technology and Applications Symposium (RTAS), 2021, pp. 387–399.

[13] H. Chen, X. Zhu, D. Qiu, L. Liu, Uncertainty-aware real-time workflow scheduling in the cloud, in: IEEE 9th Int. Conf. Cloud Computing (CLOUD), 2016, pp. 577–584.

[14] D. C. Snowdon, G. Van Der Linden, S. M. Petters, G. Heiser, Accurate run-time prediction of performance degradation under frequency scaling, in: Workshop on Operating Systems Platforms for Embedded Real-Time applications, 2007, p. 58.

[15] D. Zhu, R. Melhem, D. Mosse, The effects of energy management on reliability in real-time embedded systems, in: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2004, pp. 35–40.

[16] B. Zhao, H. Aydin, D. Zhu, Reliability-aware dynamic voltage scaling for energy-constrained real-time embedded systems, in: Proceedings of the IEEE International Conference on Computer Design (ICCD), 2008, pp. 633–639.

[17] A. Dixit, A. Wood, The impact of new technology on soft error rates, in: IEEE International on Reliability Physics Symposium (IRPS), 2011, pp. 5B.4.1–5B.4.7.

[18] G. Aupy, A. Benoit, Y. Robert, Energy-aware scheduling under reliability and makespan constraints, in: Proceedings of the International Conference on High Performance Computing (HiPC), 2012, pp. 1–10.

[19] A. Das, A. Kumar, B. Veeravalli, C. Bolchini, A. Miele, Combined DVFS and mapping exploration for lifetime and soft-error susceptibility improvement in MPSoCs, in: Proceedings of the Conference on Design, Automation & Test in Europe (DATE), 2014, pp. 61:1–61:6.

[20] M. Patterson, The effect of data center temperature on energy efficiency, in: Proceedings of 11th Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems, 2008, pp. 1167–1174.

[21] W.-C. Feng, Making a case for efficient supercomputing, Queue 1 (7) (2003) 54–64.

[22] C.-H. Hsu, W.-C. Feng, A power-aware run-time system for high-performance computing, in: Proceedings of the ACM/IEEE Supercomputing Conference (SC), 2005, pp. 1–9.

[23] O. Sarood, E. Meneses, L. V. Kale, A 'cool' way of improving the reliability of HPC machines, in: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis (SC), 2013, pp. 58:1–58:12.

[24] H. K. N. Leung, P. W. L. Wong, A study of user acceptance tests, Softw. Qual. J. 6 (2) (1997) 137–149. doi:10.1023/A:1018503800709.
URL https://doi.org/10.1023/A:1018503800709

[25] G. Manimaran, C. S. R. Murthy, A fault-tolerant dynamic scheduling algorithm for multiprocessor real-time systems and its analysis, IEEE Trans. Parallel Distributed Syst. 9 (11) (1998) 1137–1152. doi:10.1109/71.735960.
URL https://doi.org/10.1109/71.735960

[26] G. Yao, Y. Ding, K. Hao, Using imbalance characteristic for fault-tolerant workflow scheduling in cloud systems, IEEE Trans. Parallel Distributed Syst. 28 (12) (2017) 3671–3683. doi:10.1109/TPDS.2017.2687923.
URL https://doi.org/10.1109/TPDS.2017.2687923

[27] M. A. Haque, H. Aydin, D. Zhu, On reliability management of energy-aware real-time systems through task replication, IEEE Transactions on Parallel and Distributed Systems 28 (3) (2017) 813–825.

[28] G. Xie, G. Zeng, R. Li, K. Li, Scheduling Parallel Applications on Heterogeneous Distributed Systems, Springer, 2019.

[29] F. Glover, E. Woolsey, Technical note: Converting the 0-1 polynomial programming problem to a 0-1 linear program, Operations Research 22 (1) (1974) 180–182.

[30] Simulator, https://github.com/wuzhui1995/Energy-aware-mapping-and-scheduling-strategies.

[31] N. B. Rizvandi, A. Y. Zomaya, Y. C. Lee, A. J. Boloori, J. Taheri, Multiple frequency selection in dvfs-enabled processors to minimize energy consumption, Energy-Efficient Distributed Computing Systems (2012) 443–463.

[32] Frequency behavior-intel, https://en.wikichip.org/wiki/intel/frequency_behavior.

[33] Z. Wu, L. Han, J. Liu, Y. Robert, F. Vivien, Energy-aware mapping and scheduling strategies for real-time workflows under reliability constraints, Research Report RR-9469, INRIA (Jul. 2022).
URL https://hal.inria.fr/hal-03641039

[34] R. F. da Silva, W. Chen, G. Juve, K. Vahi, E. Deelman, Community resources for enabling research in distributed scientific workflows, in: e-Science (e-Science), 2014 IEEE 10th International Conference on, Vol. 1, IEEE, 2014, pp. 177–184.

[35] S. Bharathi, A. Chervenak, E. Deelman, G. Mehta, M.-H. Su, K. Vahi, Characterization of scientific workflows, in: Workflows in Support of Large-Scale Science (WORKS), IEEE, 2008, pp. 1–10.

[36] G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, K. Vahi, Characterizing and profiling scientific workflows, Future Generation Computer Systems 29 (3) (2013) 682–692.

[37] J. Choi, J. J. Dongarra, L. S. Ostrouchov, A. P. Petitet, D. W. Walker, R. C. Whaley, Design and implementation of the scalapack lu, qr, and cholesky factorization routines, Scientific Programming 5 (3) (1996) 173–184.

[38] G. H. Golub, C. F. V. Loan, Matrix Computations, Fourth Edition, 4th Edition, Johns Hopkins University Press, 2013.

[39] L. Han, L.-C. Canon, J. Liu, Y. Robert, F. Vivien, Improved energy-aware strategies for periodic real-time tasks under reliability constraints, in: RTSS'2019, the 40th IEEE Real-Time Systems Symposium, IEEE Press, 2019.

[40] S. Safari, M. Ansari, M. Salehi, A. Ejlali, Energy-budget-aware reliability management in multi-core embedded systems with hybrid energy source, CSI Journal on Computer Science and Engineering 15 (2018) 31–43.

[41] Y. Liu, H. Liang, K. Wu, Scheduling for energy efficiency and fault tolerance in hard real-time systems, in: Design, Automation and Test in Europe, DATE 2010, Dresden, Germany, March 8-12, 2010, 2010, pp. 1444–1449. doi:10.1109/DATE.2010.5457039.
URL https://doi.org/10.1109/DATE.2010.5457039

[42] S. Moulik, R. Devaraj, A. Sarkar, A. Shaw, A deadline-partition oriented heterogeneous multi-core scheduler for periodic tasks, in: S. Horng (Ed.), 18th International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT 2017, Taipei, Taiwan, December 18-20, 2017, IEEE Computer Society, 2017, pp. 204–210. doi:10.1109/PDCAT.2017.00041.
URL https://doi.org/10.1109/PDCAT.2017.00041

[43] S. Moulik, Z. Das, R. Devaraj, S. Chakraborty, SEAMERS: A semi-partitioned energy-aware scheduler for heterogeneous multicorereal-time systems, J. Syst. Archit. 114 (2021) 101953. doi:10.1016/j.sysarc.2020.101953.
URL https://doi.org/10.1016/j.sysarc.2020.101953

[44] Y. Sharma, Z. Das, S. Moulik, TEFRED: A temperature and energy cognizant fault-tolerant real-time scheduler based on deadline partitioning for heterogeneous platforms, in: H. Shen, Y. Sang, Y. Zhang, N. Xiao, H. R. Arabnia, G. C. Fox, A. Gupta, M. Malek (Eds.), Parallel and Distributed Computing, Applications and Technologies - 22nd International Conference, PDCAT 2021, Guangzhou, China, December 17-19, 2021, Proceedings, Vol. 13148 of Lecture Notes in Computer Science, Springer, 2021, pp. 358–366. doi:10.1007/978-3-030-96772-7_33.
URL https://doi.org/10.1007/978-3-030-96772-7_33

[45] S. Moulik, Z. Das, TASOR: A temperature-aware semi-partitioned real-time scheduler, in: TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), Kochi, India, October 17-20, 2019, IEEE, 2019, pp. 1578–1583. doi:10.1109/TENCON.2019.8929501.
URL https://doi.org/10.1109/TENCON.2019.8929501

[46] Y. Sharma, S. Chakraborty, S. Moulik, ETA-HP: an energy and temperature-aware real-time scheduler for heterogeneous platforms, J. Supercomput. 78 (8) (2022) 1–25. doi:10.1007/s11227-021-04257-7.
URL https://doi.org/10.1007/s11227-021-04257-7

[47] S. Moulik, RESET: A real-time scheduler for energy and temperature aware heterogeneous multi-core systems, Integr. 77 (2021) 59–69. doi:10.1016/j.vlsi.2020.11.012.
URL https://doi.org/10.1016/j.vlsi.2020.11.012

[48] A. Bhuiyan, S. Sruti, Z. Guo, K. Yang, Precise scheduling of mixed-criticality tasks by varying processor speed, in: J. Ermont, Y. Song, C. D. Gill (Eds.), Proceedings of the 27th International Conference on Real-Time Networks and Systems, RTNS 2019, Toulouse, France, November 06-08, 2019, ACM, 2019, pp. 123–132. doi:10.1145/3356401.3356410.
URL https://doi.org/10.1145/3356401.3356410

[49] Z. Guo, A. Bhuiyan, D. Liu, A. Khan, A. Saifullah, N. Guan, Energy-efficient real-time scheduling of DAGs on clustered multi-core platforms, in: B. B. Brandenburg (Ed.), 25th IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS 2019, Montreal, QC, Canada, April 16-18, 2019, IEEE, 2019, pp. 156–168. doi:10.1109/RTAS.2019.00021.
URL https://doi.org/10.1109/RTAS.2019.00021

[50] Z. Zong, A. Manzanares, X. Ruan, X. Qin, EAD and PEBD: two energy-aware duplication scheduling algorithms for parallel tasks on homogeneous clusters, IEEE Trans. Computers 60 (3) (2011) 360–374. doi:10.1109/TC.2010.216.
URL https://doi.org/10.1109/TC.2010.216

[51] Y. C. Lee, A. Y. Zomaya, Energy conscious scheduling for distributed computing systems under different operating conditions, IEEE Trans. Parallel Distributed Syst. 22 (8) (2011) 1374–1381. doi:10.1109/TPDS.2010.208.
URL https://doi.org/10.1109/TPDS.2010.208

[52] Q. Huang, S. Su, J. Li, P. Xu, K. Shuang, X. Huang, Enhanced energy-efficient scheduling for parallel applications in cloud, in: 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2012, Ottawa, Canada, May 13-16, 2012, 2012, pp. 781–786. doi:10.1109/CCGrid.2012.49.
URL https://doi.org/10.1109/CCGrid.2012.49

[53] Z. Tang, L. Qi, Z. Cheng, K. Li, S. U. Khan, K. Li, An energy-efficient task scheduling algorithm in dvfs-enabled cloud environment, J. Grid Comput. 14 (1) (2016) 55–74. doi:10.1007/s10723-015-9334-y.
URL https://doi.org/10.1007/s10723-015-9334-y

[54] L. Zhao, Y. Ren, Y. Xiang, K. Sakurai, Fault tolerant scheduling with dynamic number of replicas in heterogeneous system, in: 12th IEEE International Conference on High Performance Computing and Communications, HPCC 2010, 1-3 September 2010, Melbourne, Australia, 2010, pp. 434–441. doi:10.1109/HPCC.2010.72.
URL https://doi.org/10.1109/HPCC.2010.72

[55] L. Zhao, Y. Ren, K. Sakurai, Reliable workflow scheduling with less resource redundancy, Parallel Comput. 39 (10) (2013) 567–585. doi:10.1016/j.parco.2013.06.003.
URL https://doi.org/10.1016/j.parco.2013.06.003

[56] Y. Guo, D. Zhu, H. Aydin, Reliability-aware power management for parallel real-time applications with precedence constraints, in: Proceedings of the 2011 International Green Computing Conference and Workshops, IGCC '11, IEEE Computer Society, USA, 2011, p. 1–8. doi:10.1109/IGCC.2011.6008562.
URL https://doi.org/10.1109/IGCC.2011.6008562

[57] M. Salehi, M. K. Tavana, S. Rehman, F. Kriebel, M. Shafique, A. Ejlali, J. Henkel, DRVS: power-efficient reliability management through dynamic redundancy and voltage scaling under variations, in: IEEE/ACM International Symposium on Low Power Electronics and Design, ISLPED 2015, Rome, Italy, July 22-24, 2015, 2015, pp. 225–230. doi:10.1109/ISLPED.2015.7273518.
URL https://doi.org/10.1109/ISLPED.2015.7273518

[58] M. Salehi, A. Ejlali, B. M. Al-Hashimi, Two-phase low-energy n-modular redundancy for hard real-time multi-core systems, IEEE Trans. Parallel Distributed Syst. 27 (5) (2016) 1497–1510. doi:10.1109/TPDS.2015.2444402.
URL https://doi.org/10.1109/TPDS.2015.2444402

[59] G. Aupy, A. Benoit, Y. Robert, Energy-aware scheduling under reliability and makespan constraints, in: International Conference on High Performance Computing (HiPC'2012), IEEE Computer Society Press, 2012.

[60] P. Pop, K. H. Poulsen, V. Izosimov, P. Eles, Scheduling and voltage scaling for energy/reliability trade-offs in fault-tolerant time-triggered embedded systems, in: Proc. of IEEE/ACM Int. Conf. on Hardware/software codesign and system synthesis (CODES+ISSS), 2007, pp. 233–238.

[61] D. Zhu, H. Aydin, Energy management for real-time embedded systems with reliability requirements, in: Proceedings of the IEEE/ACM International Conference on Computer-aided Design (ICCAD), 2006, pp. 528–534.

[62] L. Zhang, K. Li, K. Li, Y. Xu, Joint optimization of energy efficiency and system reliability for precedence constrained tasks in heterogeneous systems, International Journal of Electrical Power and Energy Systems 78 (2016) 499–512. doi:https://doi.org/10.1016/j.ijepes.2015.11.102.
URL https://www.sciencedirect.com/science/article/pii/S0142061515005335

[63] I. Assayad, A. Girault, H. Kalla, Tradeoff exploration between reliability, power consumption, and execution time for embedded systems, International Journal on Software Tools for Technology Transfer 15 (3) (2013) 229–245.

[64] O. Beaumont, V. Boudet, Y. Robert, The iso-level scheduling heuristic for heterogeneous processors, in: PDP'2002, 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, IEEE Computer Society Press, 2002.

Contents

A Type of solution found by MILP depending on the size of workflows

Table 5: Type of solution found by MILP with a reliability threshold of 0.9.

	task number	dl=1	dl=2	dl=3	dl=4	dl=5
BLAST	10	Optimal	Optimal	Optimal	Optimal	Optimal
	20	Suboptimal	Suboptimal	Optimal	Optimal	Optimal
	30	No Result	No Result	Optimal	Optimal	Optimal
	40	No Result	No Result	Suboptimal	Optimal	Optimal
	50	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	60	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	70	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
BWA	10	Optimal	Optimal	Optimal	Optimal	Optimal
	20	Suboptimal	Suboptimal	Optimal	Optimal	Optimal
	30	Suboptimal	Suboptimal	Suboptimal	Optimal	Optimal
	40	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	50	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	60	No Result	No Result	Suboptimal	Suboptimal	Suboptimal
	70	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
Cycles	10	Optimal	Optimal	Optimal	Optimal	Optimal
	20	Optimal	Optimal	Optimal	Optimal	Optimal
	30	Suboptimal	Suboptimal	Optimal	Optimal	Optimal
	40	Suboptimal	Suboptimal	Suboptimal	Optimal	Optimal
	50	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	60	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	70	No Result	No Result	No Result	Suboptimal	Suboptimal
Epigenomics	10	Optimal	Optimal	Optimal	Optimal	Optimal
	20	Optimal	Optimal	Optimal	Optimal	Optimal
	30	Optimal	Optimal	Optimal	Optimal	Optimal
	40	Suboptimal	Suboptimal	Optimal	Optimal	Optimal
	50	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	60	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	70	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Suboptimal
Genome	10	Optimal	Optimal	Optimal	Optimal	Optimal
	20	Optimal	Suboptimal	Suboptimal	Suboptimal	Optimal
	30	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	40	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	50	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	60	No Result	No Result	Suboptimal	No Result	Suboptimal
	70	No Result	No Result	No Result	No Result	Suboptimal
Seismology	10	Optimal	Optimal	Optimal	Optimal	Optimal
	20	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Optimal
	30	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Optimal
	40	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Optimal
	50	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	60	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	70	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
SoyKB	10	Optimal	Optimal	Optimal	Optimal	Optimal
	20	Optimal	Optimal	Optimal	Optimal	Optimal
	30	Suboptimal	Optimal	Optimal	Optimal	Optimal
	40	Suboptimal	Optimal	Optimal	Optimal	Optimal
	50	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	60	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	70	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
SRASearch	10	Optimal	Optimal	Optimal	Optimal	Optimal
	20	Suboptimal	Optimal	Optimal	Optimal	Optimal
	30	No Result	Optimal	Optimal	Optimal	Optimal
	40	No Result	Suboptimal	Suboptimal	Optimal	Optimal
	50	No Result	No Result	No Result	No Result	Suboptimal
	60	No Result	No Result	No Result	No Result	Suboptimal
	70	No Result	No Result	No Result	No Result	Suboptimal

Table 6: Type of solution found by MILP with a reliability threshold of 0.93.

	task number	dl=1	dl=2	dl=3	dl=4	dl=5
BLAST	10	Optimal	Suboptimal	Optimal	Optimal	Optimal
	20	Suboptimal	Suboptimal	Optimal	Optimal	Optimal
	30	No Result	No Result	Optimal	Optimal	Optimal
	40	No Result	No Result	No Result	Suboptimal	Suboptimal
	50	No Result	No Result	No Result	Suboptimal	Suboptimal
	60	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	70	No Result	No Result	No Result	Suboptimal	Suboptimal
BWA	10	Optimal	Optimal	Optimal	Optimal	Optimal
	20	Suboptimal	Suboptimal	Optimal	Optimal	Optimal
	30	Suboptimal	Suboptimal	Optimal	Optimal	Optimal
	40	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	50	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	60	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	70	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
Cycles	10	Optimal	Optimal	Optimal	Optimal	Optimal
	20	Optimal	Optimal	Optimal	Optimal	Optimal
	30	Suboptimal	Suboptimal	Suboptimal	Optimal	Optimal
	40	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	50	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	60	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	70	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
Epigenomics	10	Optimal	Optimal	Optimal	Optimal	Optimal
	20	Optimal	Optimal	Optimal	Optimal	Optimal
	30	Optimal	Optimal	Optimal	Optimal	Optimal
	40	Suboptimal	Suboptimal	Optimal	Optimal	Optimal
	50	No Result	Suboptimal	Suboptimal	No Result	Suboptimal
	60	No Result	Suboptimal	Suboptimal	No Result	Suboptimal
	70	No Result	Suboptimal	No Result	Suboptimal	Suboptimal
Genome	10	Optimal	Optimal	Optimal	Optimal	Optimal
	20	Suboptimal	Suboptimal	Suboptimal	Optimal	Optimal
	30	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	40	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	50	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	60	No Result	No Result	Suboptimal	Suboptimal	Suboptimal
	70	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
Seismology	10	Optimal	Optimal	Optimal	Optimal	Optimal
	20	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Optimal
	30	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Optimal
	40	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	50	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	60	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	70	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
SoyKB	10	Optimal	Optimal	Optimal	Optimal	Optimal
	20	Optimal	Optimal	Optimal	Optimal	Optimal
	30	Suboptimal	Optimal	Optimal	Optimal	Optimal
	40	Suboptimal	Optimal	Optimal	Optimal	Optimal
	50	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	60	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	70	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
SRASearch	10	Optimal	Optimal	Optimal	Optimal	Optimal
	20	Optimal	Optimal	Optimal	Optimal	Optimal
	30	Suboptimal	Optimal	Optimal	Optimal	Optimal
	40	No Result	Optimal	Suboptimal	Optimal	Suboptimal
	50	No Result	No Result	No Result	No Result	Suboptimal
	60	No Result	No Result	No Result	No Result	Suboptimal
	70	No Result	No Result	No Result	No Result	Suboptimal

Table 7: Type of solution found by MILP with a reliability threshold of 0.96.

	task number	dl=1	dl=2	dl=3	dl=4	dl=5
BLAST	10	Optimal	Optimal	Optimal	Optimal	Optimal
	20	Suboptimal	Suboptimal	Optimal	Optimal	Optimal
	30	No Result	Optimal	Optimal	Optimal	Optimal
	40	No Result	No Result	Suboptimal	Suboptimal	Suboptimal
	50	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	60	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	70	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
BWA	10	Optimal	Optimal	Optimal	Optimal	Optimal
	20	Suboptimal	Suboptimal	Optimal	Optimal	Optimal
	30	Suboptimal	Suboptimal	Optimal	Optimal	Optimal
	40	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	50	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	60	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	70	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
Cycles	10	Optimal	Optimal	Optimal	Optimal	Optimal
	20	Optimal	Optimal	Optimal	Optimal	Optimal
	30	Suboptimal	Suboptimal	Suboptimal	Optimal	Optimal
	40	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	50	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	60	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	70	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
Epigenomics	10	Optimal	Optimal	Optimal	Optimal	Optimal
	20	Optimal	Optimal	Optimal	Optimal	Optimal
	30	Optimal	Optimal	Optimal	Optimal	Optimal
	40	Optimal	Optimal	Optimal	Optimal	Optimal
	50	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	60	No Result	Suboptimal	No Result	Suboptimal	Suboptimal
	70	No Result	No Result	No Result	Suboptimal	Suboptimal
Genome	10	Optimal	Optimal	Optimal	Optimal	Optimal
	20	Suboptimal	Suboptimal	Suboptimal	Optimal	Optimal
	30	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Optimal
	40	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	50	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	60	No Result	Suboptimal	No Result	No Result	Suboptimal
	70	No Result	No Result	No Result	Suboptimal	Suboptimal
Seismology	10	Optimal	Optimal	Optimal	Optimal	Optimal
	20	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Optimal
	30	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Optimal
	40	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	50	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	60	Suboptimal	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	70	Suboptimal	Suboptimal	No Result	Suboptimal	Suboptimal
SoyKB	10	Optimal	Optimal	Optimal	Optimal	Optimal
	20	Optimal	Optimal	Optimal	Optimal	Optimal
	30	Optimal	Optimal	Optimal	Optimal	Optimal
	40	Optimal	Optimal	Optimal	Optimal	Optimal
	50(replica number=3)	No Result	Suboptimal	Suboptimal	No Result	Suboptimal
	60(replica number=3)	No Result	No Result	No Result	No Result	No Result
	70(replica number=3)	No Result	No Result	No Result	No Result	No Result
SRASearch	10	Optimal	Optimal	Optimal	Optimal	Optimal
	20	Suboptimal	Optimal	Optimal	Optimal	Optimal
	30	No Result	Optimal	Optimal	Optimal	Optimal
	40	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	50	No Result	Suboptimal	Suboptimal	Suboptimal	Suboptimal
	60	No Result	No Result	No Result	No Result	Suboptimal
	70	No Result	No Result	No Result	No Result	No Result

B Statistics on the performance of the different heursitics for the different workflows

Table 8: Statistics on the performance of each heuristic for each workflow, with best cases highlighted in boldface.

	instance	Genome	BLAST	BWA	Cycles	Epigenomics	Montage
LayerSize	Best	**0.0504**	**0.0508**	0.0479	0.0506	0.0430	0.0478
	Worst	0.6772	**0.6855**	0.6808	0.6352	**0.7699**	0.8793
	Mean	0.1582	0.1508	0.1898	0.1851	0.1887	0.2245
TaskSize	Best	**0.0504**	**0.0508**	**0.0477**	**0.0505**	**0.0428**	**0.0477**
	Worst	**0.5118**	0.6879	**0.6759**	0.7140	0.8316	0.9017
	Mean	**0.1513**	**0.1444**	0.1753	**0.1683**	**0.1429**	**0.1554**
TopoLayerSize	Best	0.0715	**0.0508**	0.0477	0.0715	0.0576	0.0982
	Worst	0.7260	0.6983	0.6953	0.6723	0.8318	**0.8769**
	Mean	0.2035	0.1512	0.1745	0.1985	0.1785	0.2159
OptFrequency	Best	0.0597	0.1685	0.0564	0.0653	0.0833	0.0672
	Worst	0.6635	0.6861	0.6942	0.6492	0.8293	0.8842
	Mean	0.1774	0.2562	**0.1696**	0.1961	0.2395	0.1844
MinRepNumber	Best	0.0757	0.2009	0.1011	0.0768	0.0672	0.1157
	Worst	0.6687	**0.6855**	0.6845	**0.6284**	0.7769	0.8786
	Mean	0.2303	0.3344	0.2265	0.2133	0.2546	0.2986
Heft-OEA	Best	0.2509	0.2569	0.2063	0.2745	0.2864	0.2471
	Worst	0.7517	0.8423	0.6947	0.8023	0.8919	0.9922
	Mean	0.4466	0.4541	0.4186	0.4510	0.4862	0.4758
ODS-OEA	Best	0.2540	0.2505	0.2070	0.2743	0.2863	0.2436
	Worst	0.7497	0.7897	0.6946	0.8285	0.8929	0.9922
	Mean	0.4291	0.4520	0.4173	0.4510	0.4862	0.4756
	instance	Seismology	SoyKB	SRASearch	LU	QR	Cholesky
LayerSize	Best	**0.0502**	0.0613	0.5316	**0.0502**	**0.0502**	**0.0502**
	Worst	0.6203	0.7927	0.6410	0.6247	0.6442	**0.6704**
	Mean	0.1530	0.2593	0.2273	**0.1927**	0.1842	0.2093
TaskSize	Best	**0.0502**	**0.0524**	**0.0521**	**0.0502**	**0.0502**	0.0502
	Worst	0.6209	**0.7746**	0.7439	0.6241	0.6446	0.6706
	Mean	**0.1409**	**0.2390**	**0.1578**	0.1963	**0.1834**	**0.2082**
TopoLayerSize	Best	**0.0502**	0.0558	0.1290	**0.0502**	**0.0502**	**0.0502**
	Worst	0.7053	0.8037	0.7441	0.6246	0.6444	0.6707
	Mean	0.1426	0.2478	0.2469	0.1967	0.1892	0.2183
OptFrequency	Best	**0.0502**	0.1713	0.1995	0.0994	0.0605	0.1055
	Worst	0.6368	0.8074	**0.6273**	**0.6239**	**0.6438**	0.6813
	Mean	0.1432	0.2676	0.2809	0.2651	0.2030	0.2922
MinRepNumber	Best	**0.0502**	0.1198	0.2271	**0.0502**	**0.0502**	**0.0502**
	Worst	**0.6202**	0.7843	0.6440	0.6246	0.6446	0.6706
	Mean	0.1664	0.2961	0.3308	0.1971	0.1902	0.2188
Heft-OEA	Best	0.1994	0.3406	0.2822	0.2106	0.1995	0.2234
	Worst	0.7706	0.8979	0.7518	0.6411	0.6521	0.6968
	Mean	0.3978	0.5030	0.4823	0.3793	0.4031	0.4266
ODS-OEA	Best	0.2015	0.3402	0.2866	0.2127	0.2145	0.2231
	Worst	0.7207	0.8979	0.7519	0.6402	0.6446	0.6747
	Mean	0.4060	0.5028	0.4922	0.3767	0.3905	0.4159

C Timing experiments

Workflow	Heuristic	Number of tasks				
		200	300	600	1000	10 000
BWA	LayerSize	0.023	0.044	0.090	0.138	2.564
	TaskSize	0.013	0.030	0.057	0.102	2.015
Cycles	LayerSize	0.012	0.024	0.049	0.084	5.238
	TaskSize	0.015	0.054	0.105	0.402	25.731
Genome	LayerSize	0.009	0.025	0.051	0.093	2.510
	TaskSize	0.012	0.025	0.039	0.080	14.884
Epigenomics	LayerSize	0.023	0.434	0.348	0.183	30.532
	TaskSize	0.007	0.026	0.036	0.082	4.135

Table 9: Execution times in seconds of heuristics TaskSize and LayerSize for different workflow sizes.

Heuristic	Number of tasks	
	1240	2870
LayerSize	0.876	5.153
TaskSize	0.075	0.278

Table 10: Execution times in seconds of heuristics TaskSize and LayerSize for QR for different matrix sizes.

D Actual execution times drawn from a truncated normal distribution

D.1 $BC/WC = 0.1$

D.1.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

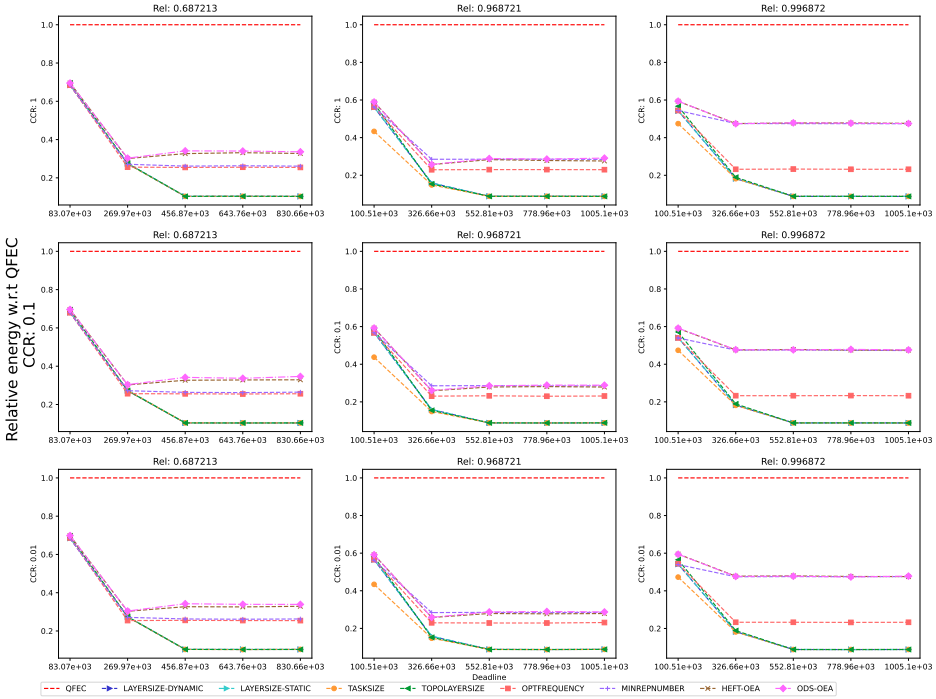

Figure 10: Performance of the different heuristics on the BLAST workflow.

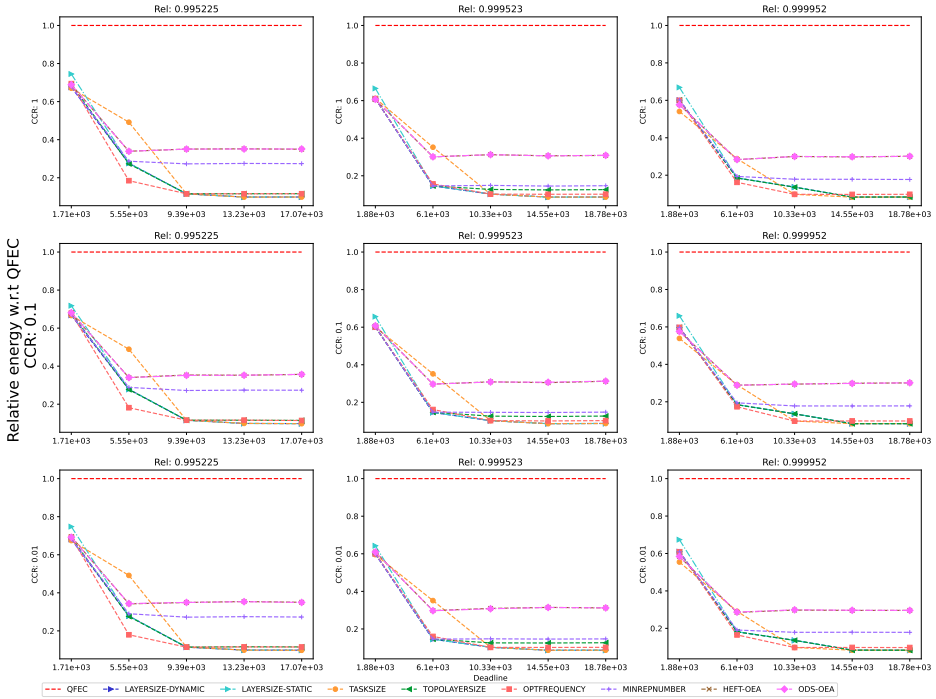

Figure 11: Performance of the different heuristics on the BWA workflow.

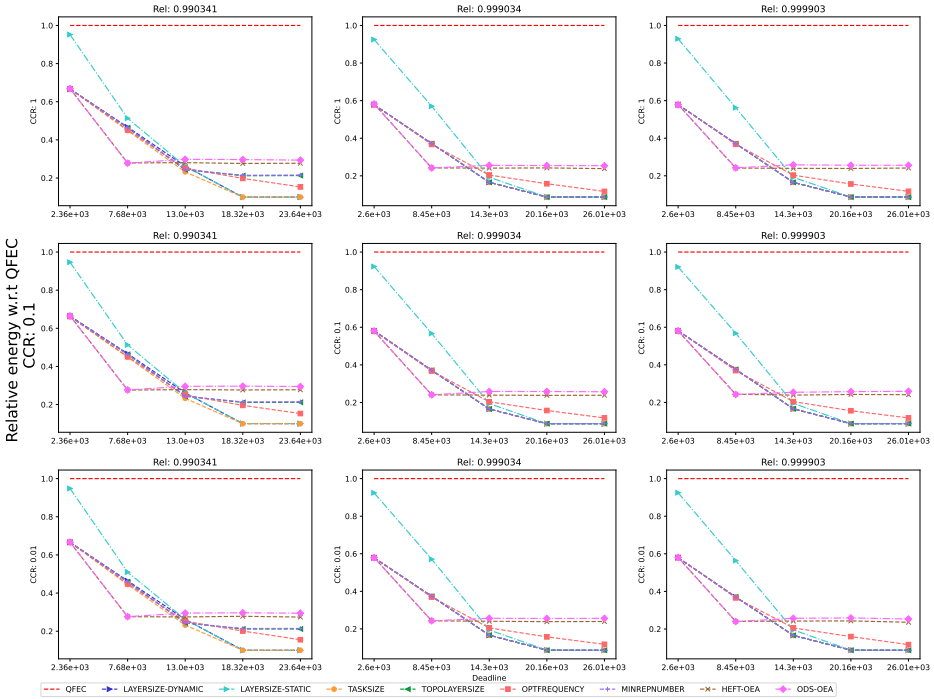

Figure 12: Performance of the different heuristics on the Cholesky workflow.

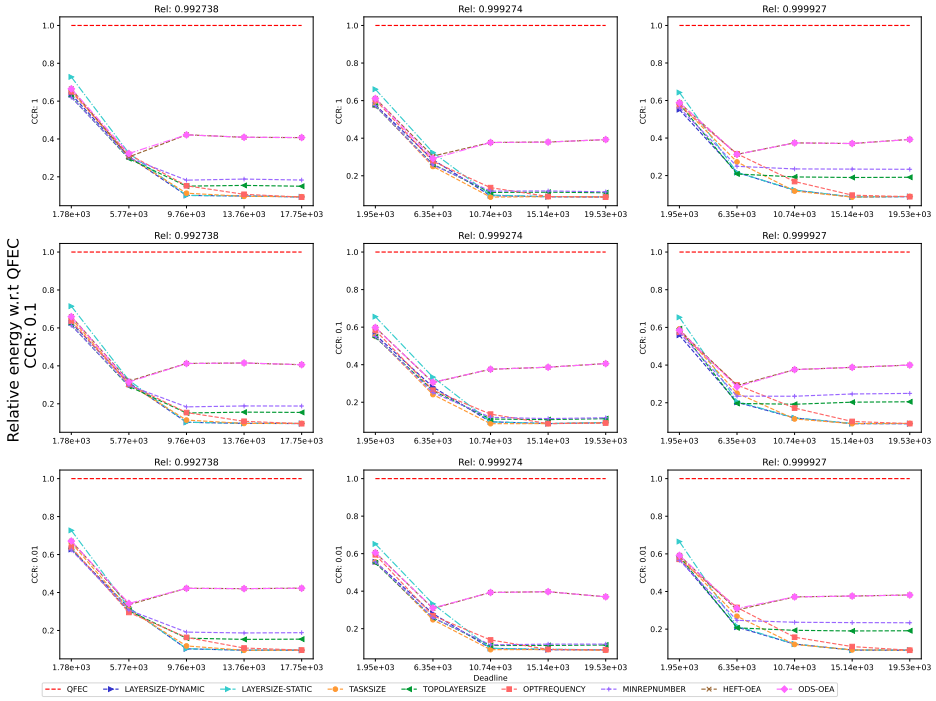

Figure 13: Performance of the different heuristics on the Cycles workflow.

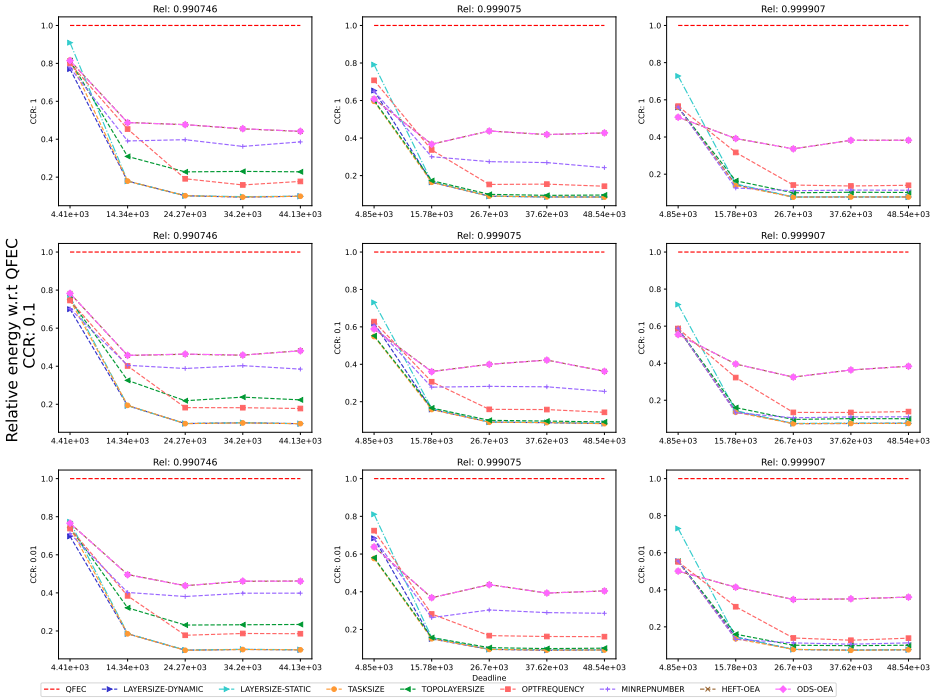

Figure 14: Performance of the different heuristics on the Epigenomics workflow.

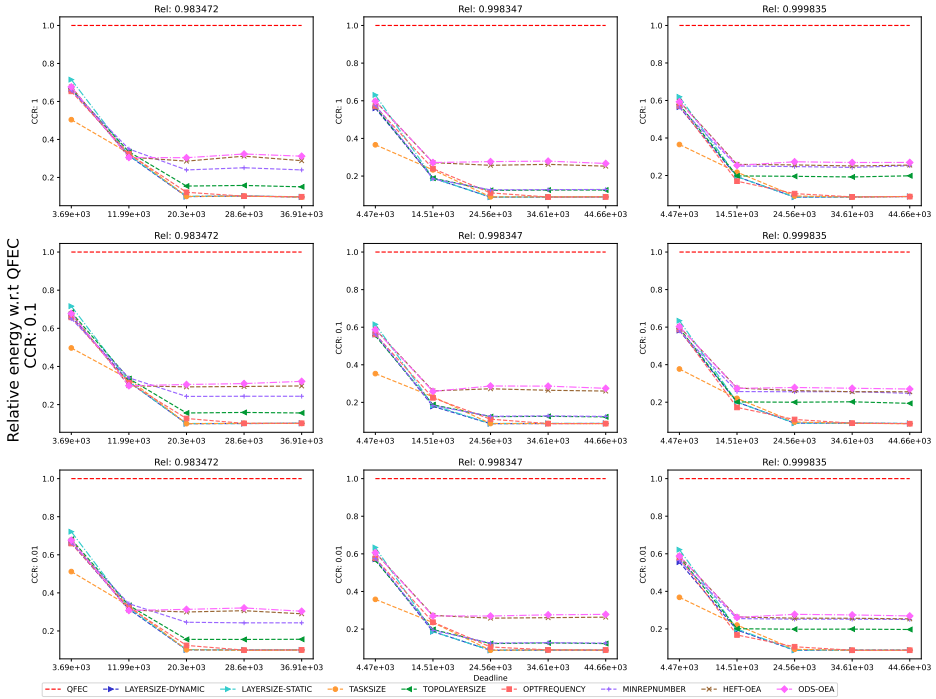

Figure 15: Performance of the different heuristics on the Genome workflow.

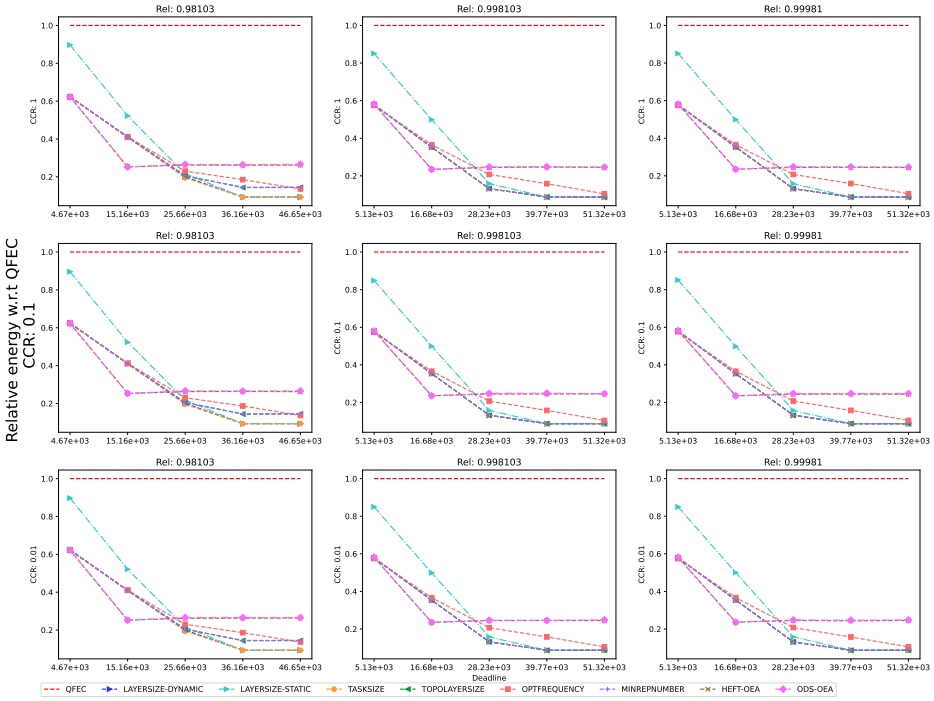

Figure 16: Performance of the different heuristics on the LU workflow.

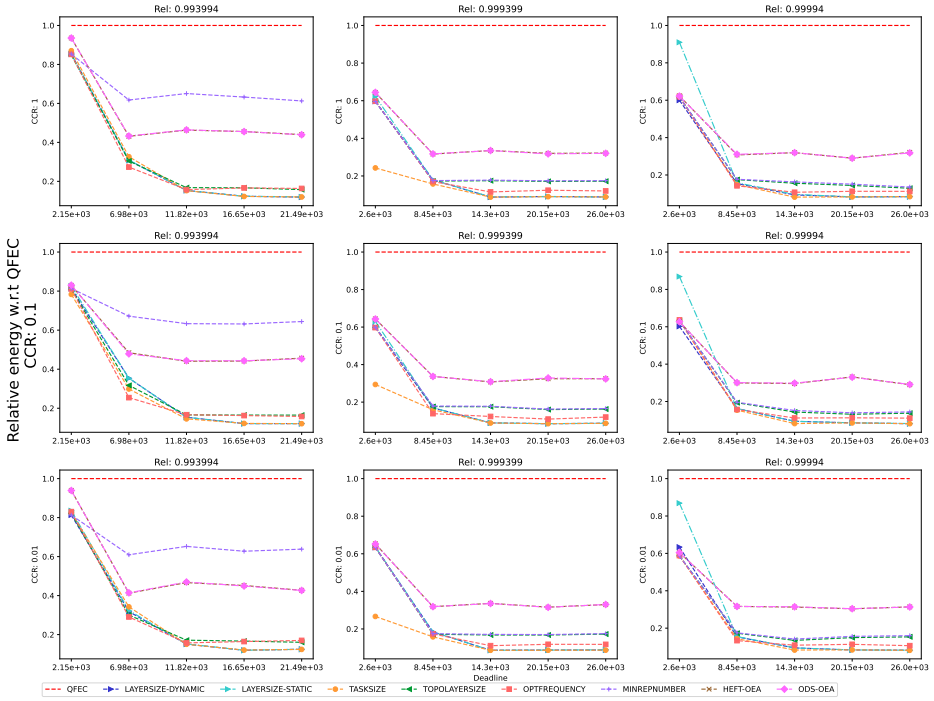

Figure 17: Performance of the different heuristics on the Montage workflow.

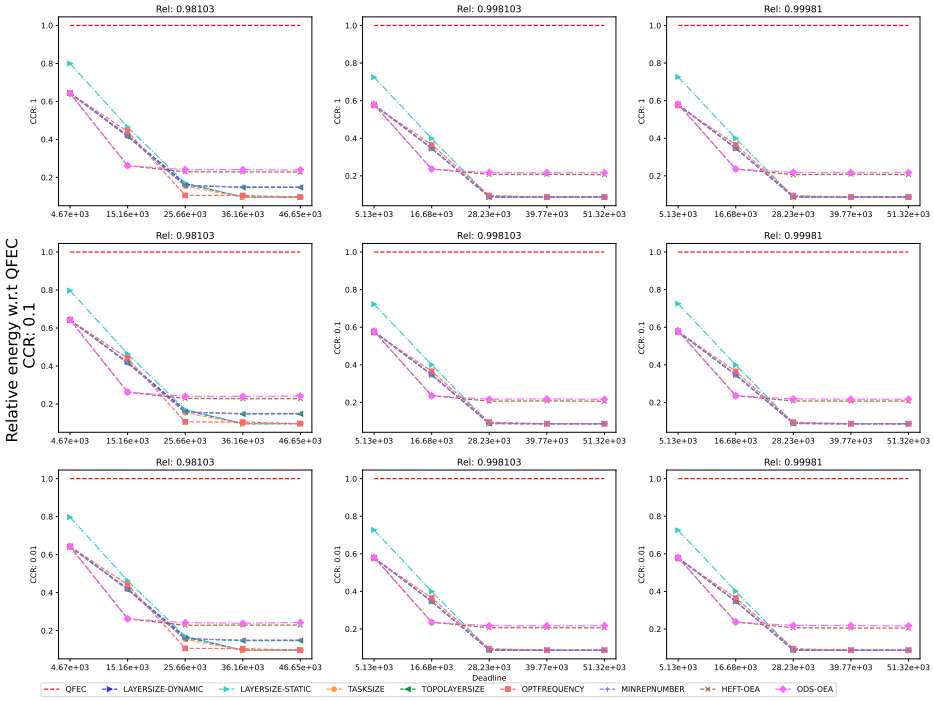

Figure 18: Performance of the different heuristics on the QR workflow.

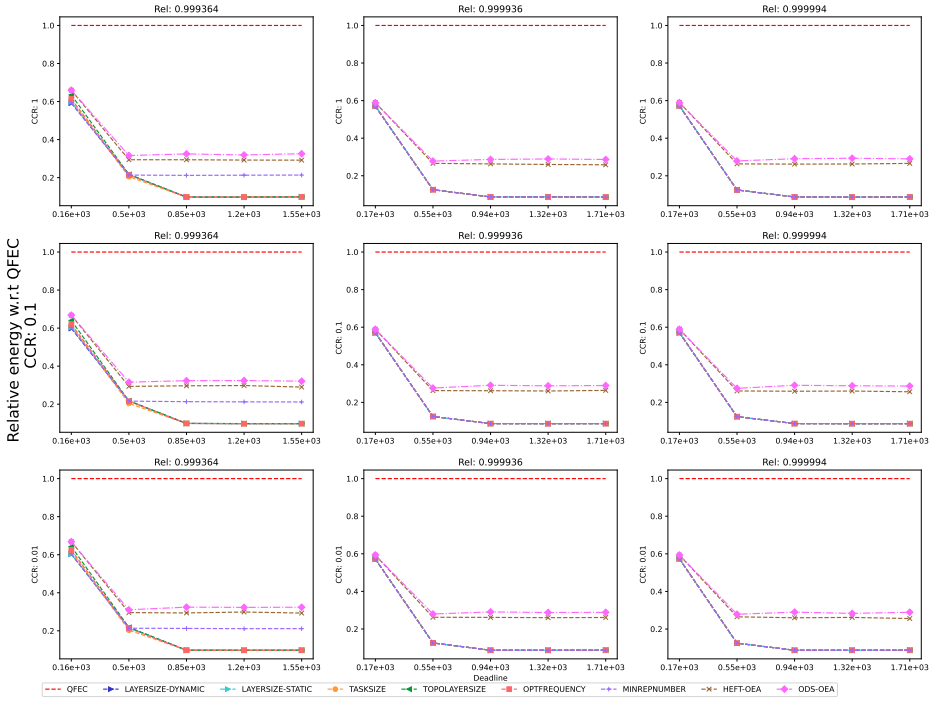

Figure 19: Performance of the different heuristics on the Seismology workflow.

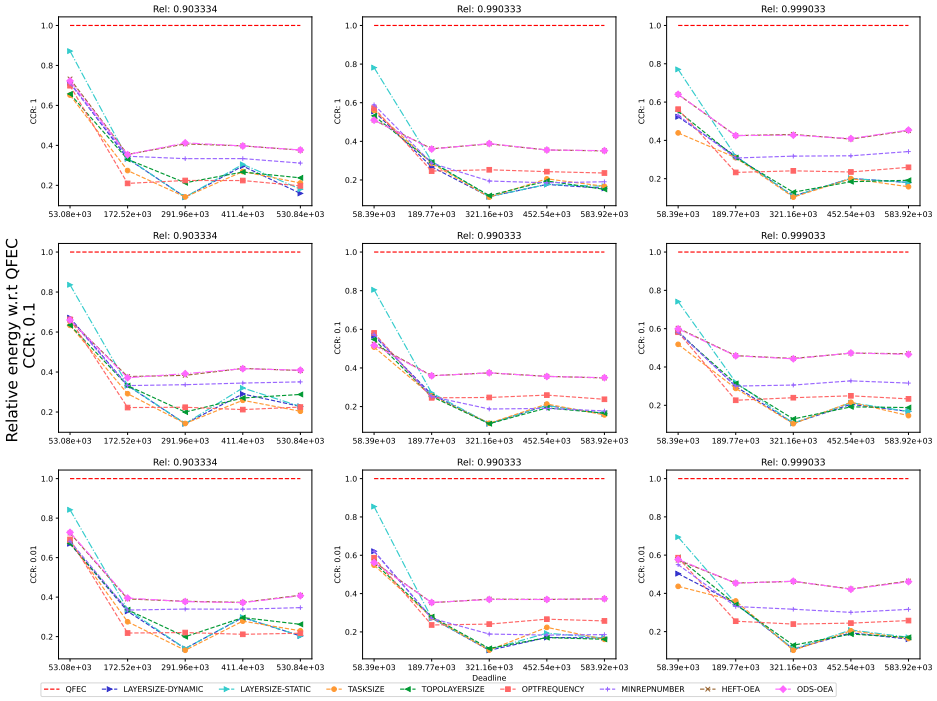

Figure 20: Performance of the different heuristics on the SoyKB workflow.

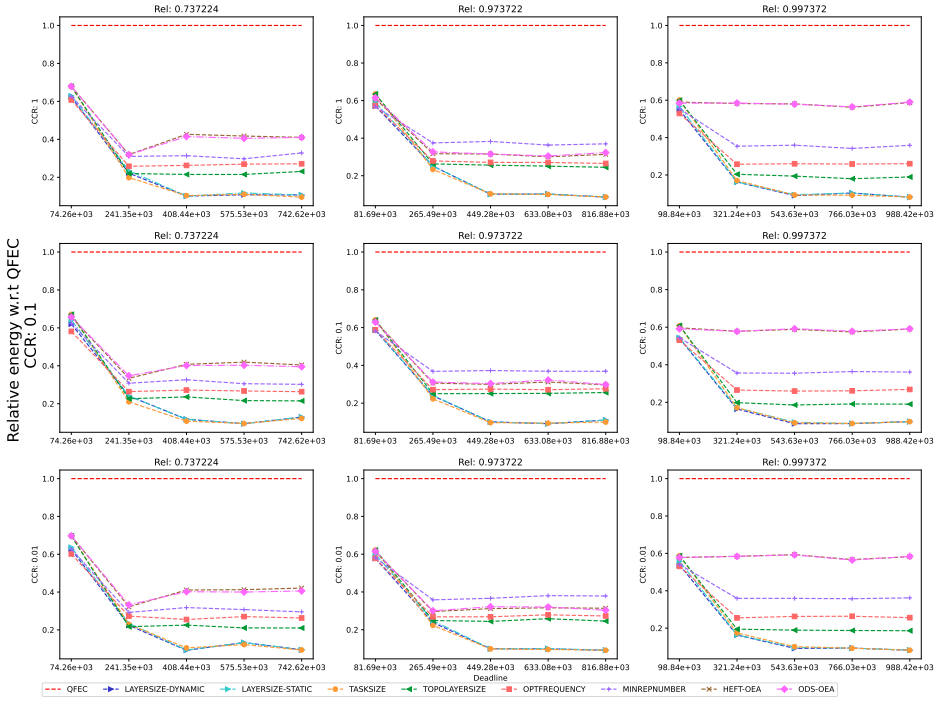

Figure 21: Performance of the different heuristics on the SRASearch.

D.1.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

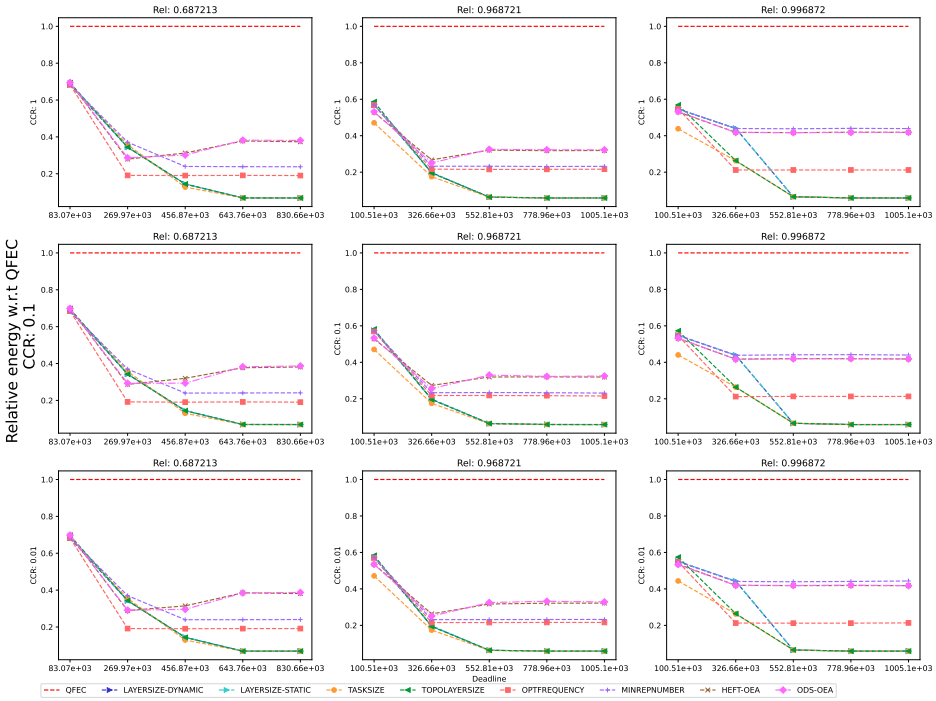

Figure 22: Performance of the different heuristics on the BLAST workflow.

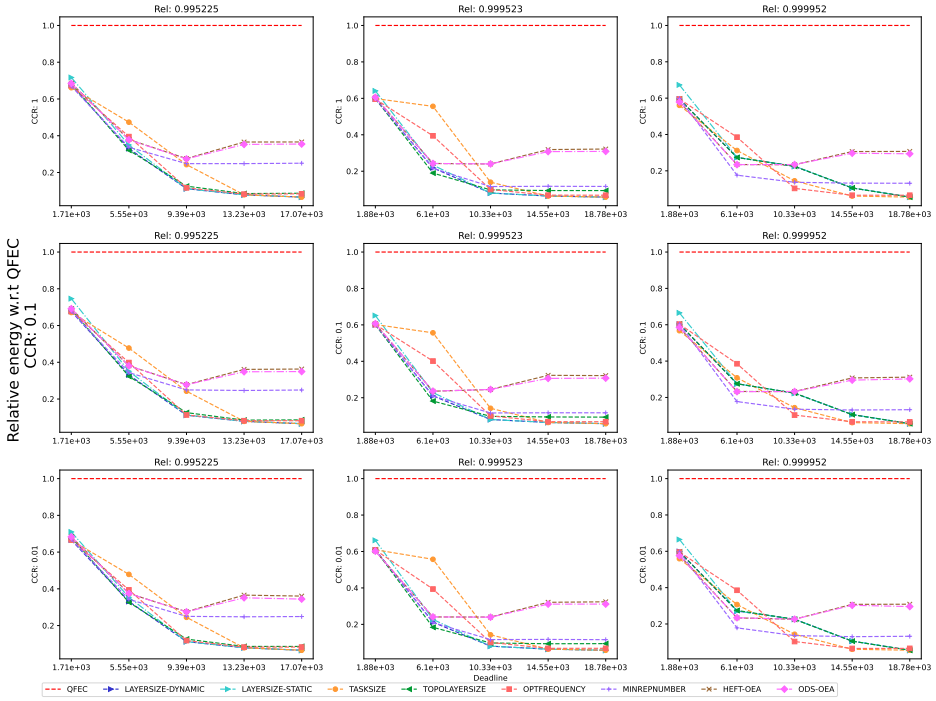

Figure 23: Performance of the different heuristics on the BWA workflow.

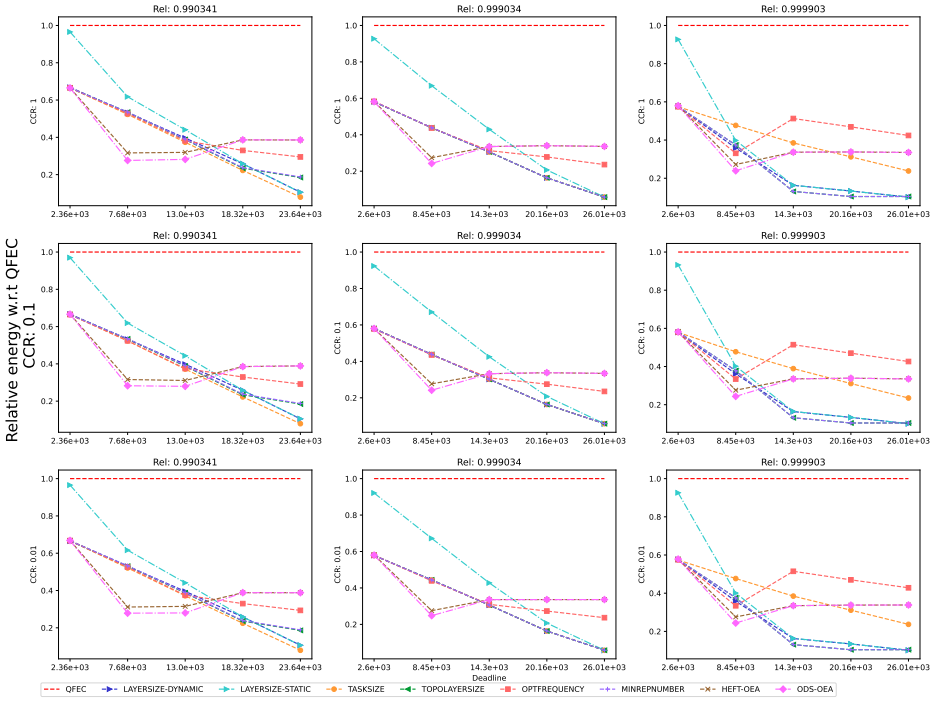

Figure 24: Performance of the different heuristics on the Cholesky workflow.

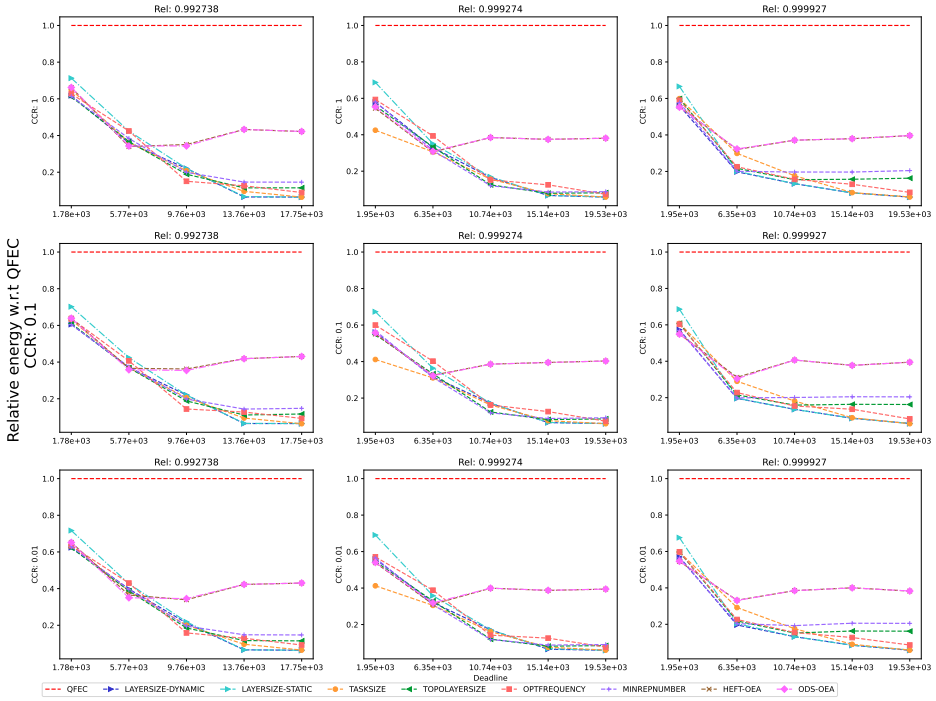

Figure 25: Performance of the different heuristics on the Cycles workflow.

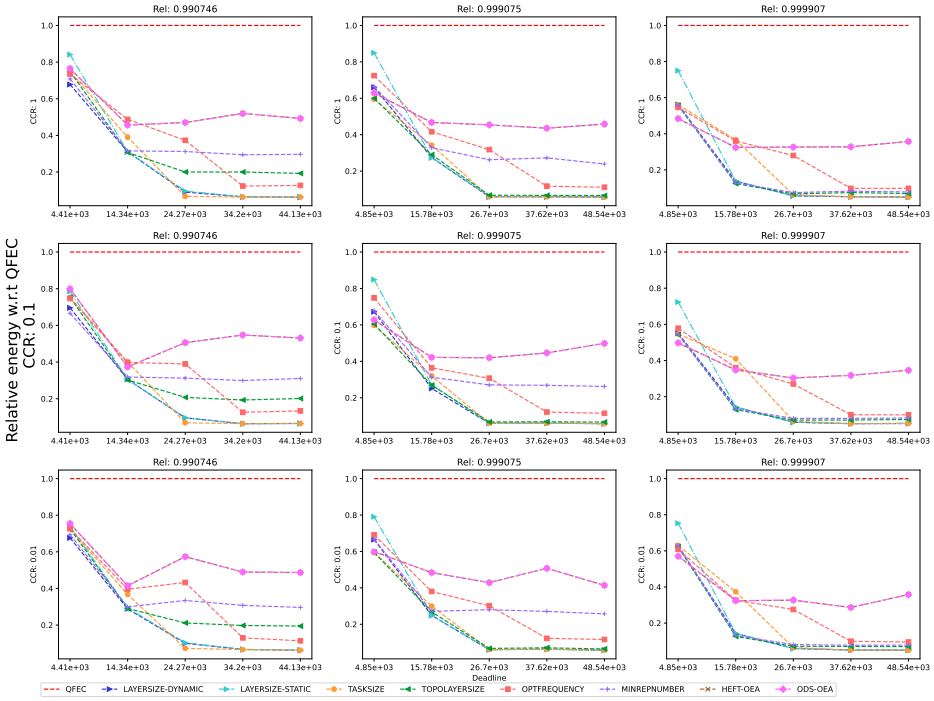

Figure 26: Performance of the different heuristics on the Epigenomics workflow.

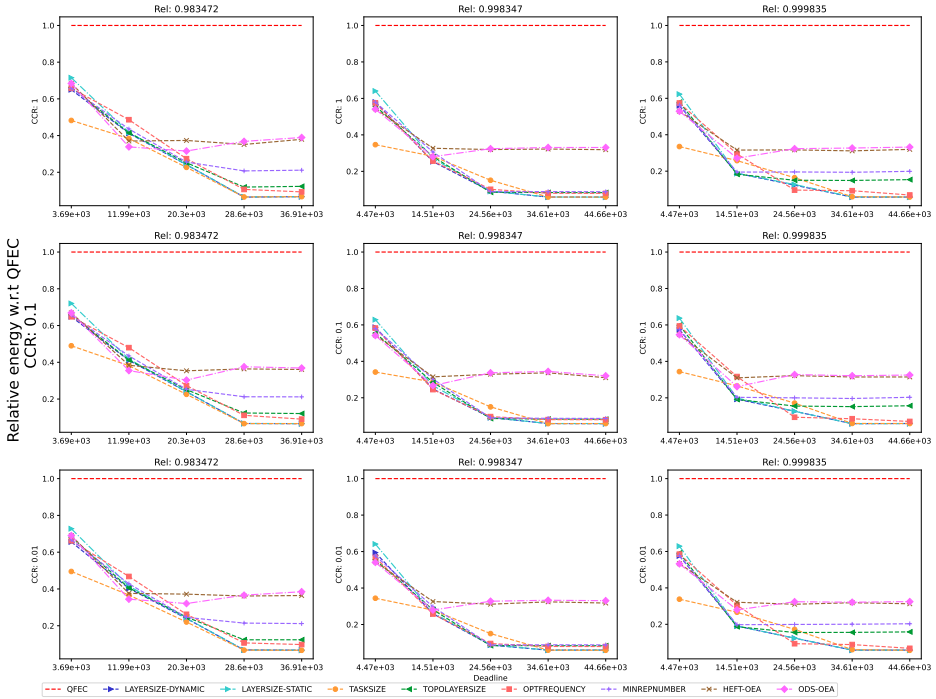

Figure 27: Performance of the different heuristics on the Genome workflow.

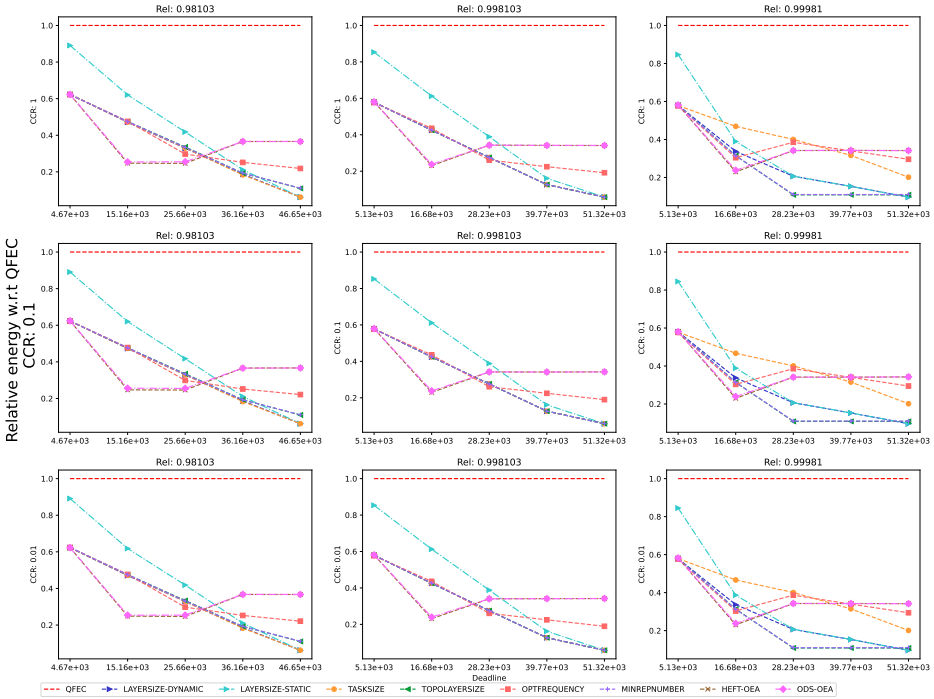

Figure 28: Performance of the different heuristics on the LU workflow.

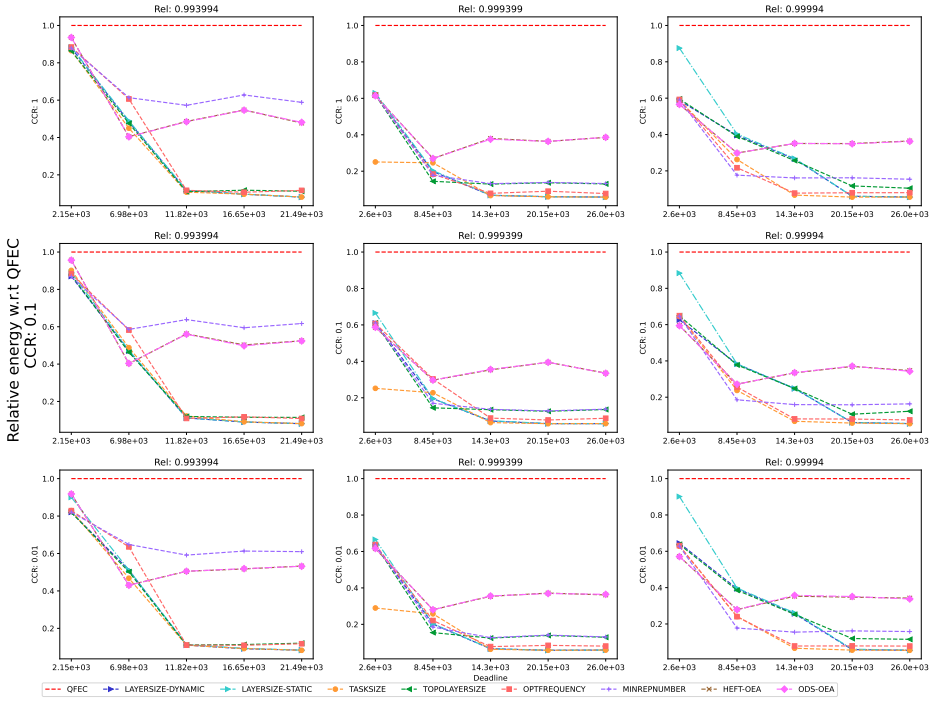

Figure 29: Performance of the different heuristics on the Montage workflow.

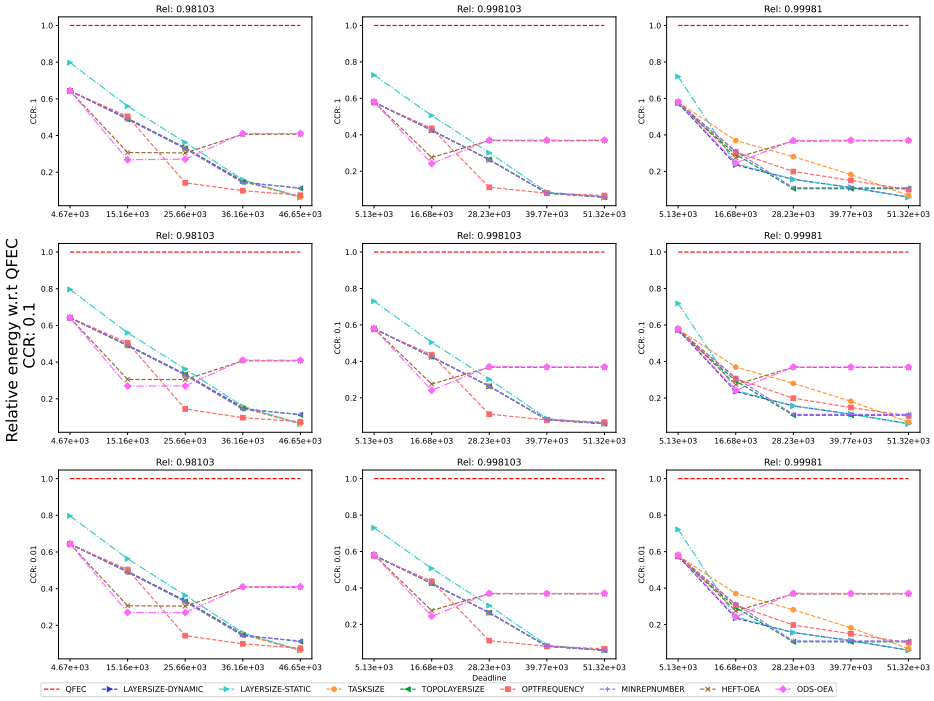

Figure 30: Performance of the different heuristics on the QR workflow.

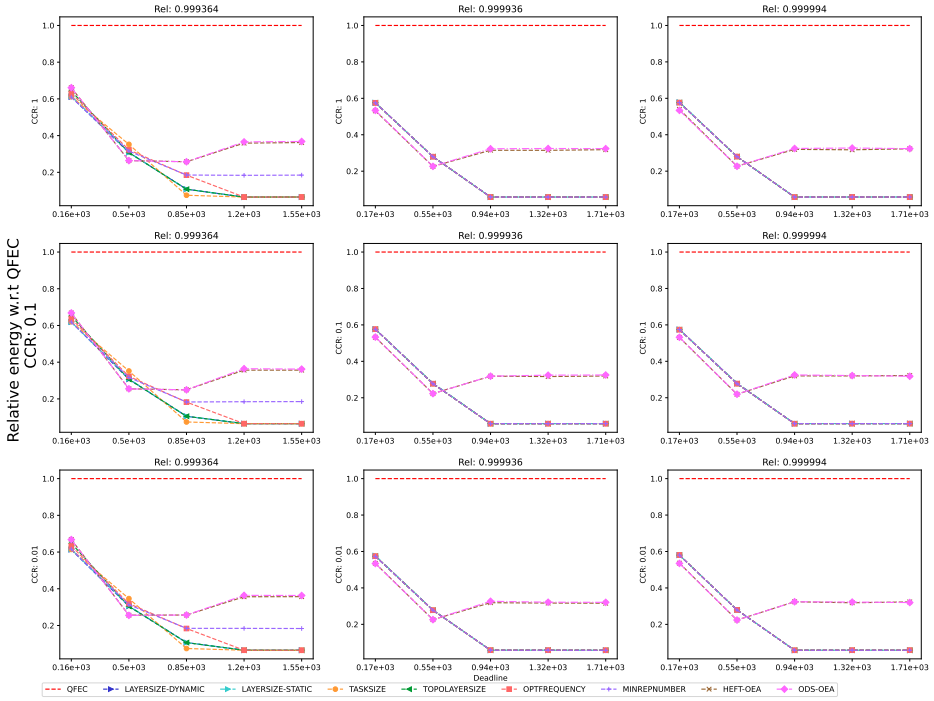

Figure 31: Performance of the different heuristics on the Seismology workflow.

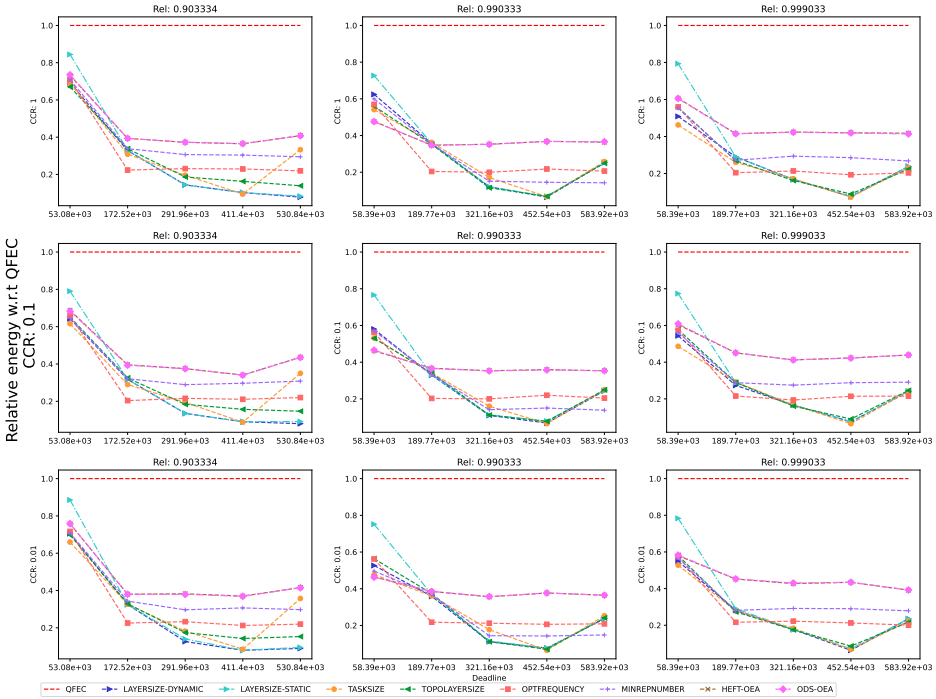

Figure 32: Performance of the different heuristics on the SoyKB workflow.

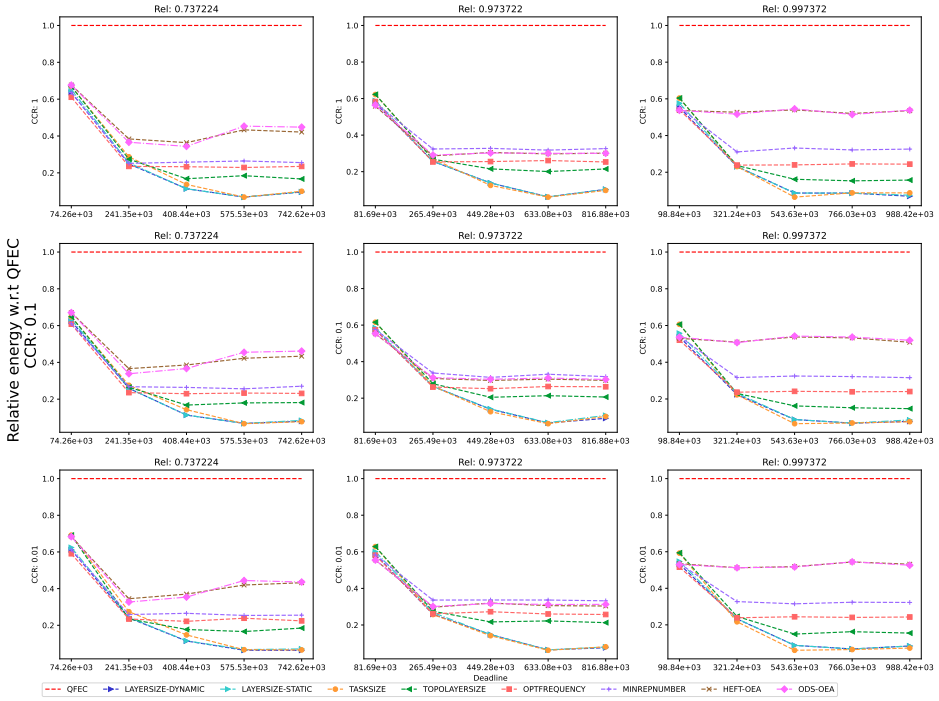

Figure 33: Performance of the different heuristics on the SRASearch.

D.1.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

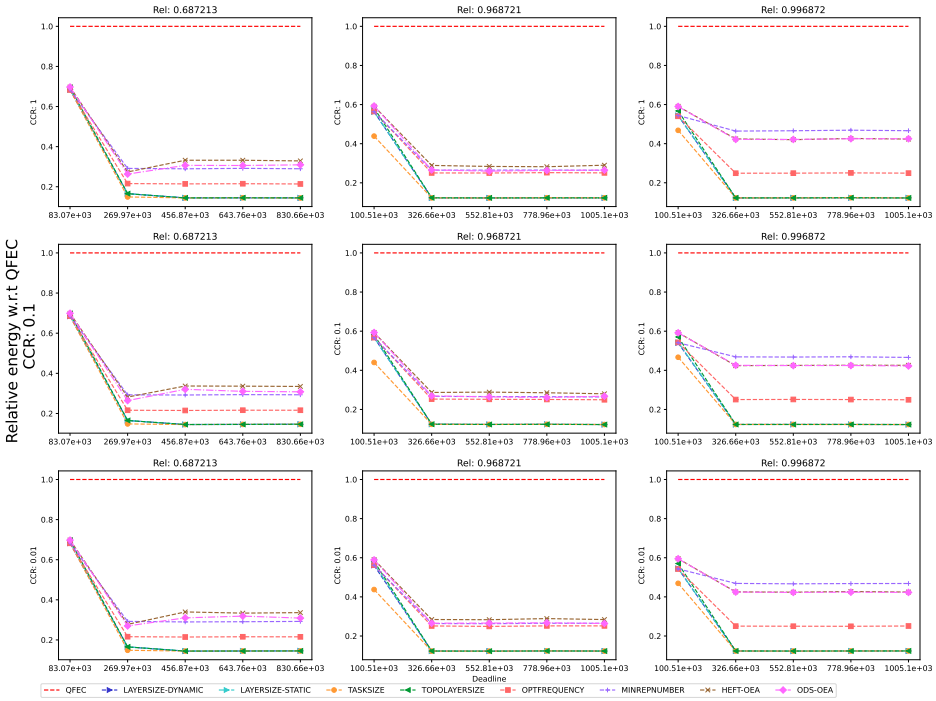

Figure 34: Performance of the different heuristics on the BLAST workflow.

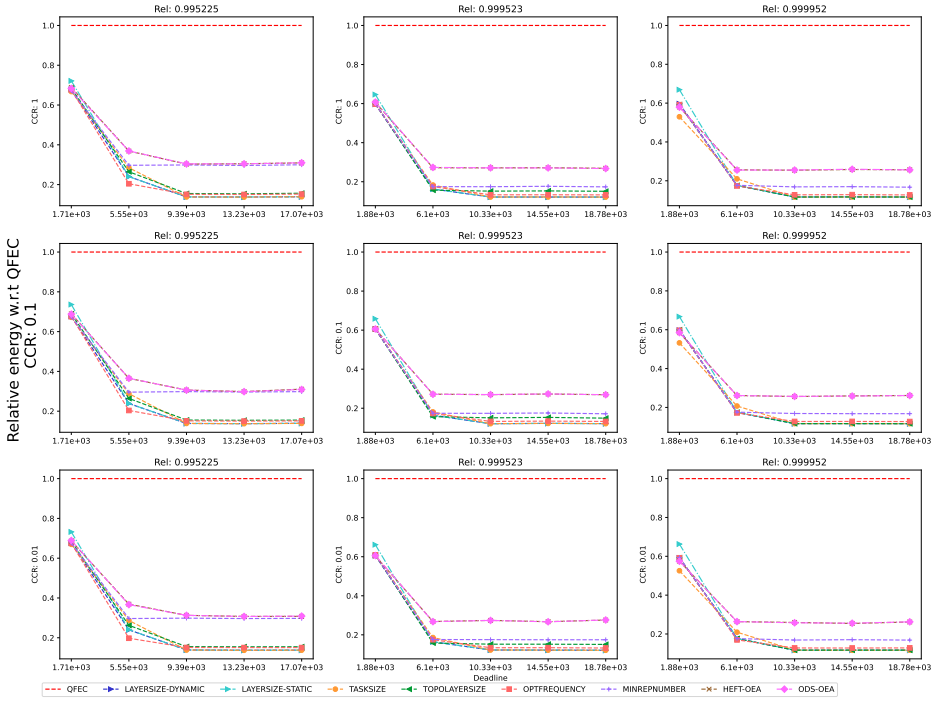

Figure 35: Performance of the different heuristics on the BWA workflow.

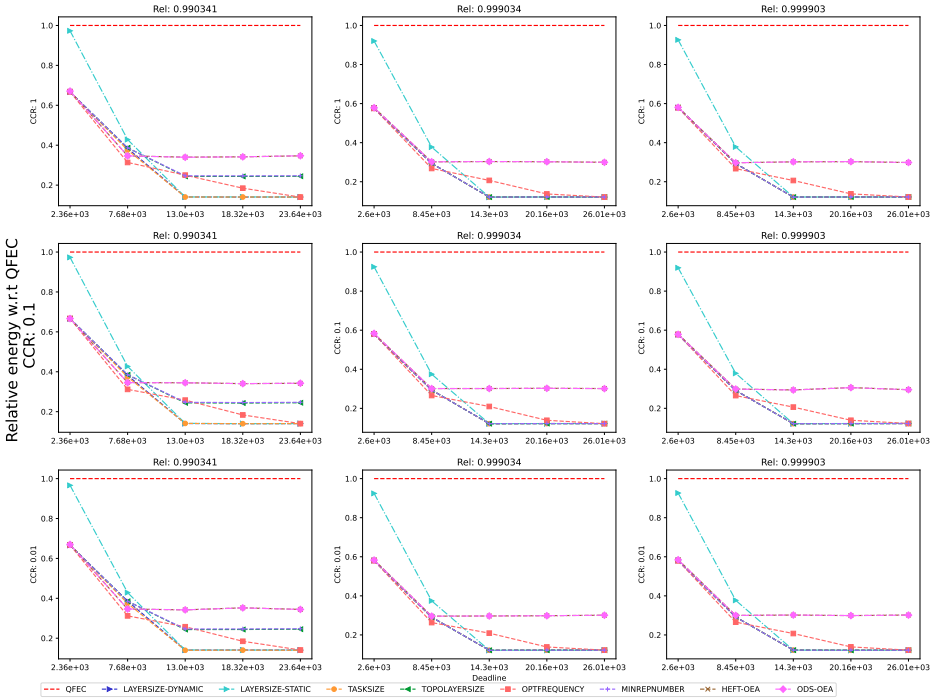

Figure 36: Performance of the different heuristics on the Cholesky workflow.

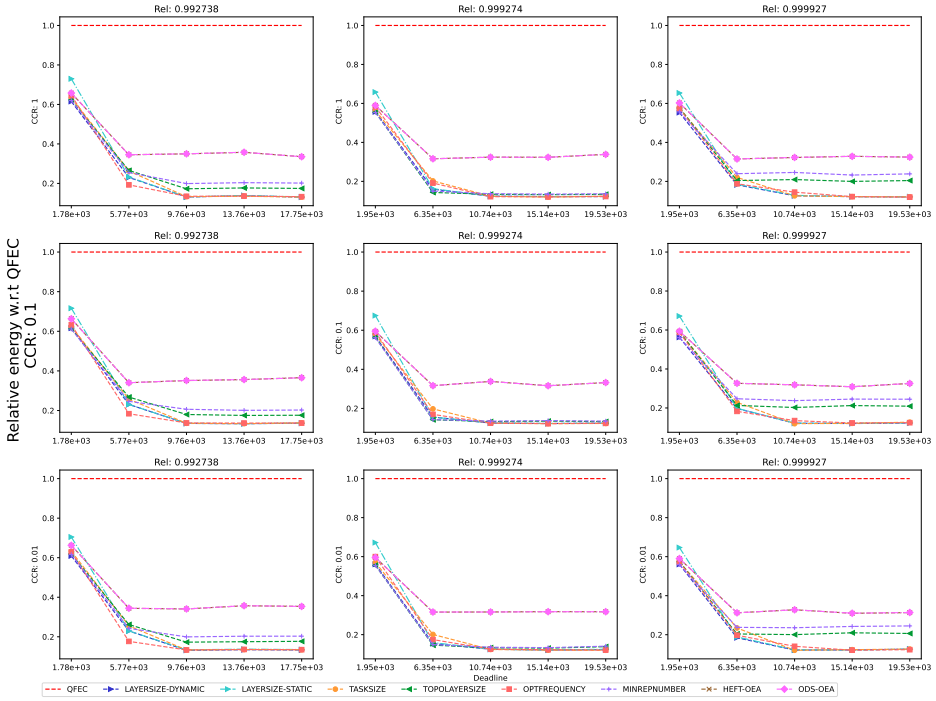

Figure 37: Performance of the different heuristics on the Cycles workflow.

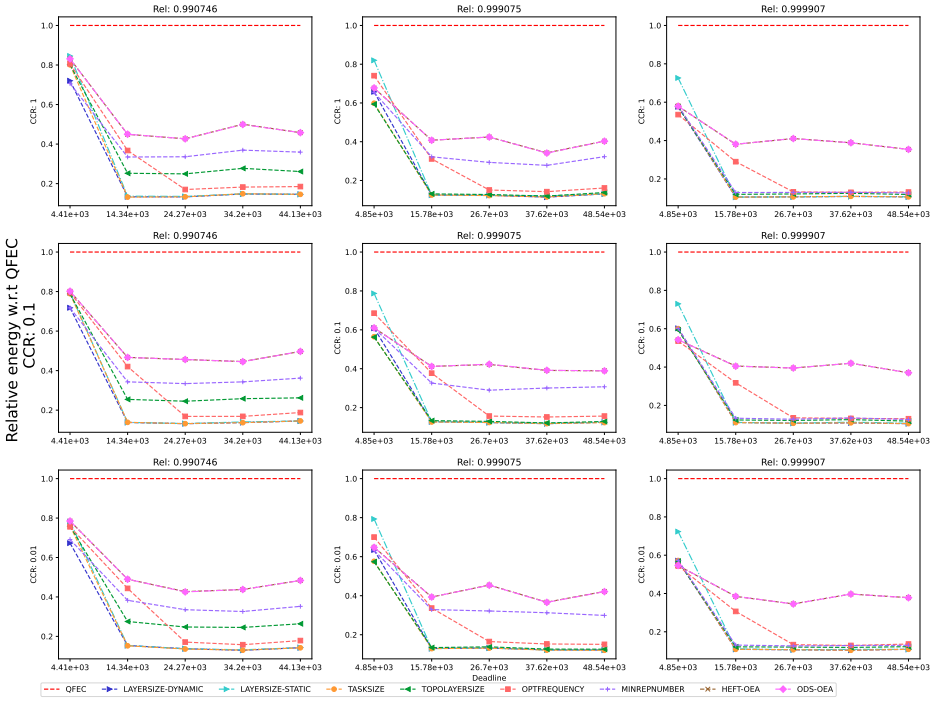

Figure 38: Performance of the different heuristics on the Epigenomics workflow.

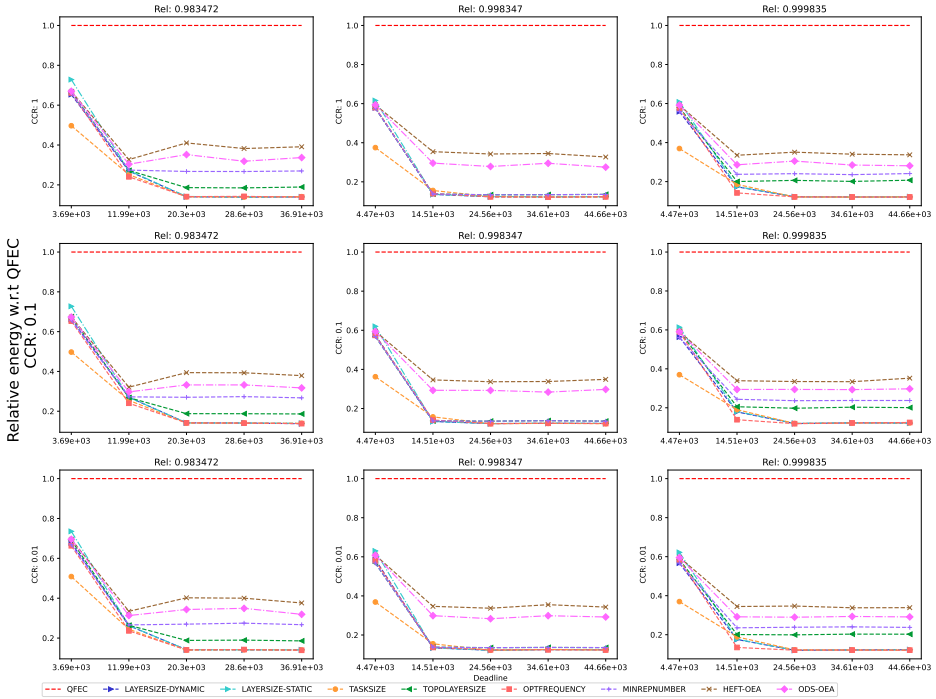

Figure 39: Performance of the different heuristics on the Genome workflow.

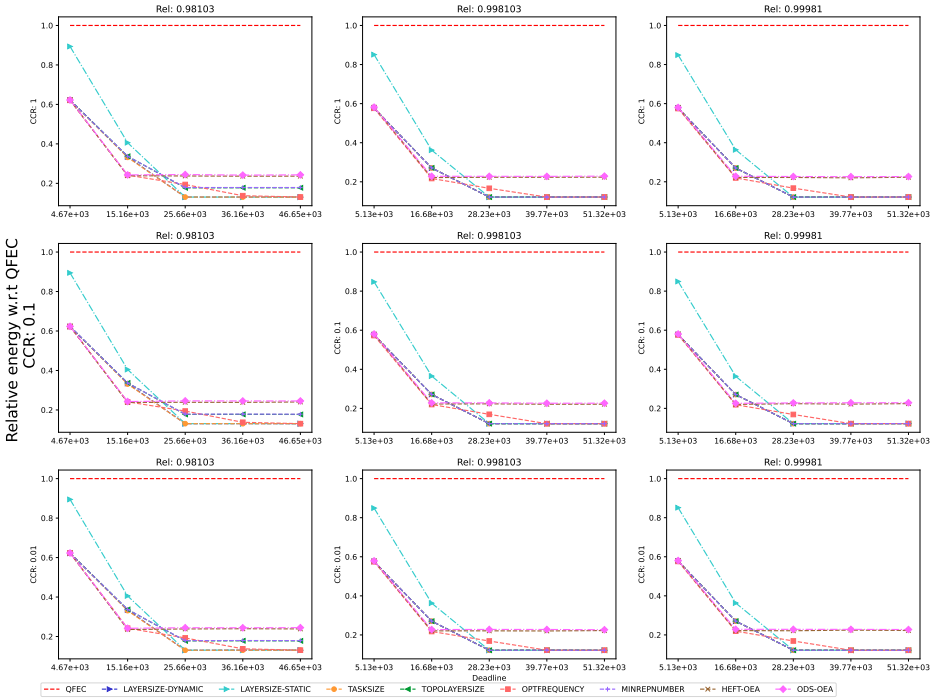

Figure 40: Performance of the different heuristics on the LU workflow.

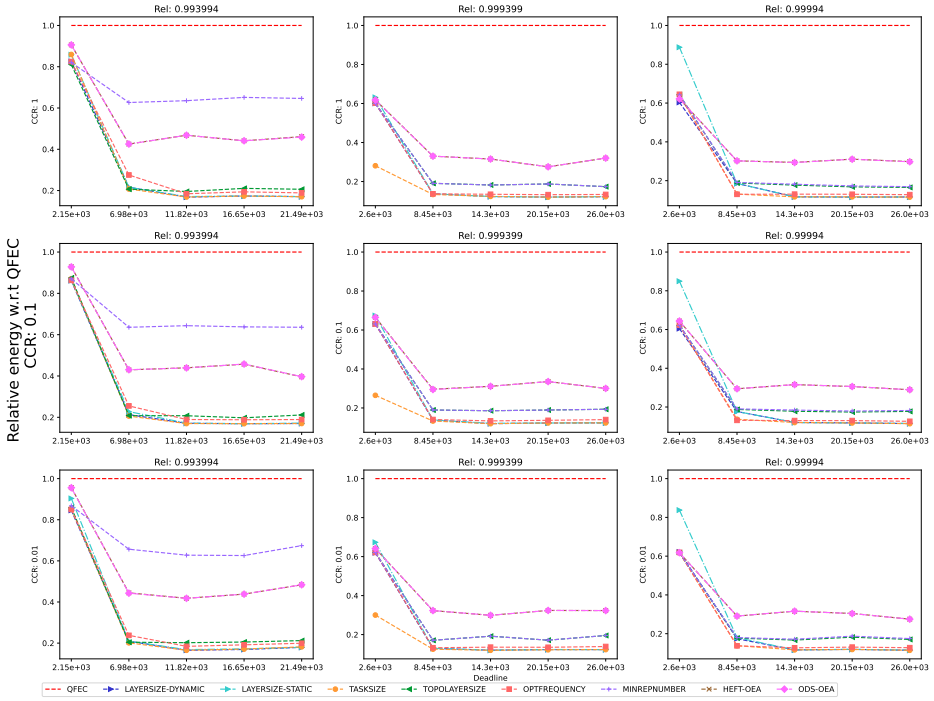

Figure 41: Performance of the different heuristics on the Montage workflow.

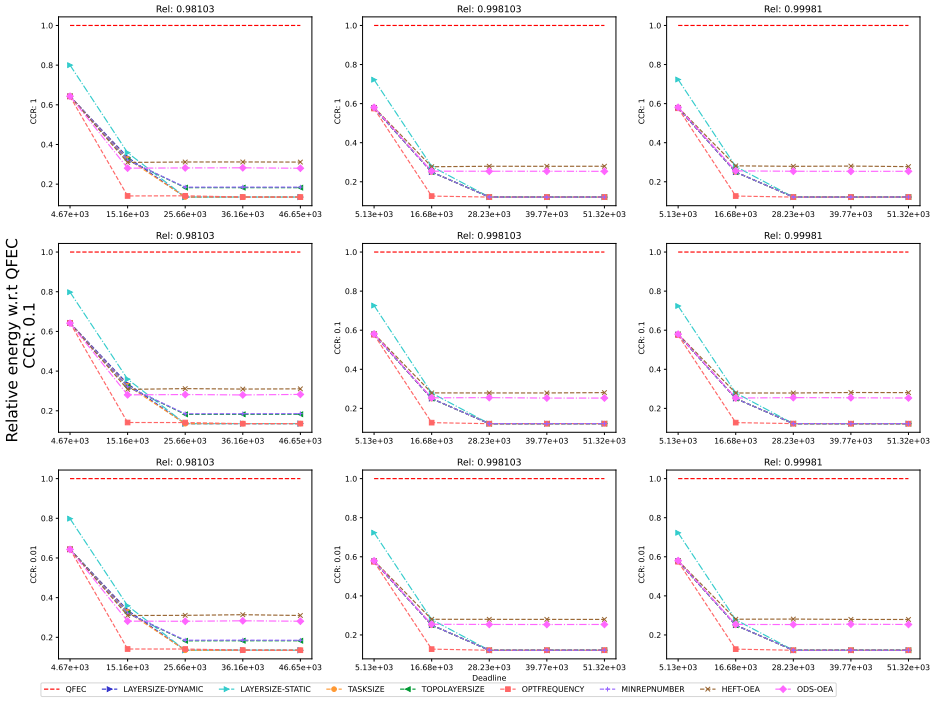

Figure 42: Performance of the different heuristics on the QR workflow.

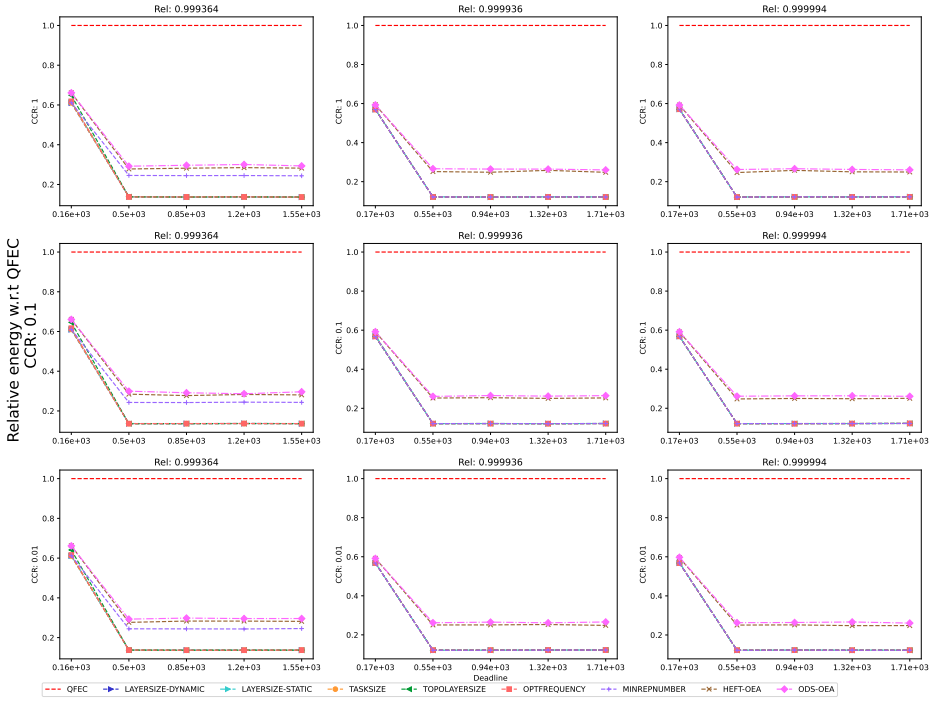

Figure 43: Performance of the different heuristics on the Seismology workflow.

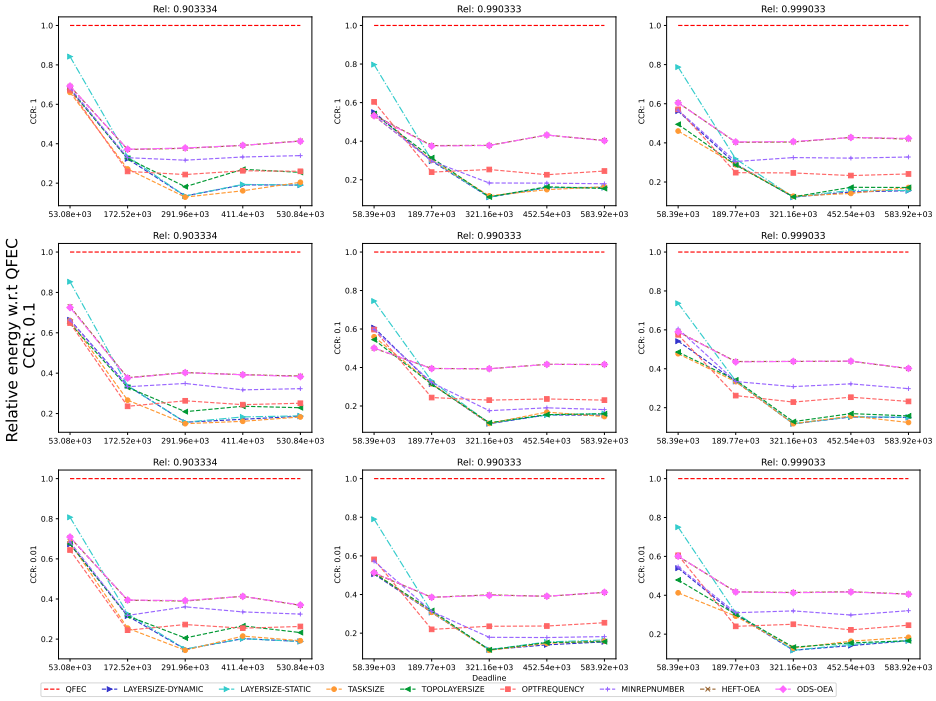

Figure 44: Performance of the different heuristics on the SoyKB workflow.

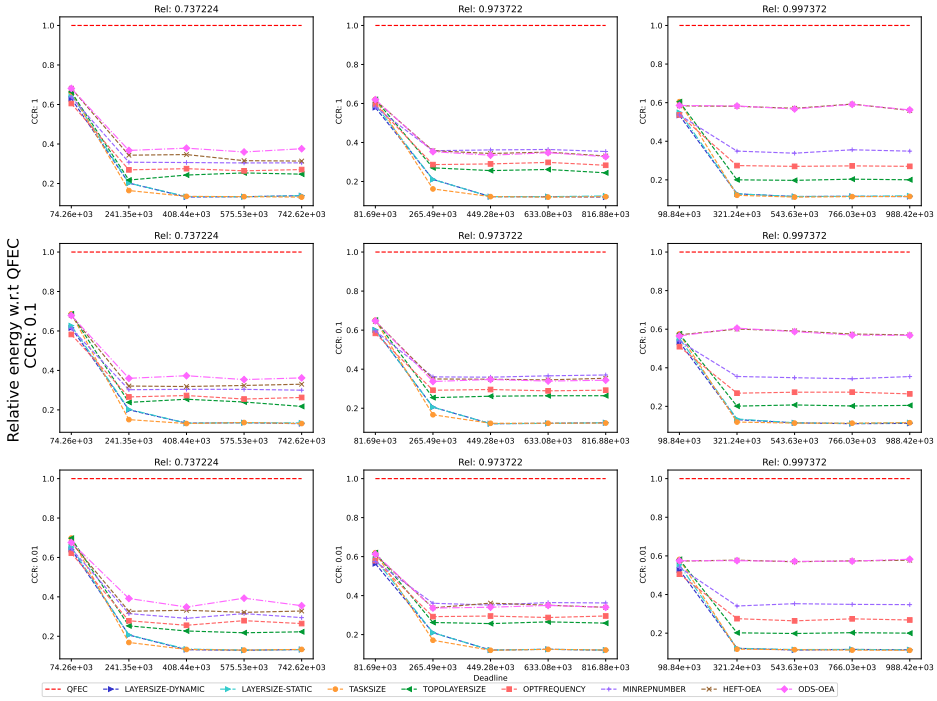

Figure 45: Performance of the different heuristics on the SRASearch.

D.1.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

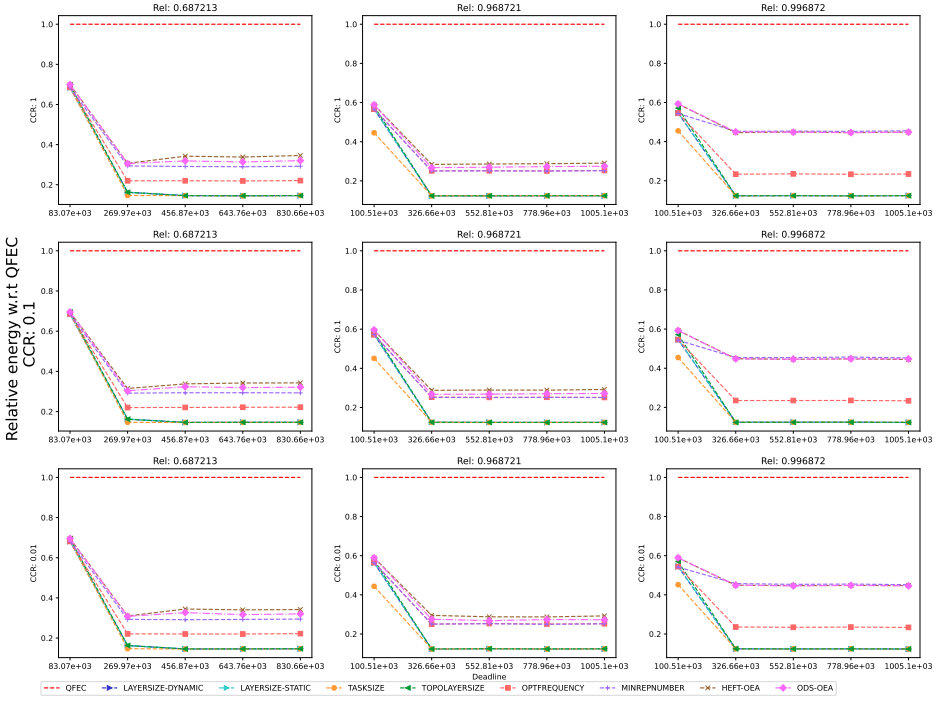

Figure 46: Performance of the different heuristics on the BLAST workflow.

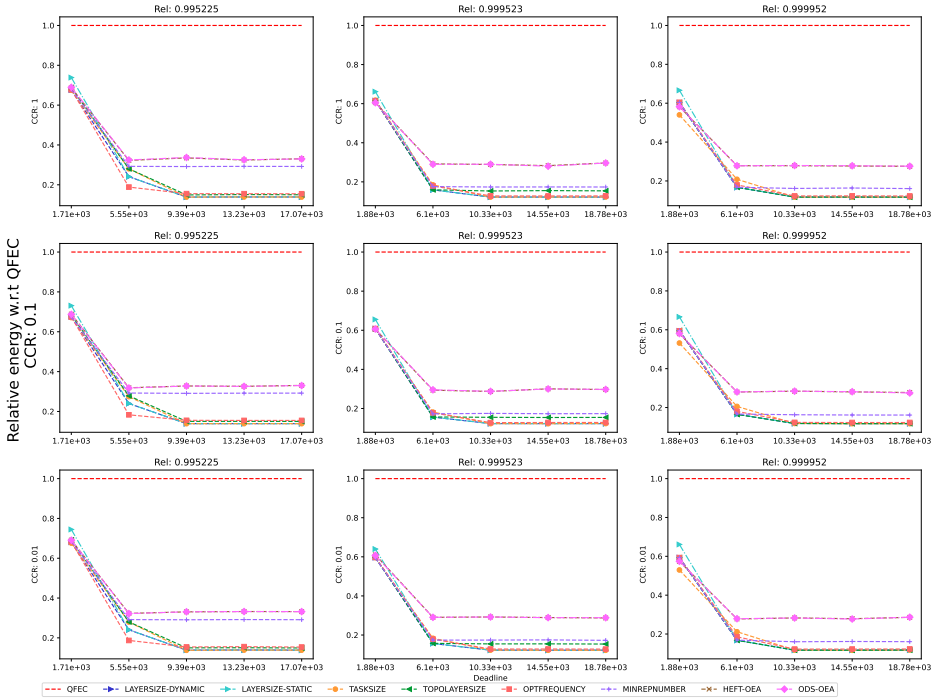

Figure 47: Performance of the different heuristics on the BWA workflow.

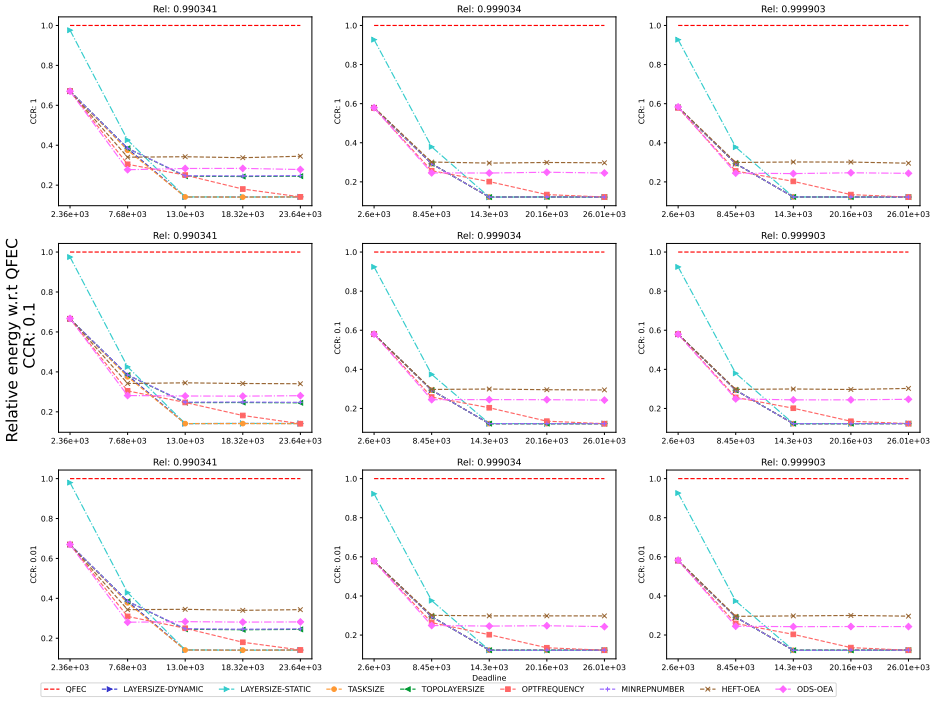

Figure 48: Performance of the different heuristics on the Cholesky workflow.

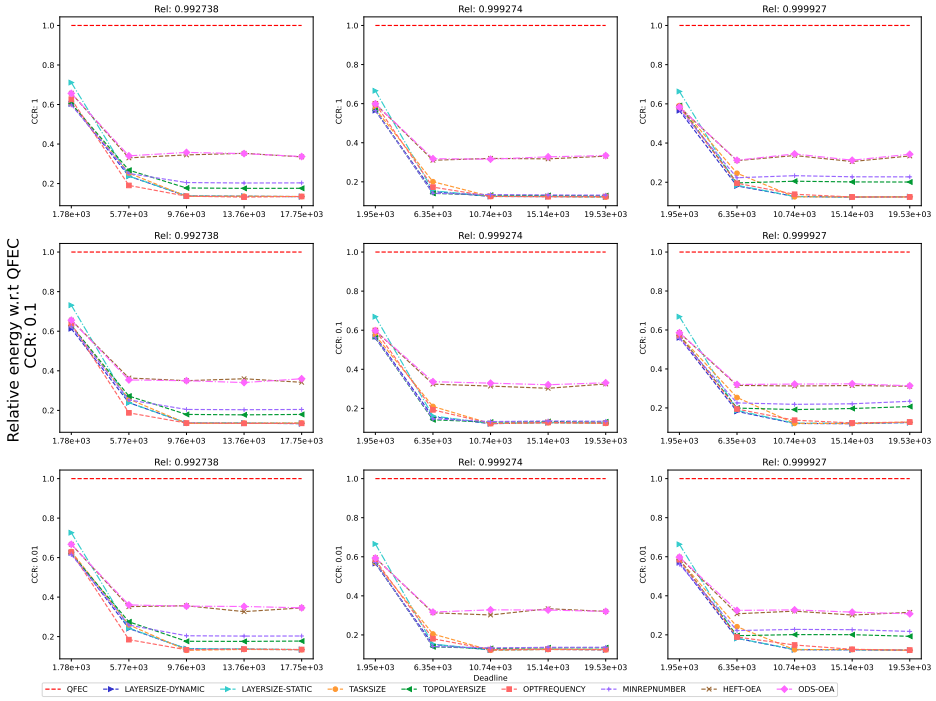

Figure 49: Performance of the different heuristics on the Cycles workflow.

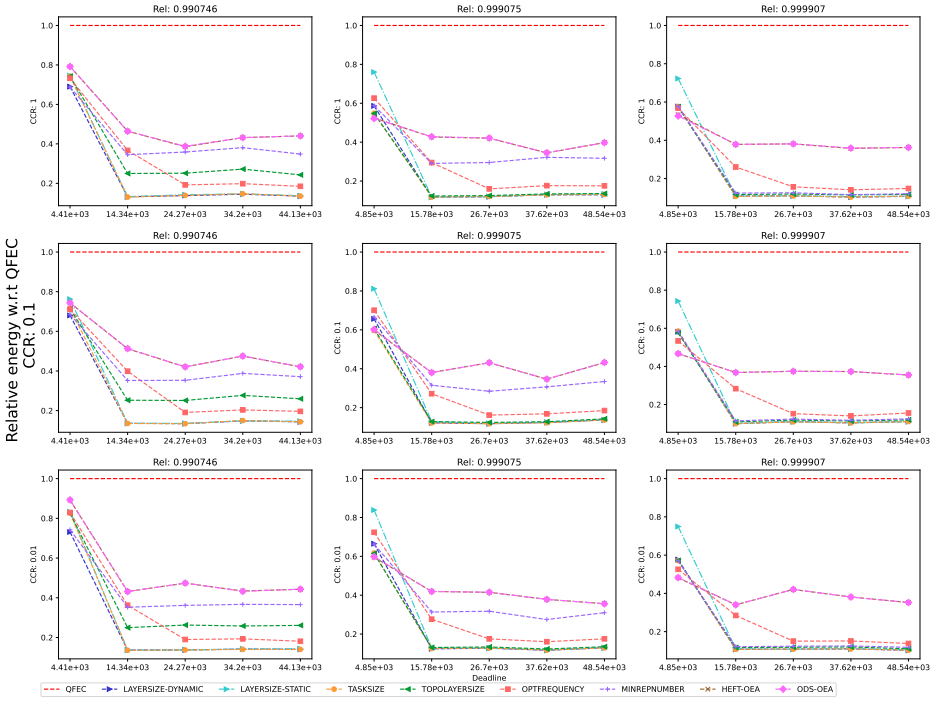

Figure 50: Performance of the different heuristics on the Epigenomics workflow.

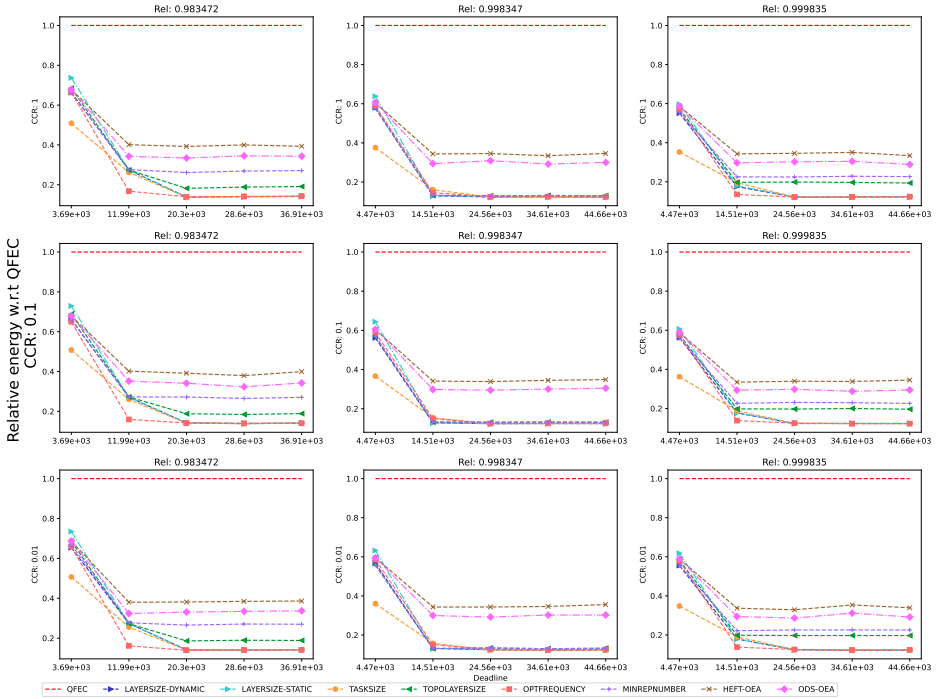

Figure 51: Performance of the different heuristics on the Genome workflow.

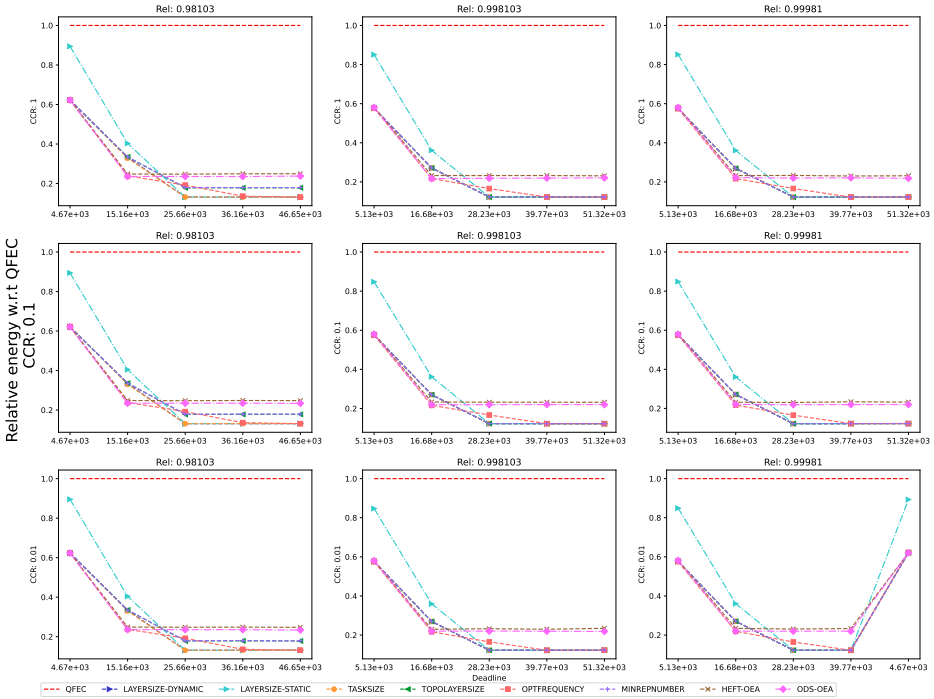

Figure 52: Performance of the different heuristics on the LU workflow.

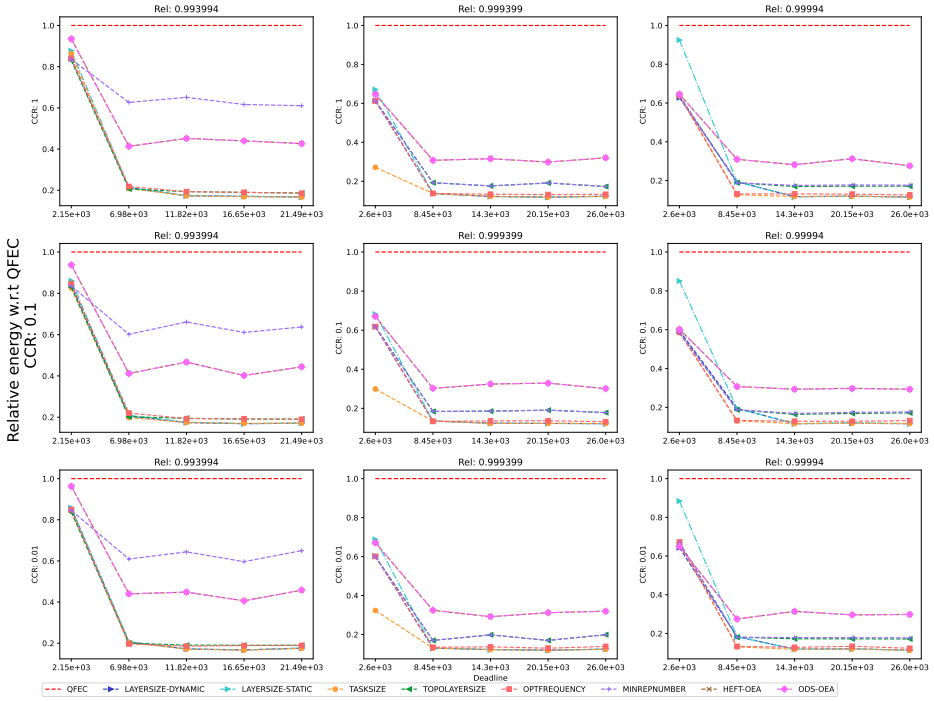

Figure 53: Performance of the different heuristics on the Montage workflow.

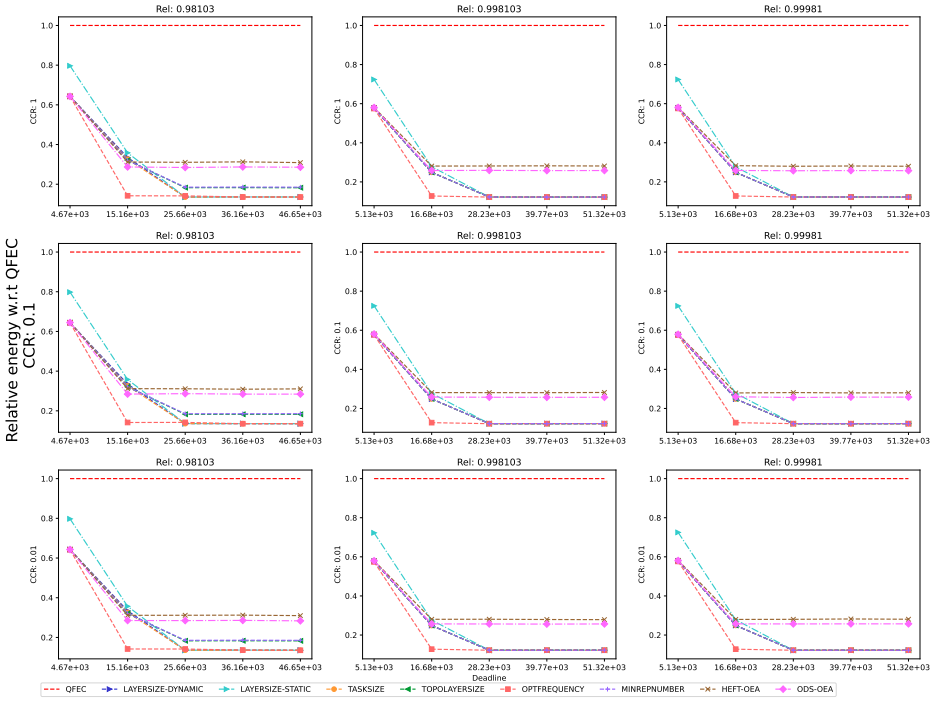

Figure 54: Performance of the different heuristics on the QR workflow.

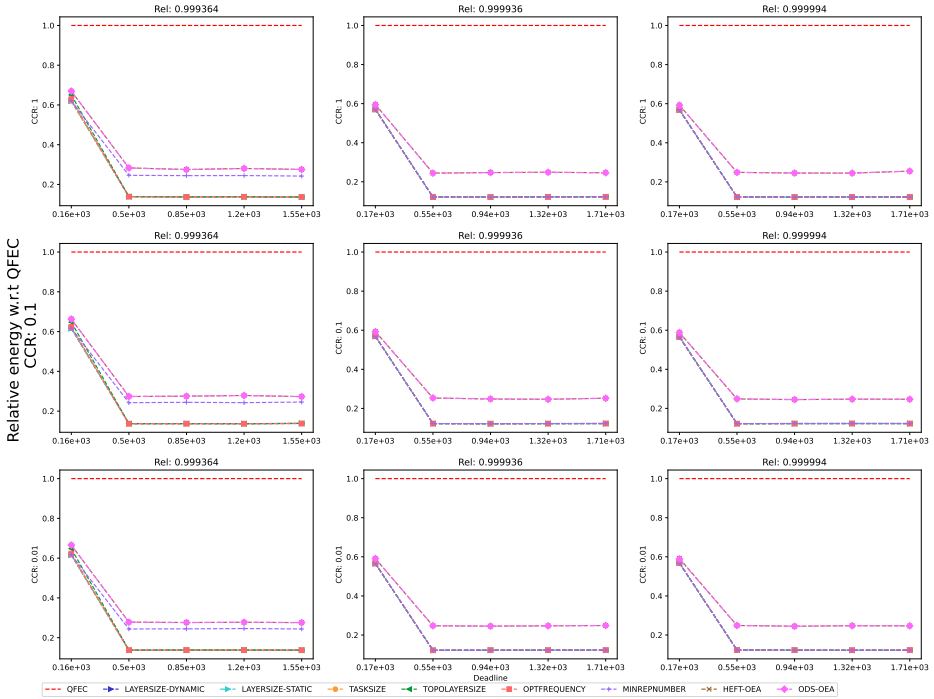

Figure 55: Performance of the different heuristics on the Seismology workflow.

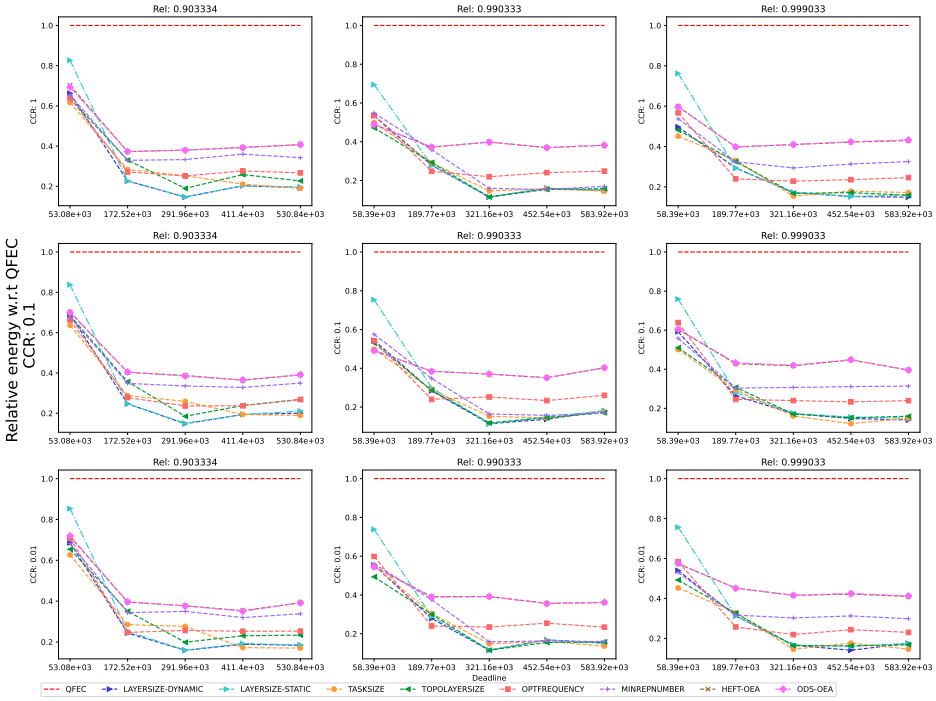

Figure 56: Performance of the different heuristics on the SoyKB workflow.

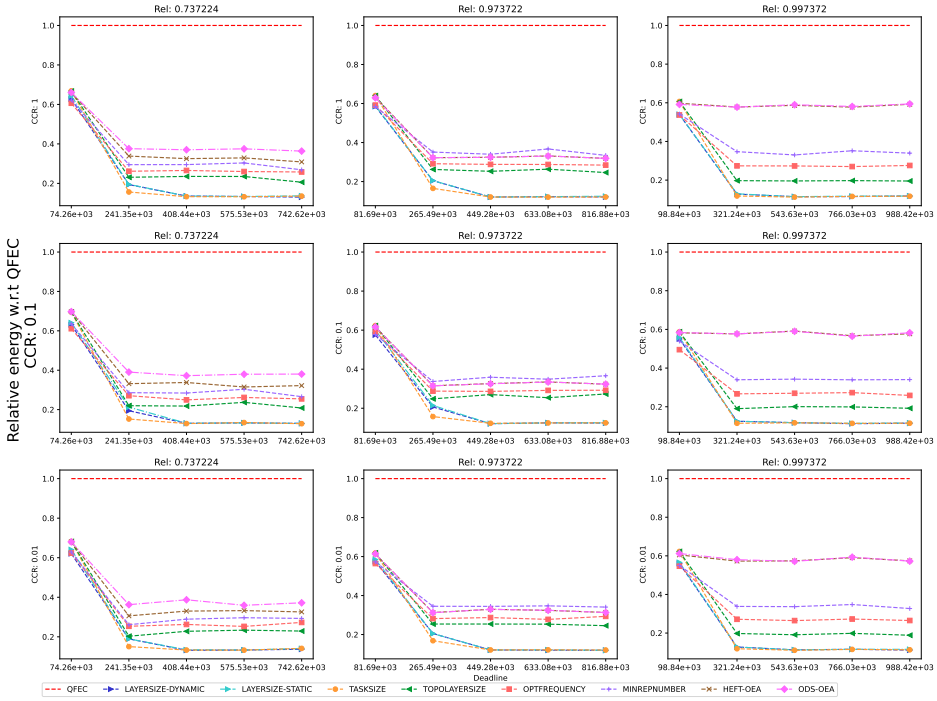

Figure 57: Performance of the different heuristics on the SRASearch.

D.2 $BC/WC = 0.2$

D.2.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

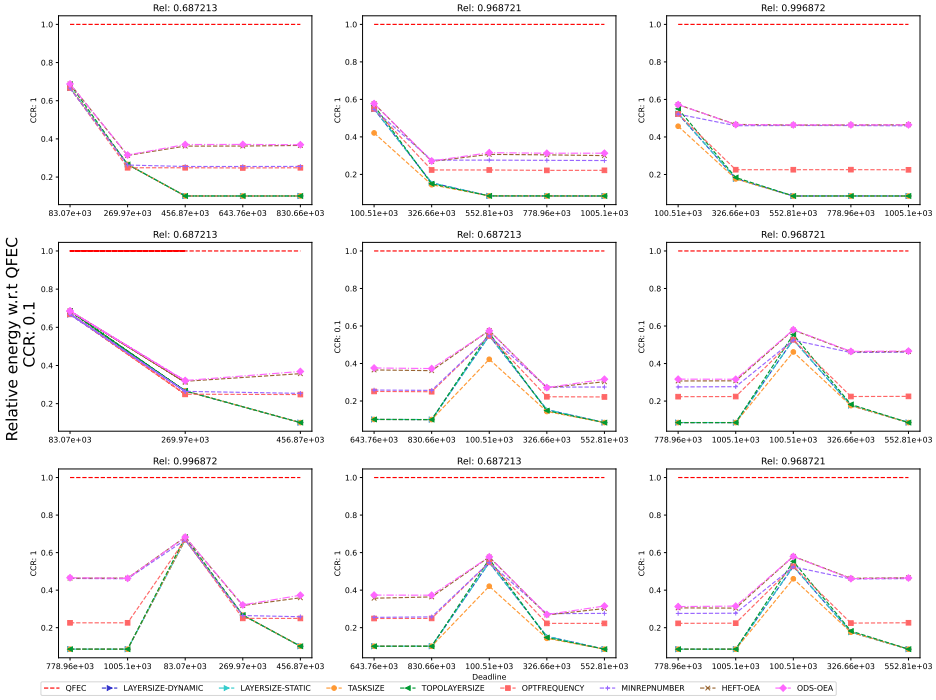

Figure 58: Performance of the different heuristics on the BLAST workflow.

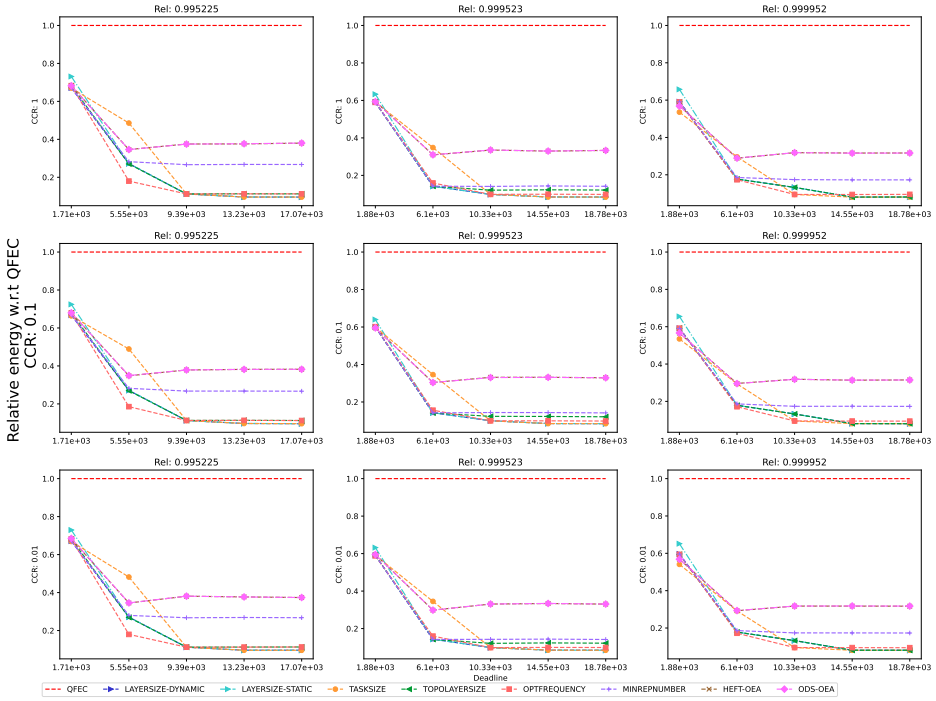

Figure 59: Performance of the different heuristics on the BWA workflow.

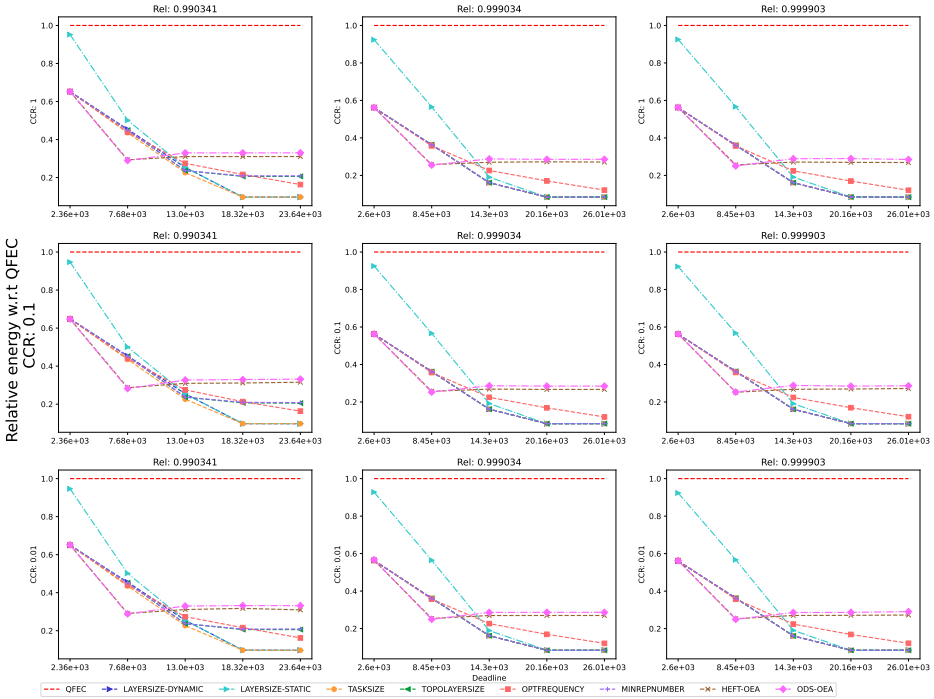

Figure 60: Performance of the different heuristics on the Cholesky workflow.

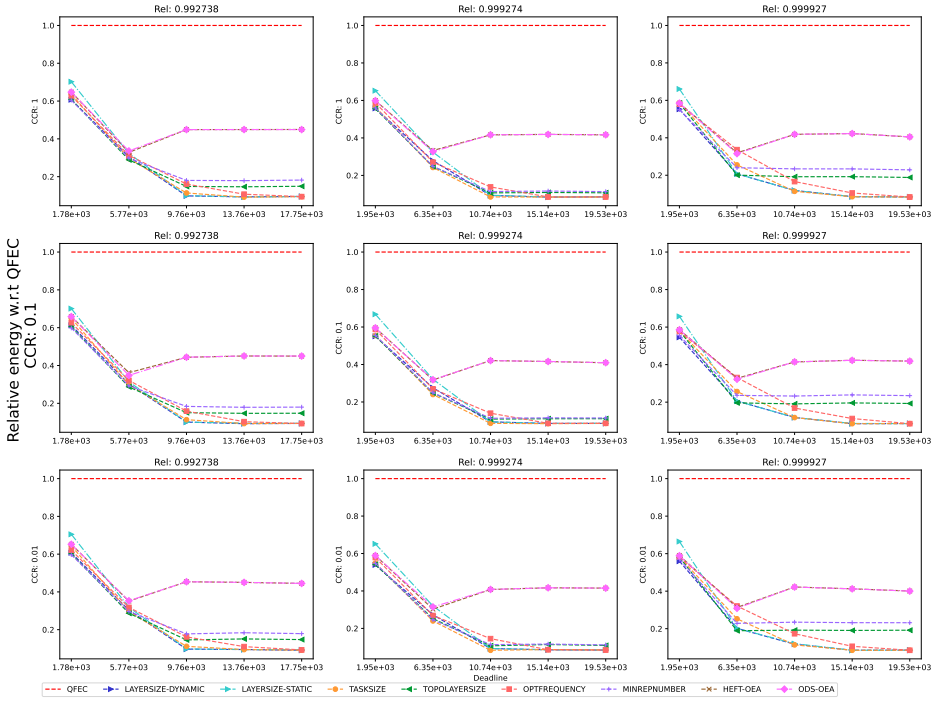

Figure 61: Performance of the different heuristics on the Cycles workflow.

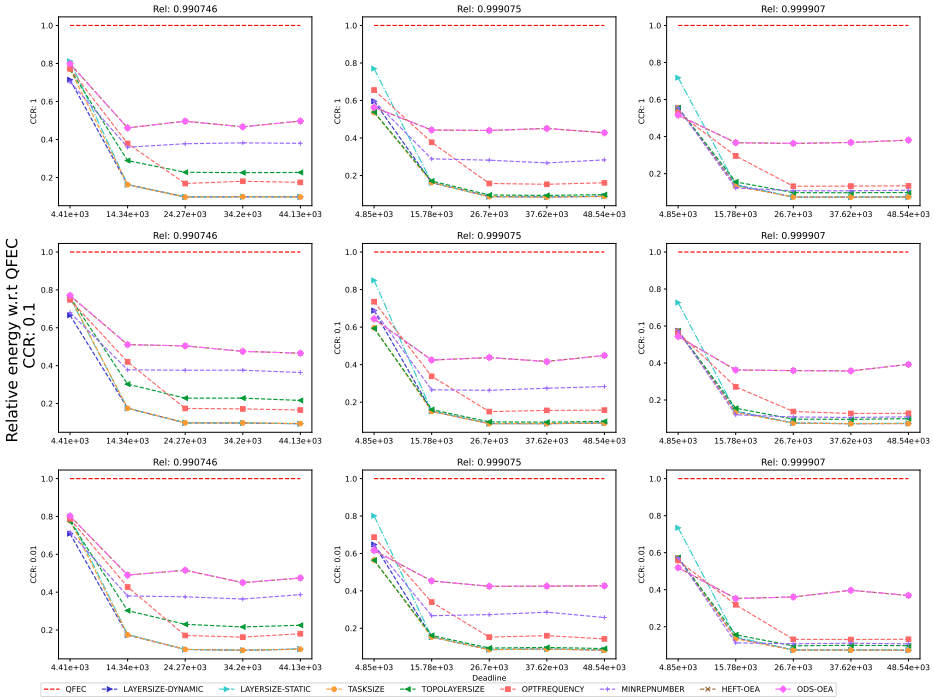

Figure 62: Performance of the different heuristics on the Epigenomics workflow.

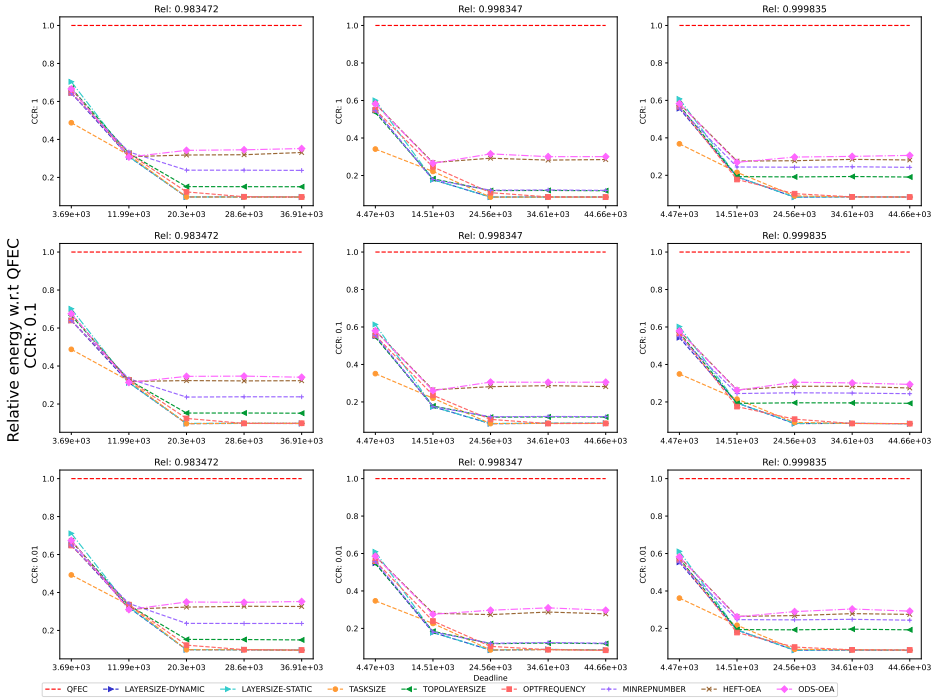

Figure 63: Performance of the different heuristics on the Genome workflow.

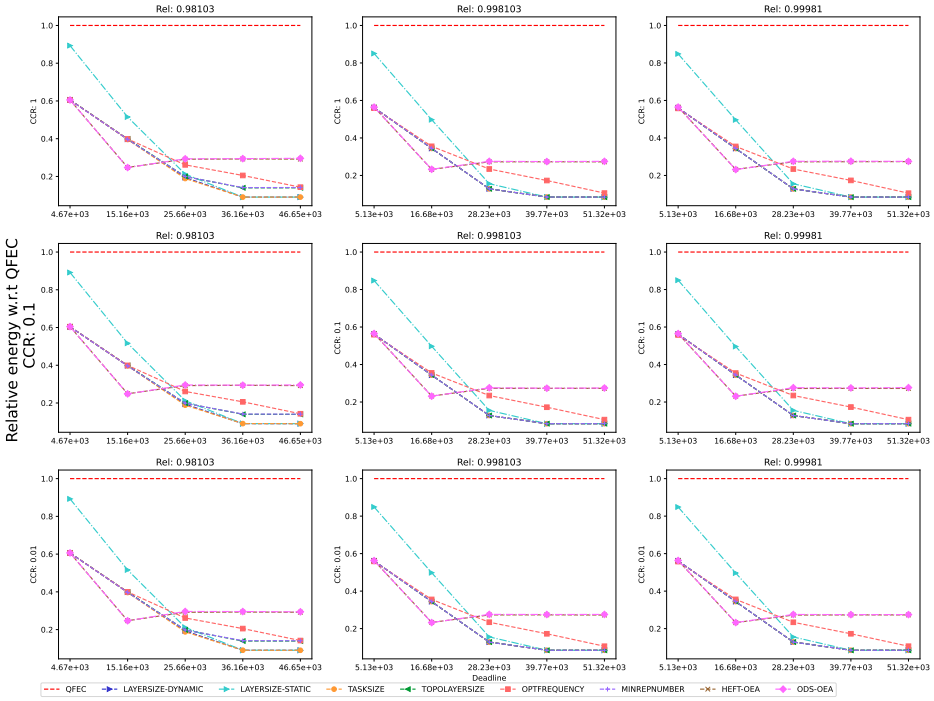

Figure 64: Performance of the different heuristics on the LU workflow.

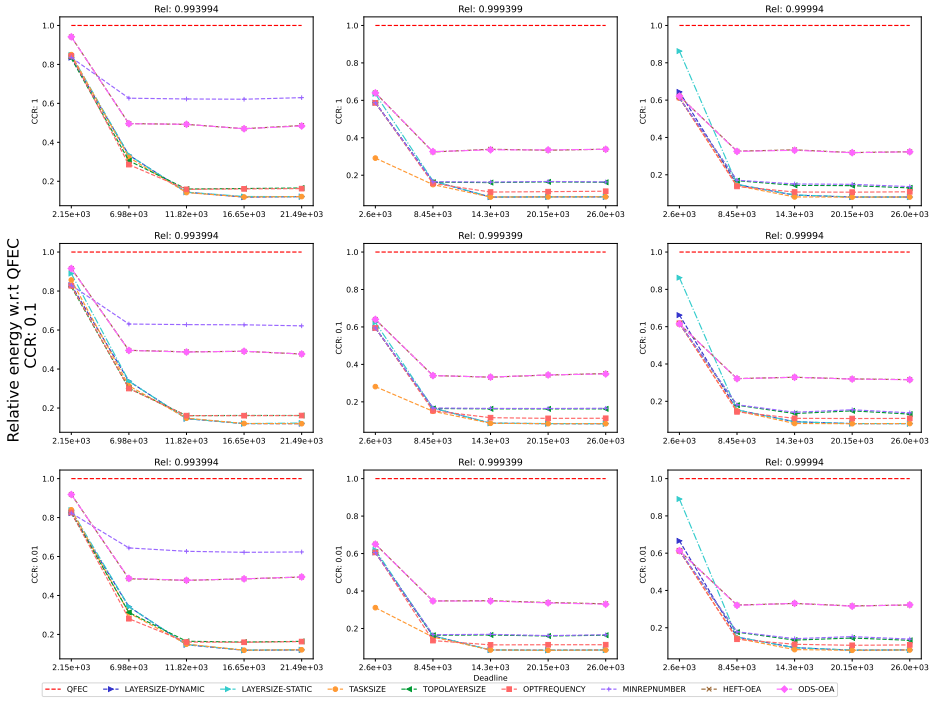

Figure 65: Performance of the different heuristics on the Montage workflow.

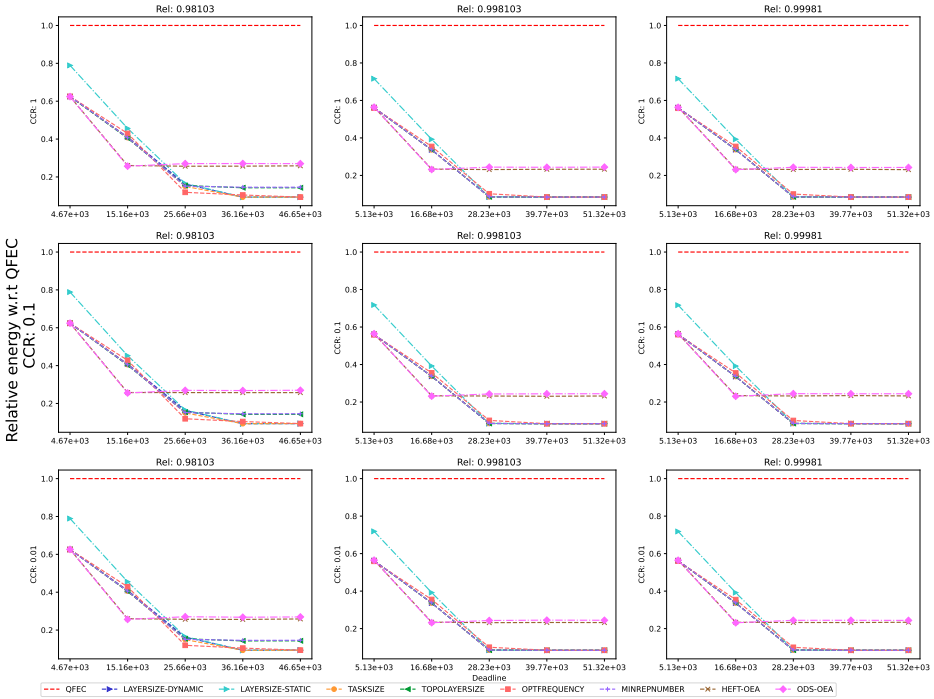

Figure 66: Performance of the different heuristics on the QR workflow.

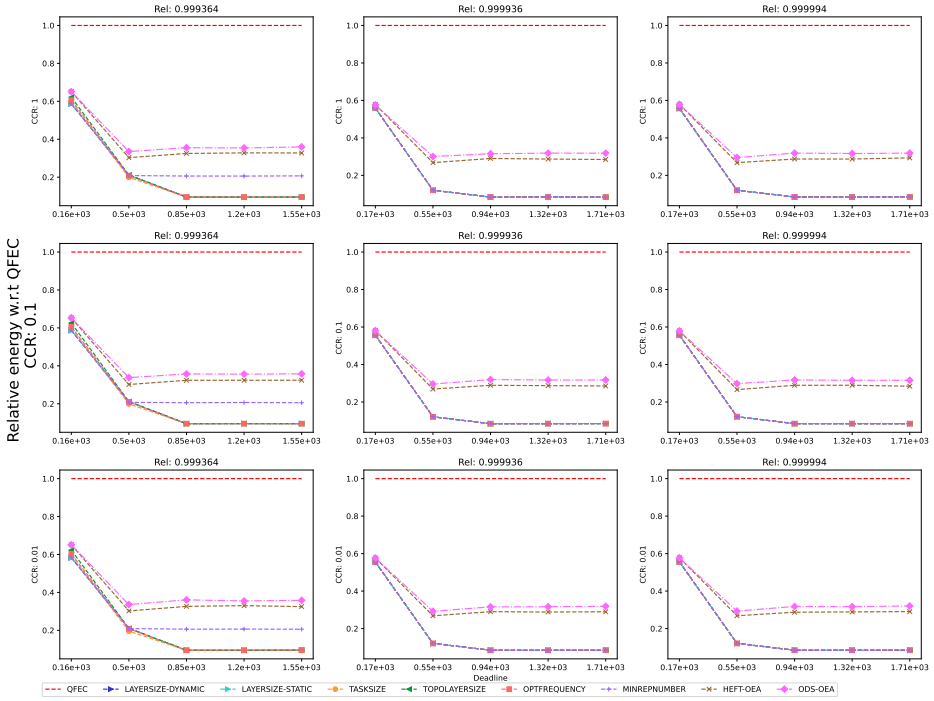

Figure 67: Performance of the different heuristics on the Seismology workflow.

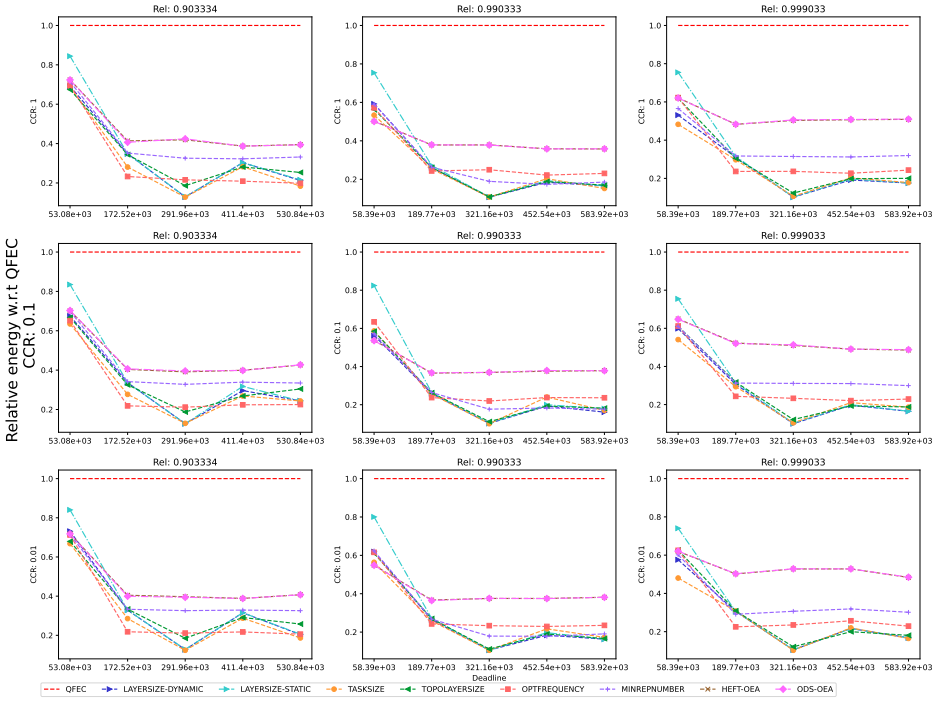

Figure 68: Performance of the different heuristics on the SoyKB workflow.

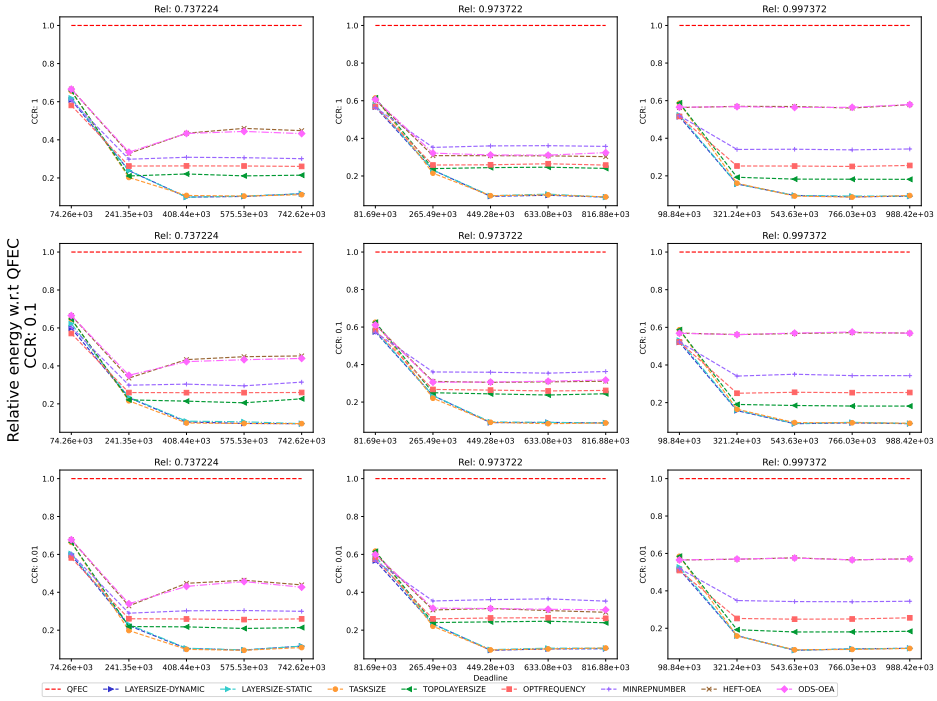

Figure 69: Performance of the different heuristics on the SRASearch.

D.2.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

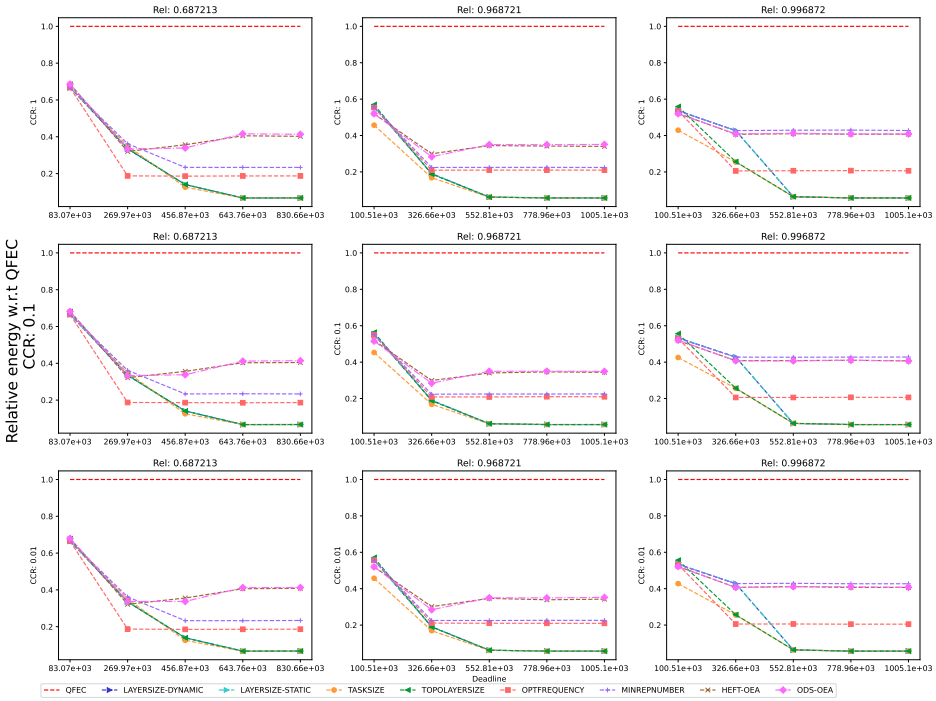

Figure 70: Performance of the different heuristics on the BLAST workflow.

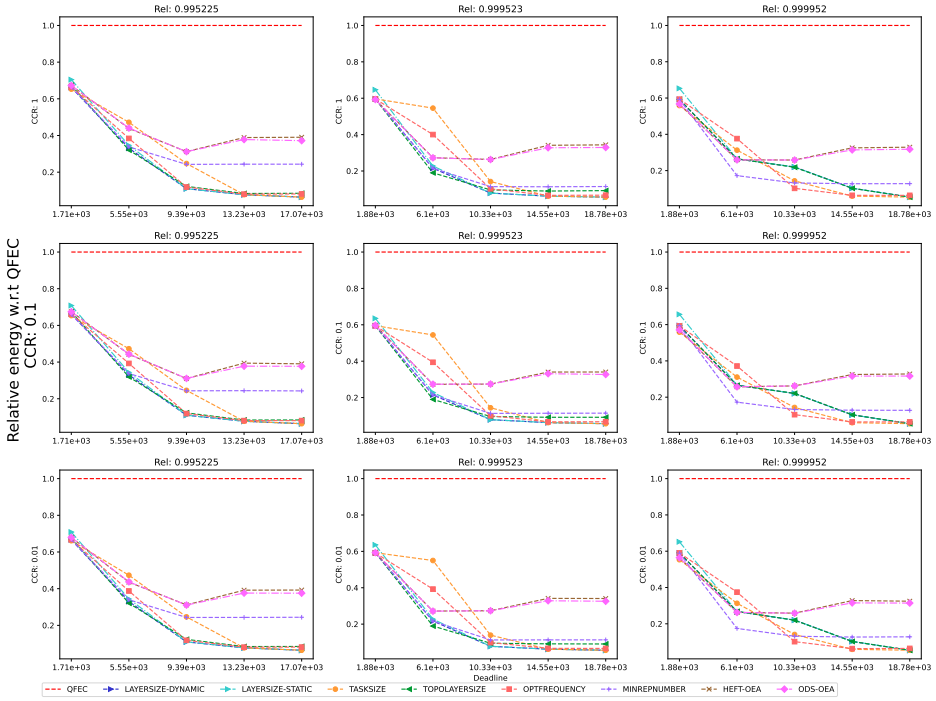

Figure 71: Performance of the different heuristics on the BWA workflow.

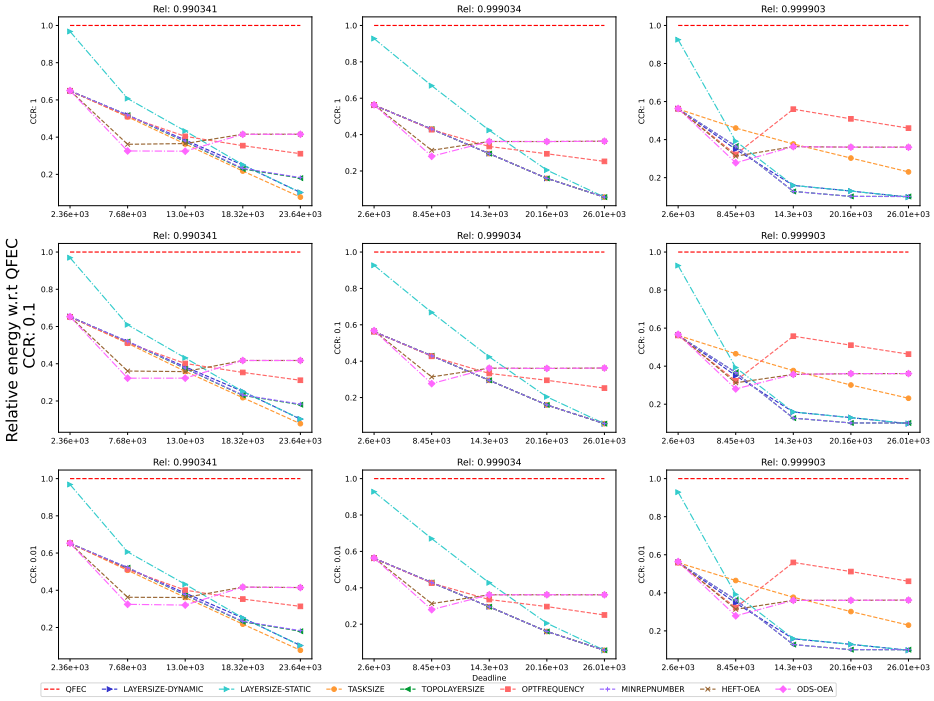

Figure 72: Performance of the different heuristics on the Cholesky workflow.

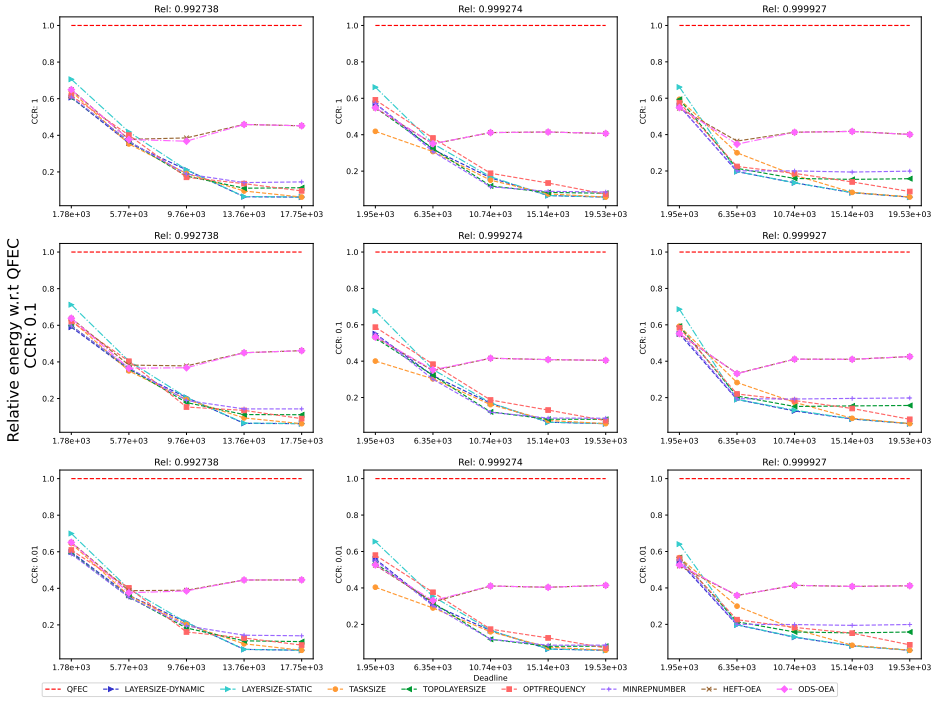

Figure 73: Performance of the different heuristics on the Cycles workflow.

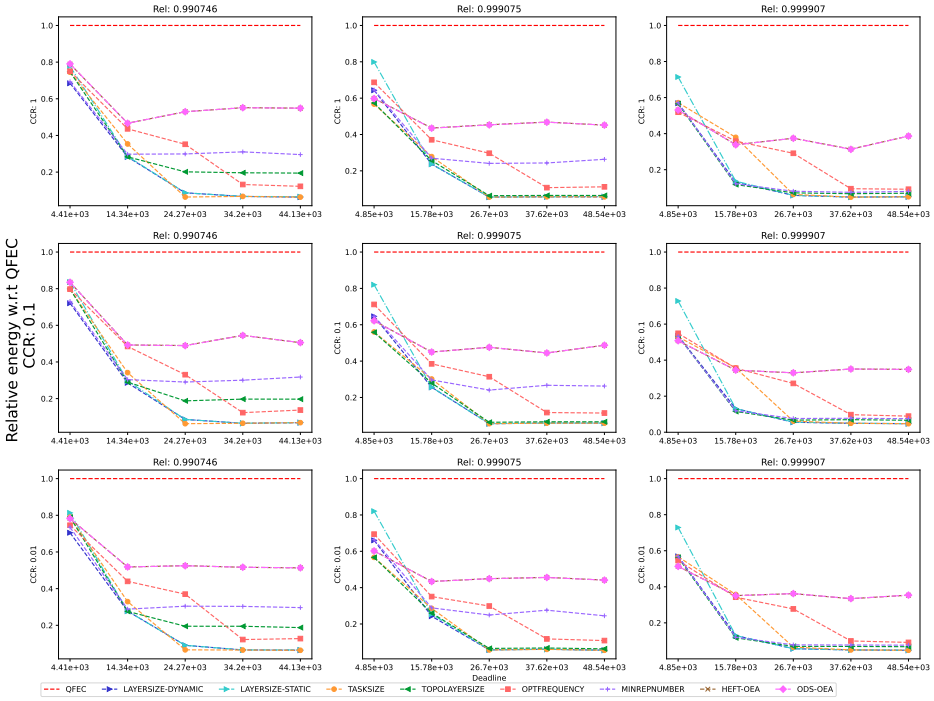

Figure 74: Performance of the different heuristics on the Epigenomics workflow.

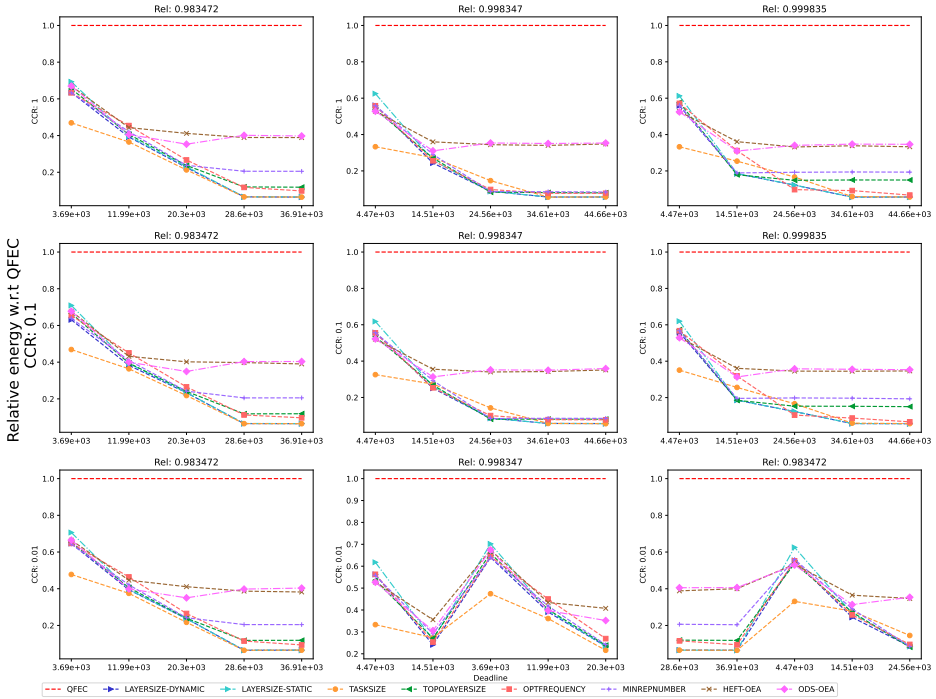

Figure 75: Performance of the different heuristics on the Genome workflow.

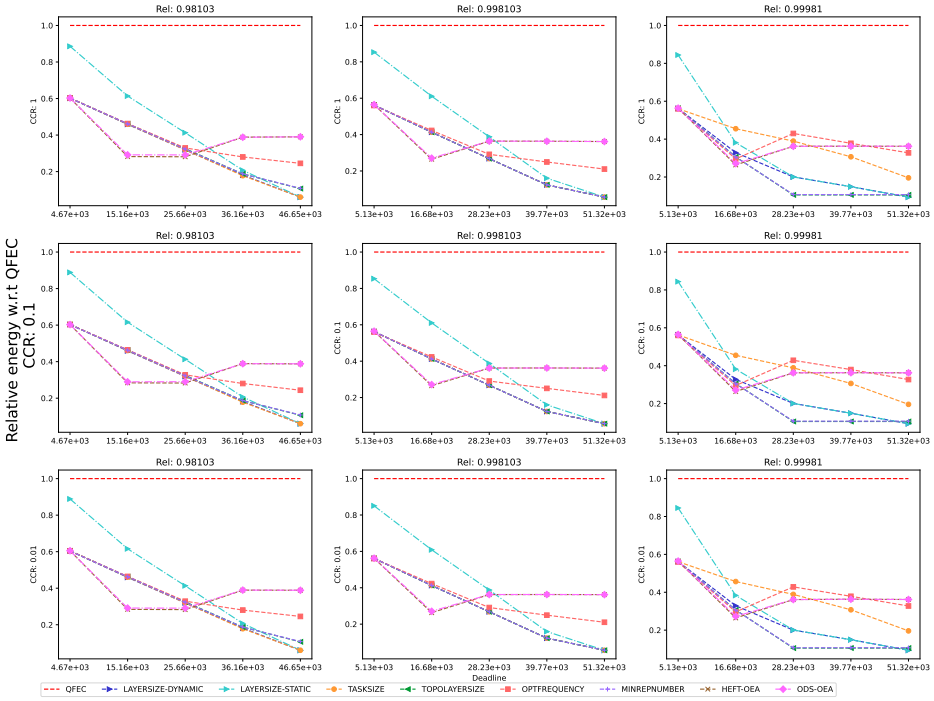

Figure 76: Performance of the different heuristics on the LU workflow.

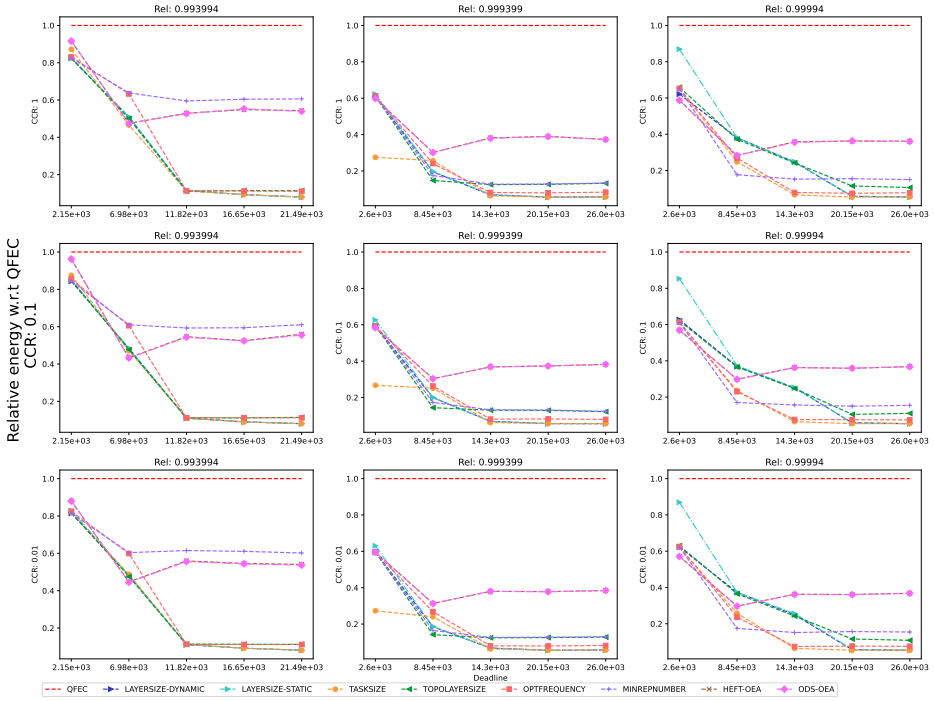

Figure 77: Performance of the different heuristics on the Montage workflow.

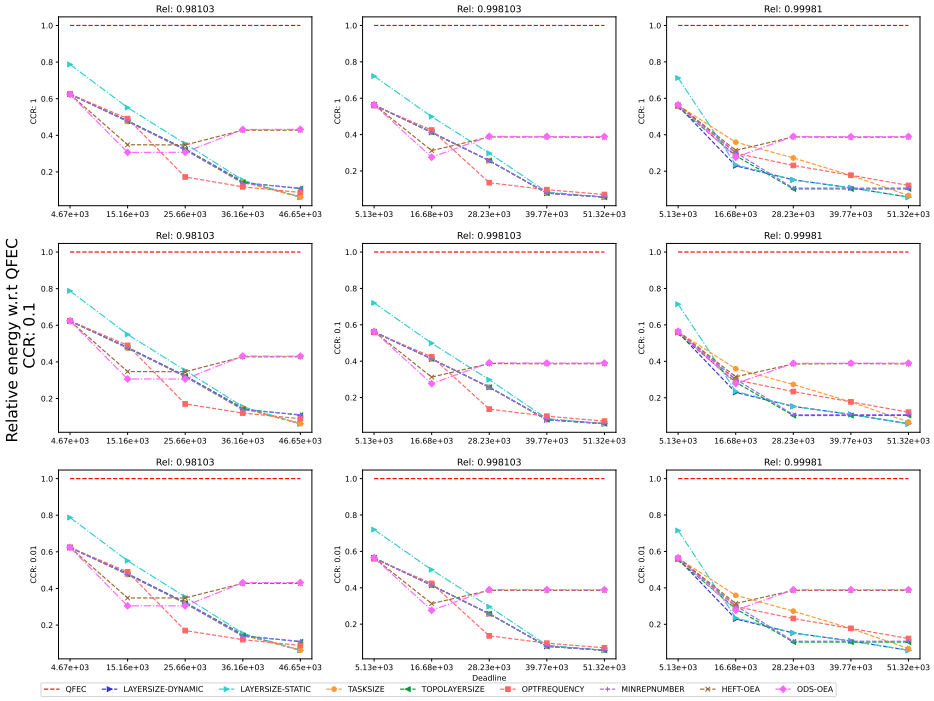

Figure 78: Performance of the different heuristics on the QR workflow.

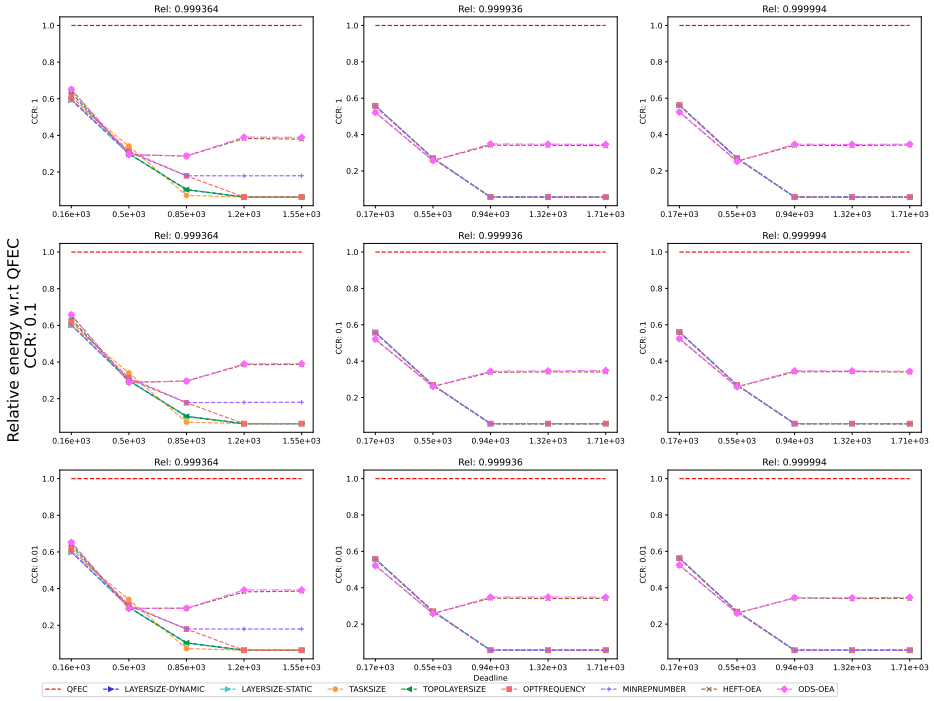

Figure 79: Performance of the different heuristics on the Seismology workflow.

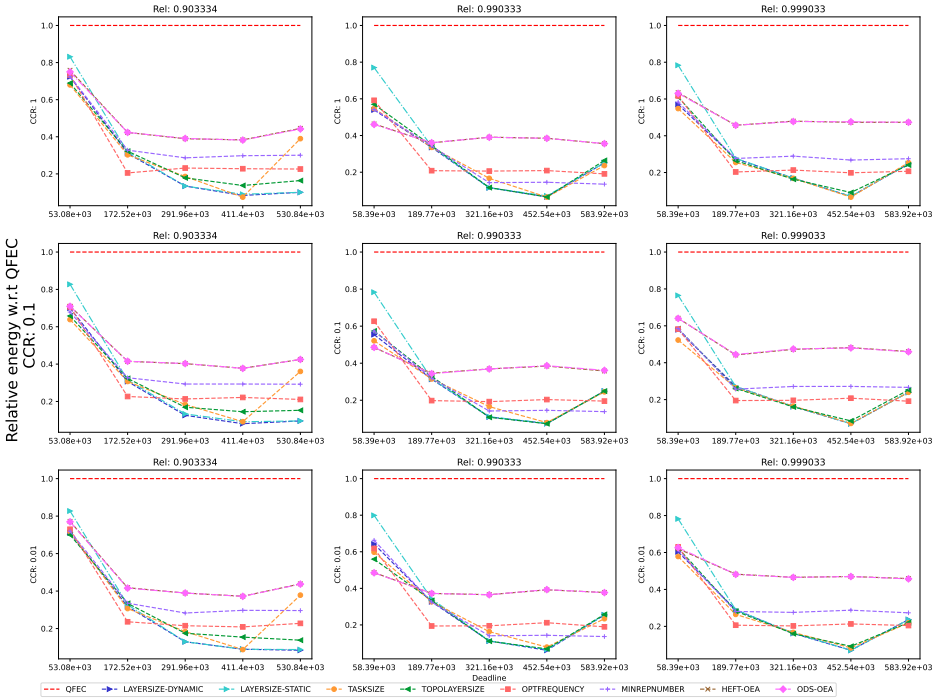

Figure 80: Performance of the different heuristics on the SoyKB workflow.

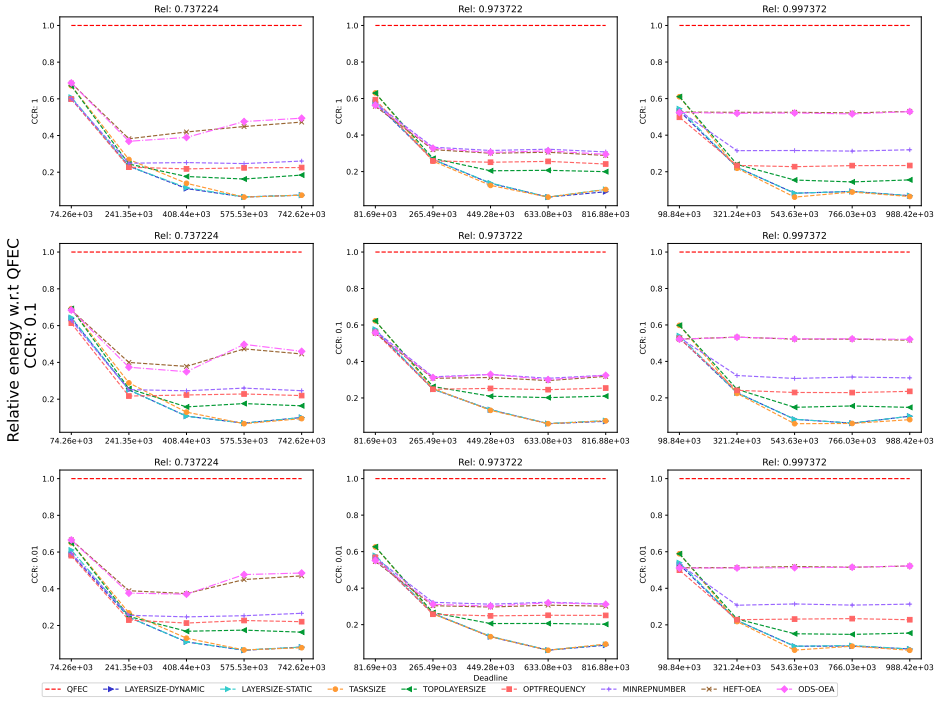

Figure 81: Performance of the different heuristics on the SRASearch.

D.2.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

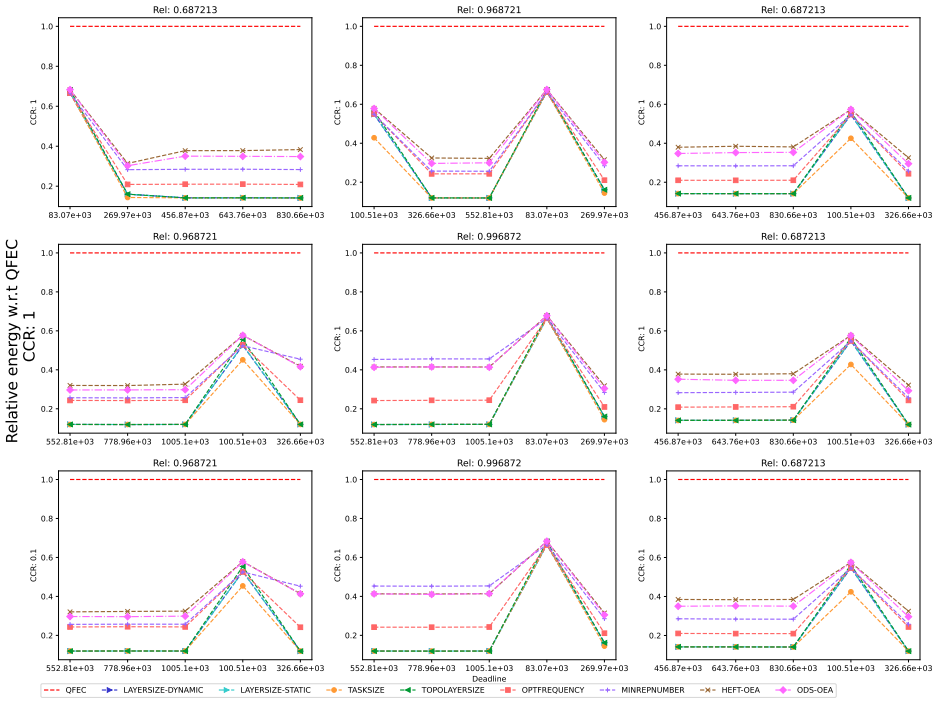

Figure 82: Performance of the different heuristics on the BLAST workflow.

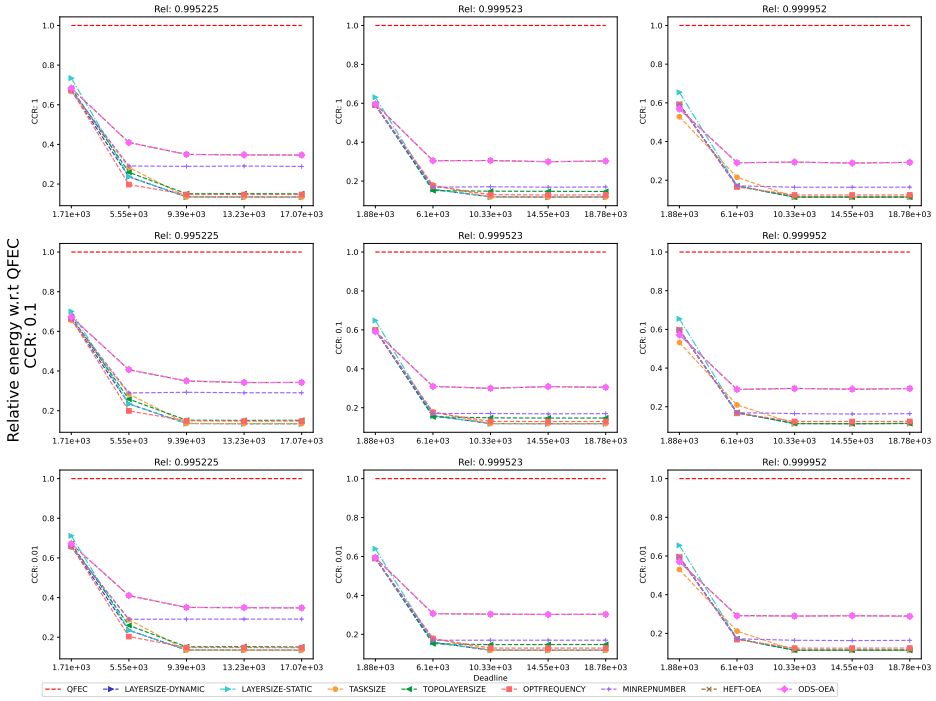

Figure 83: Performance of the different heuristics on the BWA workflow.

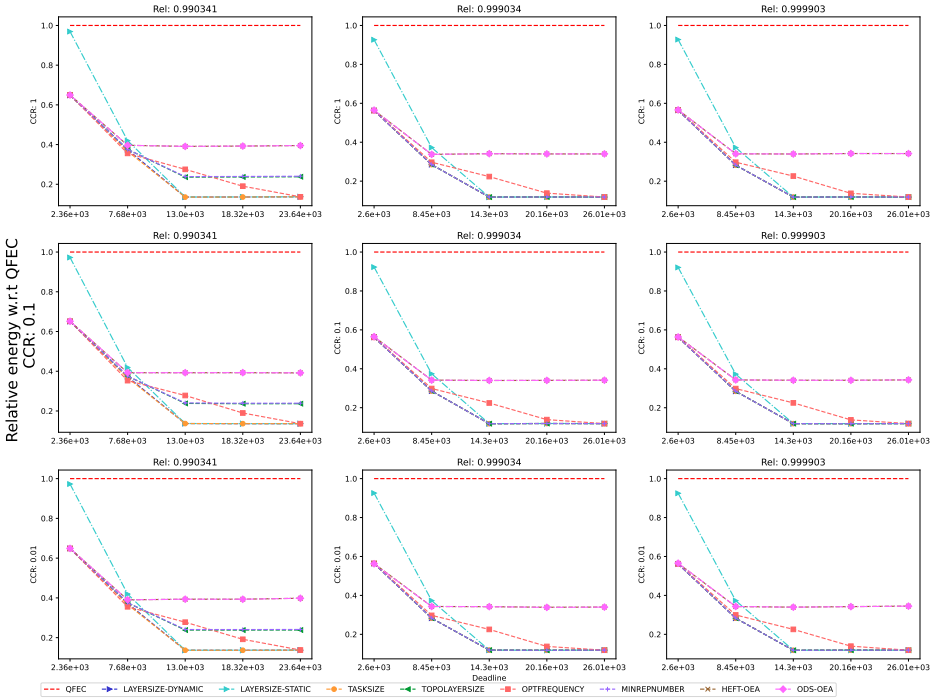

Figure 84: Performance of the different heuristics on the Cholesky workflow.

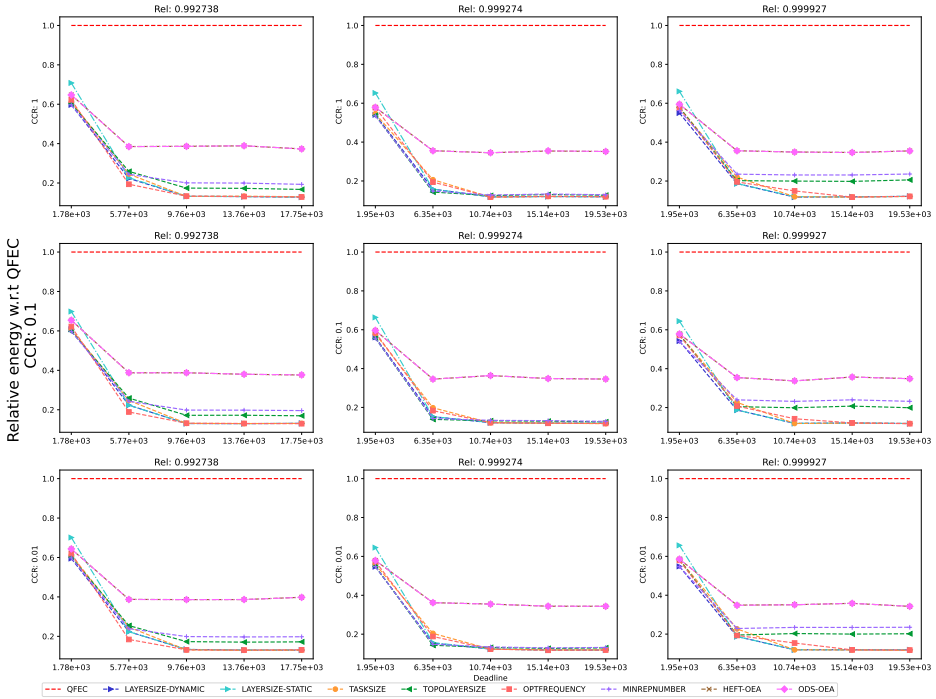

Figure 85: Performance of the different heuristics on the Cycles workflow.

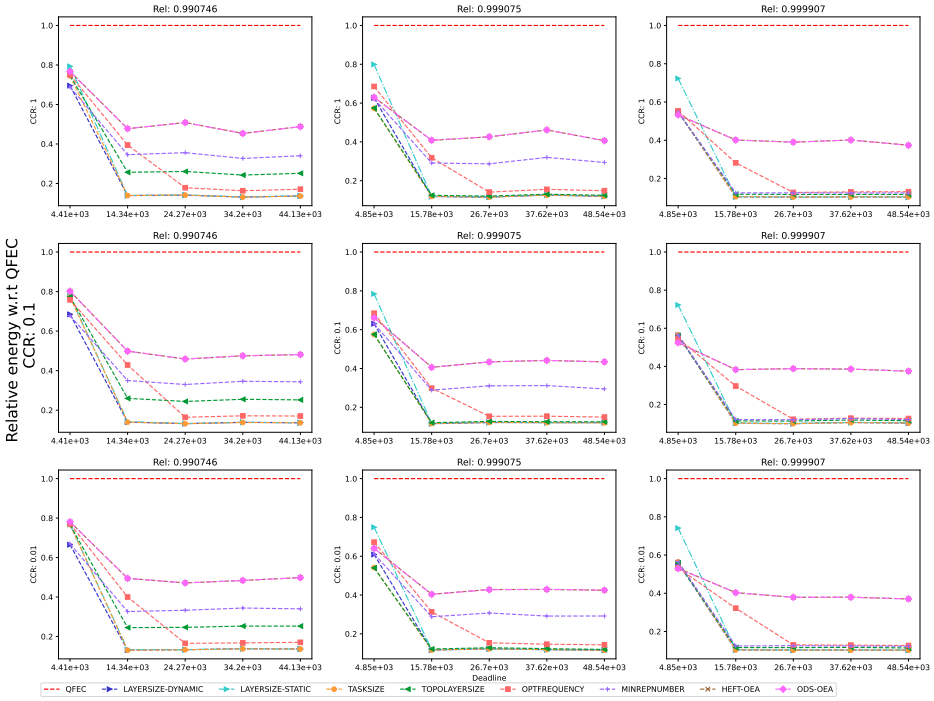

Figure 86: Performance of the different heuristics on the Epigenomics workflow.

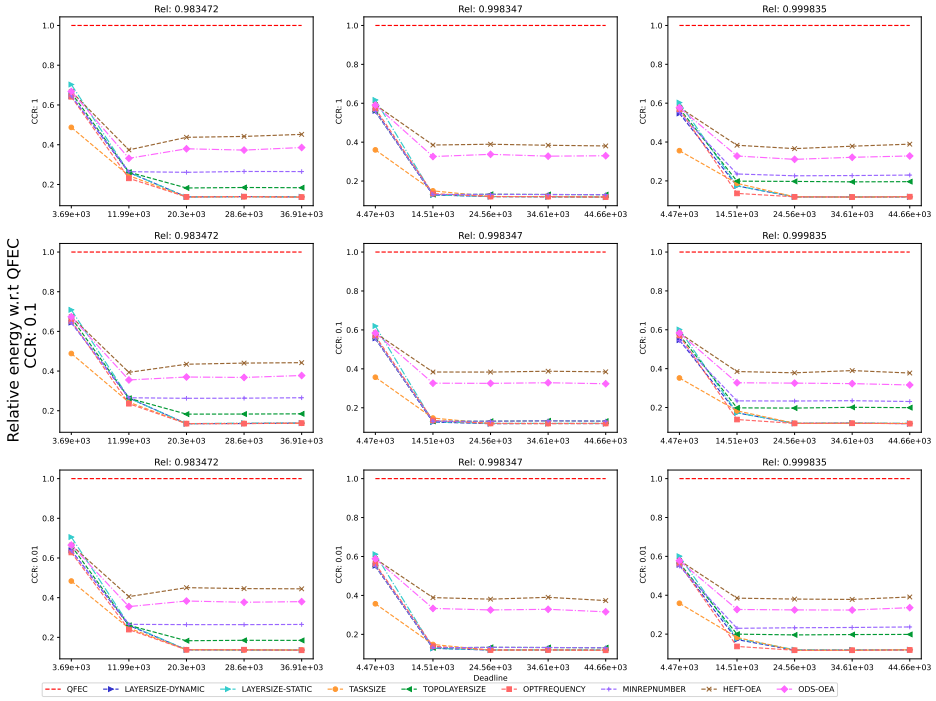

Figure 87: Performance of the different heuristics on the Genome workflow.

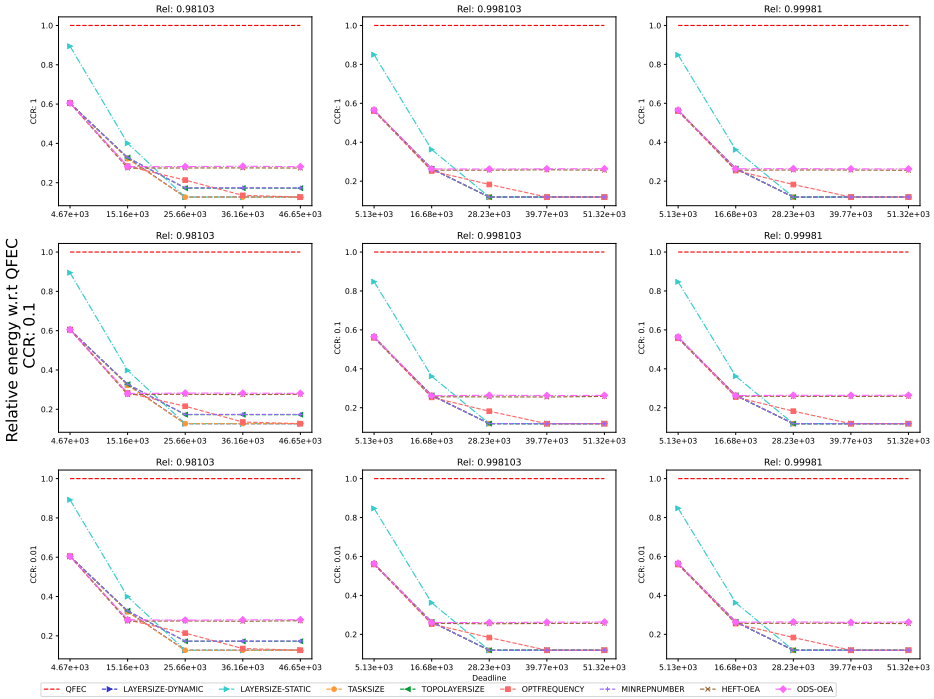

Figure 88: Performance of the different heuristics on the LU workflow.

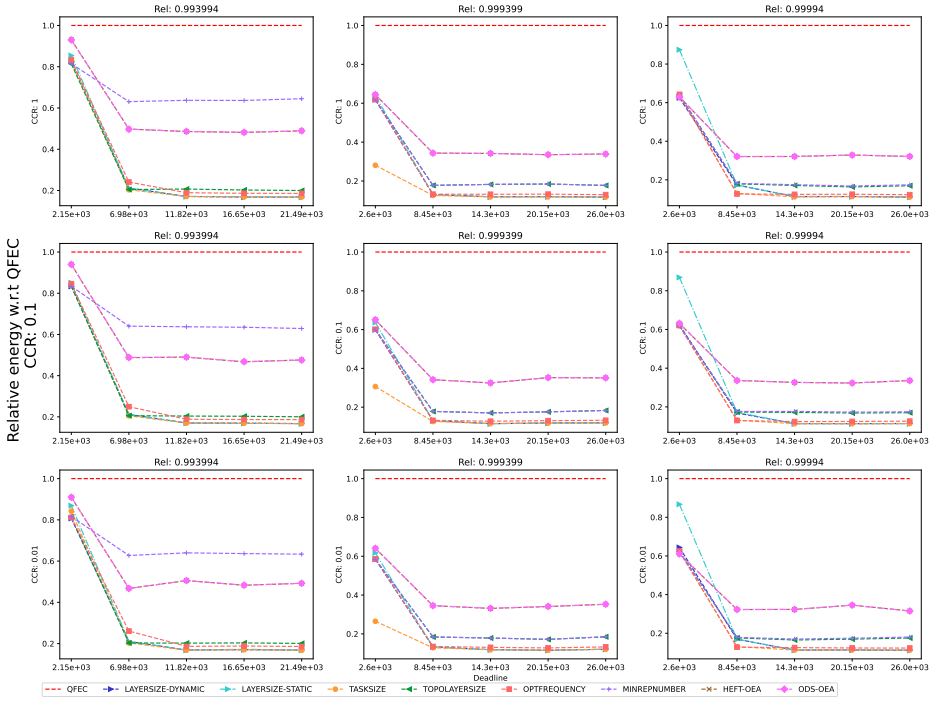

Figure 89: Performance of the different heuristics on the Montage workflow.

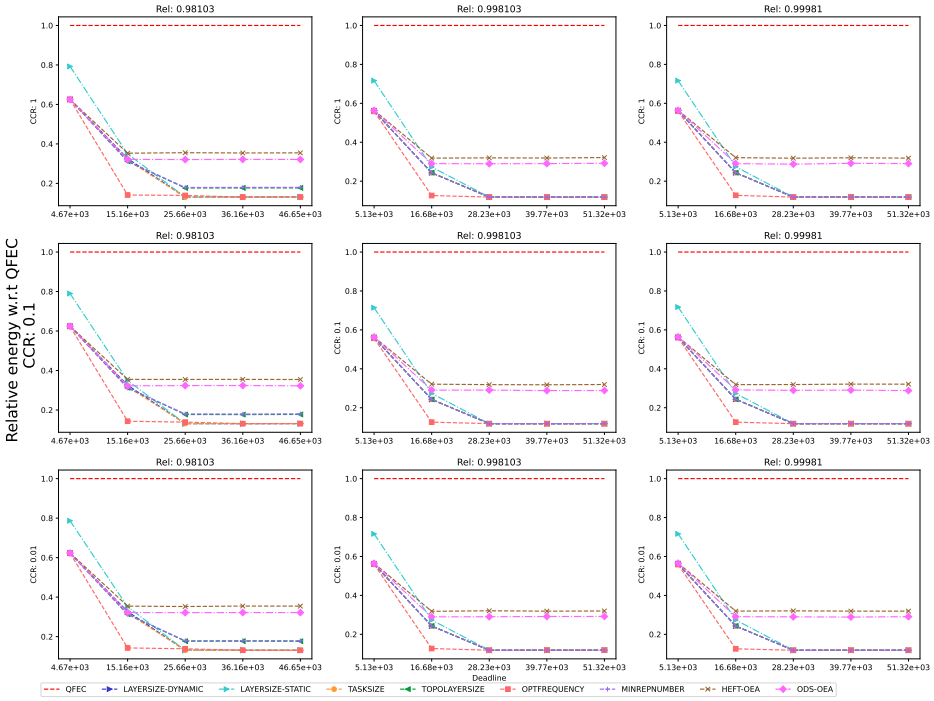

Figure 90: Performance of the different heuristics on the QR workflow.

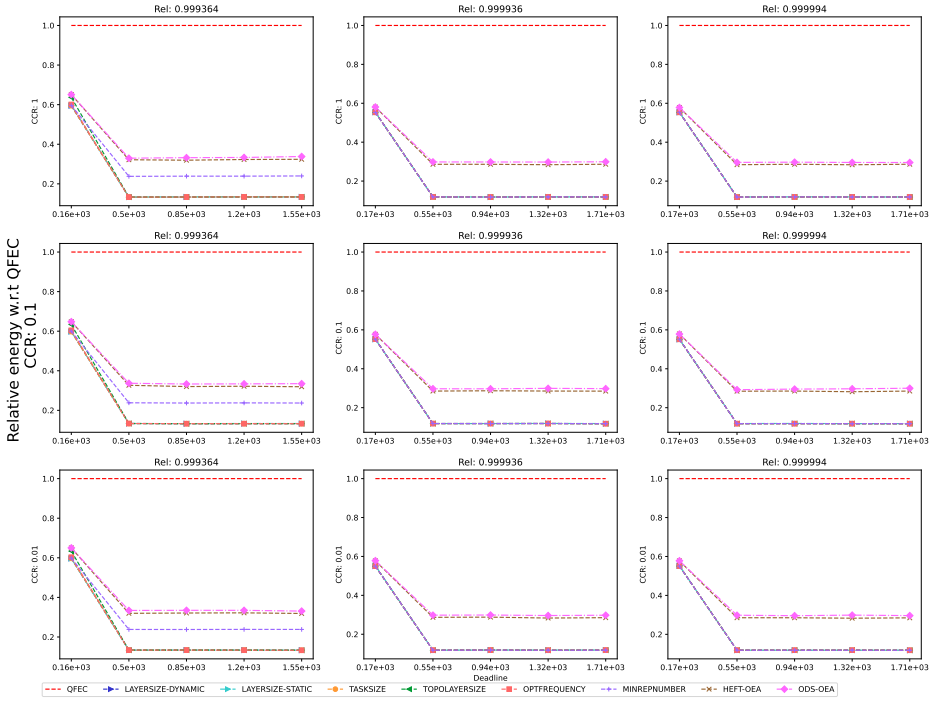

Figure 91: Performance of the different heuristics on the Seismology workflow.

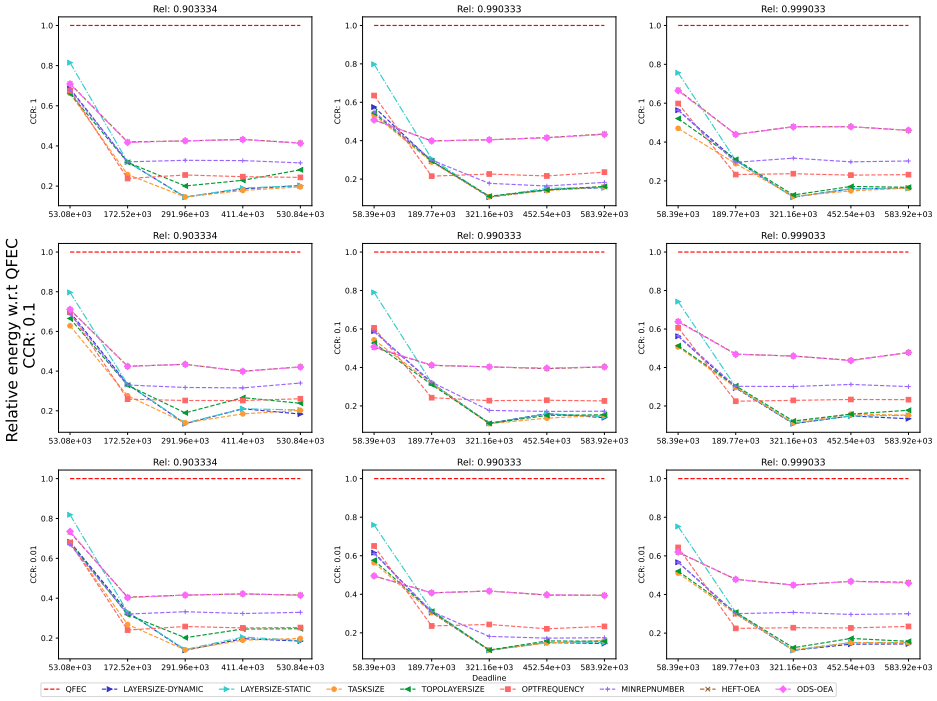

Figure 92: Performance of the different heuristics on the SoyKB workflow.

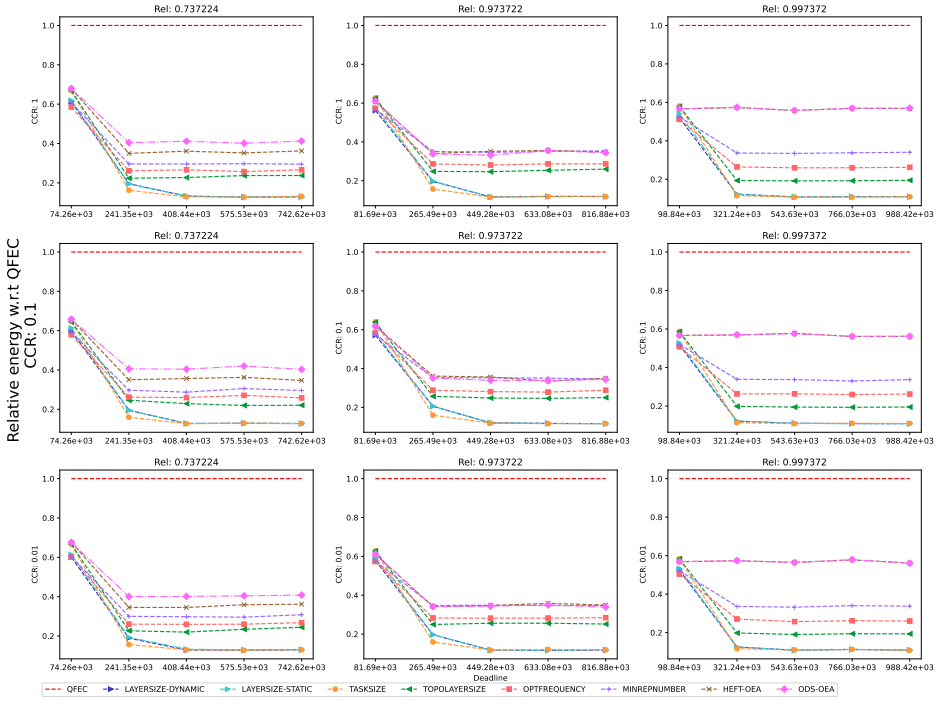

Figure 93: Performance of the different heuristics on the SRASearch.

D.2.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

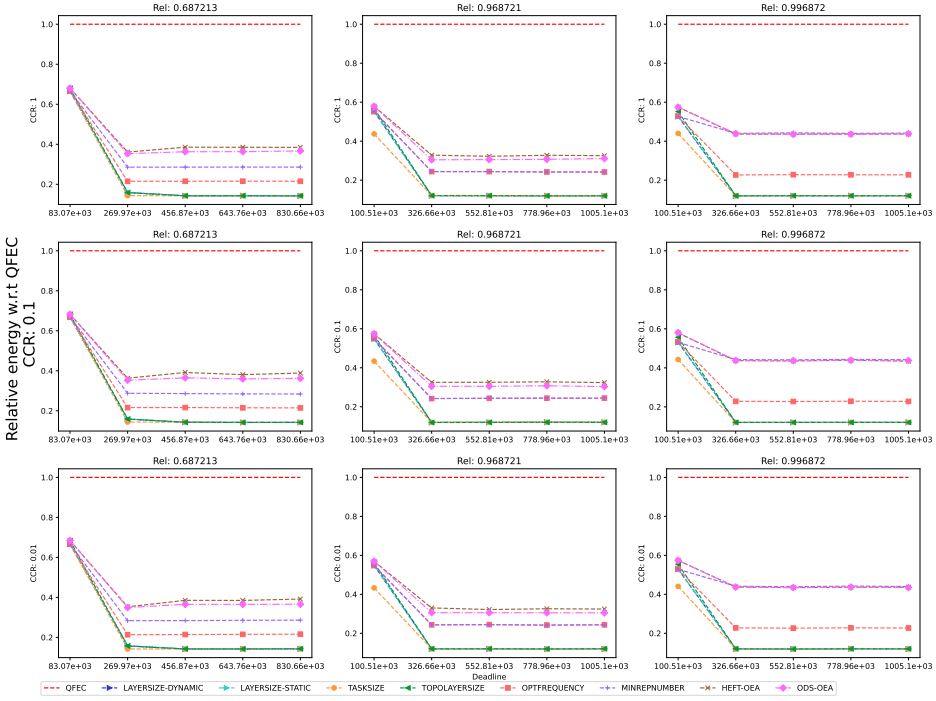

Figure 94: Performance of the different heuristics on the BLAST workflow.

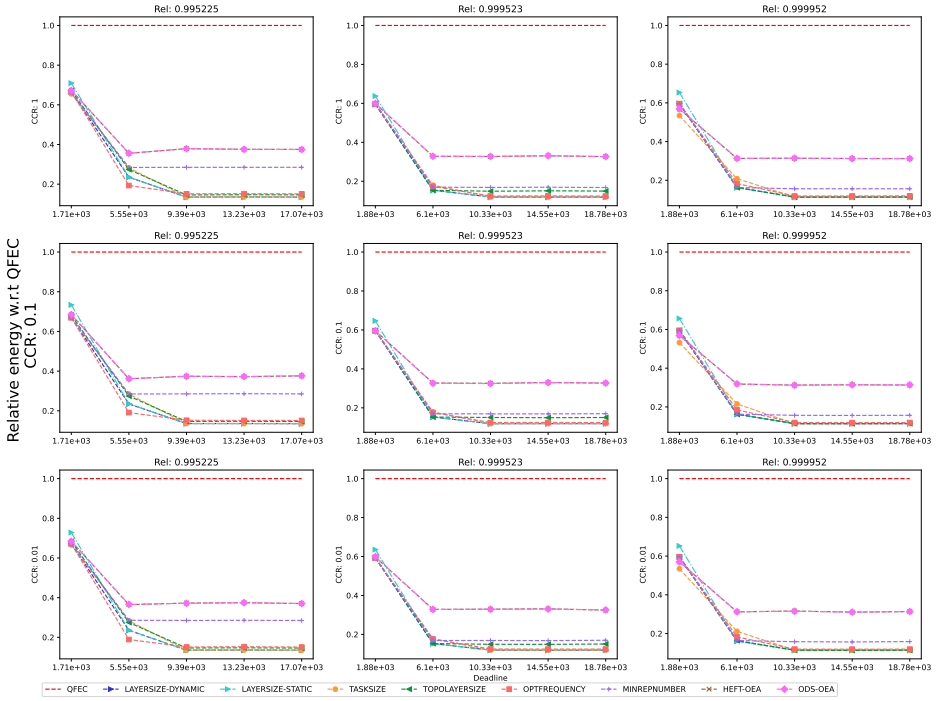

Figure 95: Performance of the different heuristics on the BWA workflow.

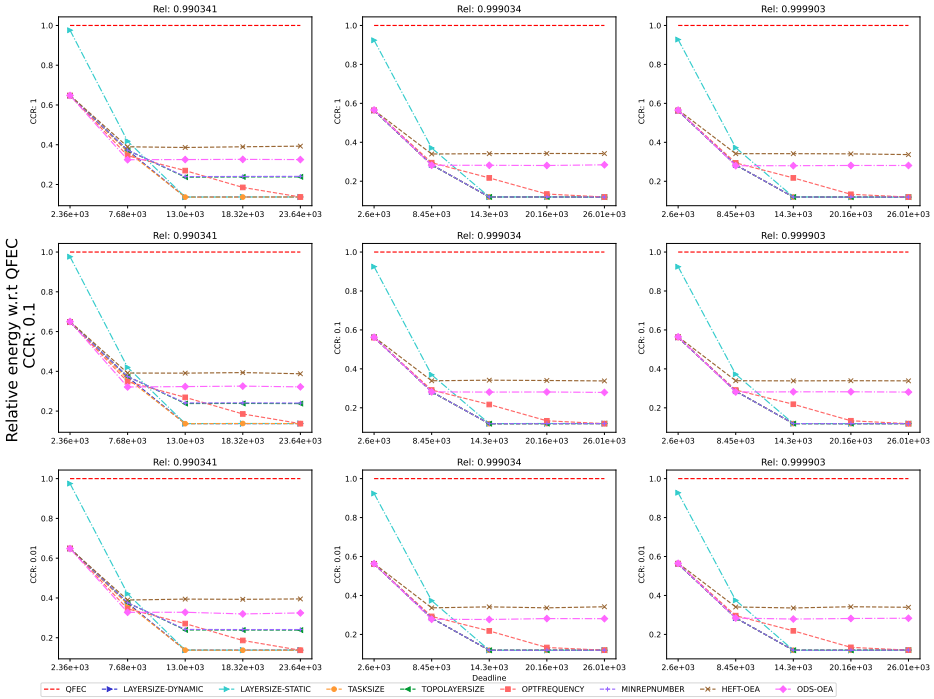

Figure 96: Performance of the different heuristics on the Cholesky workflow.

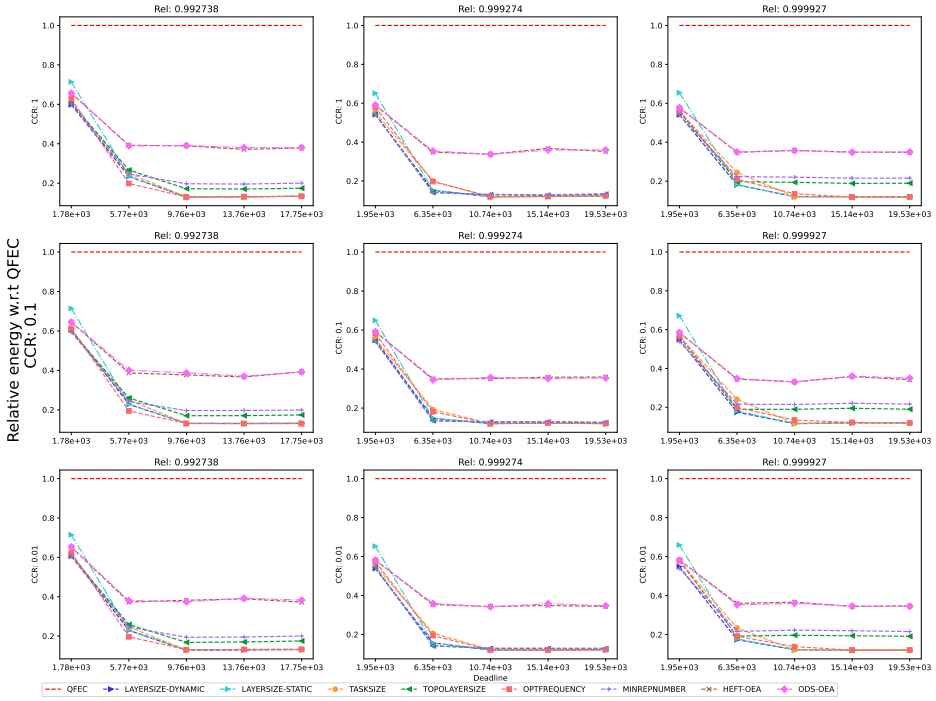

Figure 97: Performance of the different heuristics on the Cycles workflow.

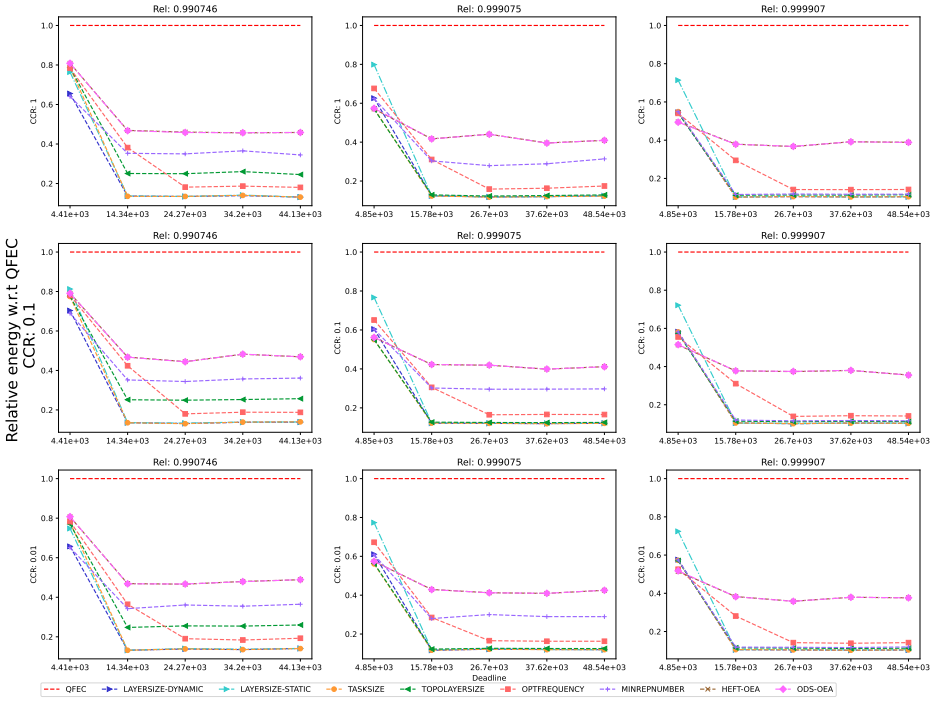

Figure 98: Performance of the different heuristics on the Epigenomics workflow.

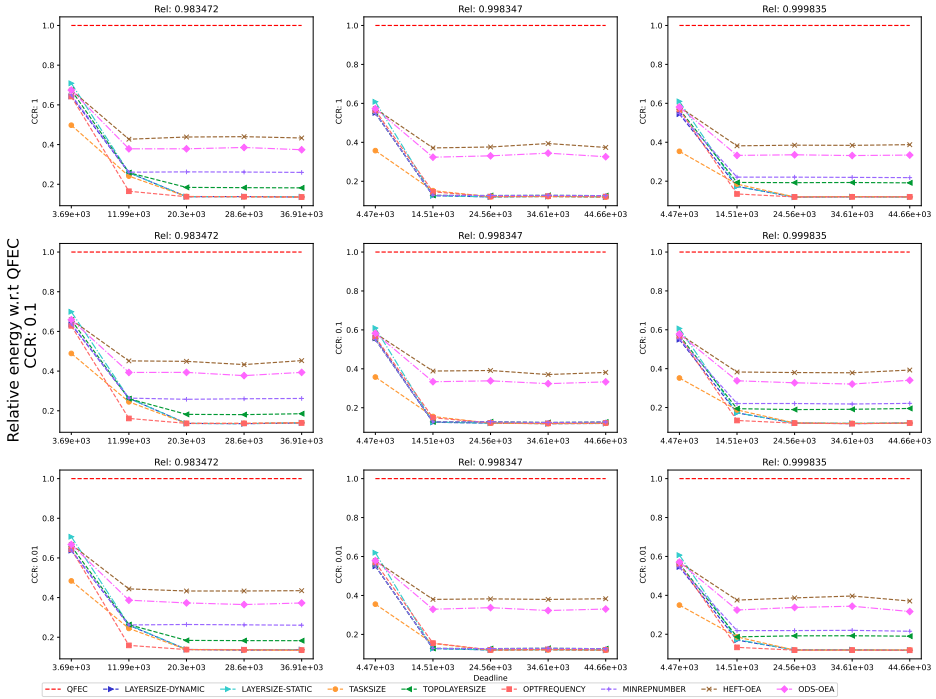

Figure 99: Performance of the different heuristics on the Genome workflow.

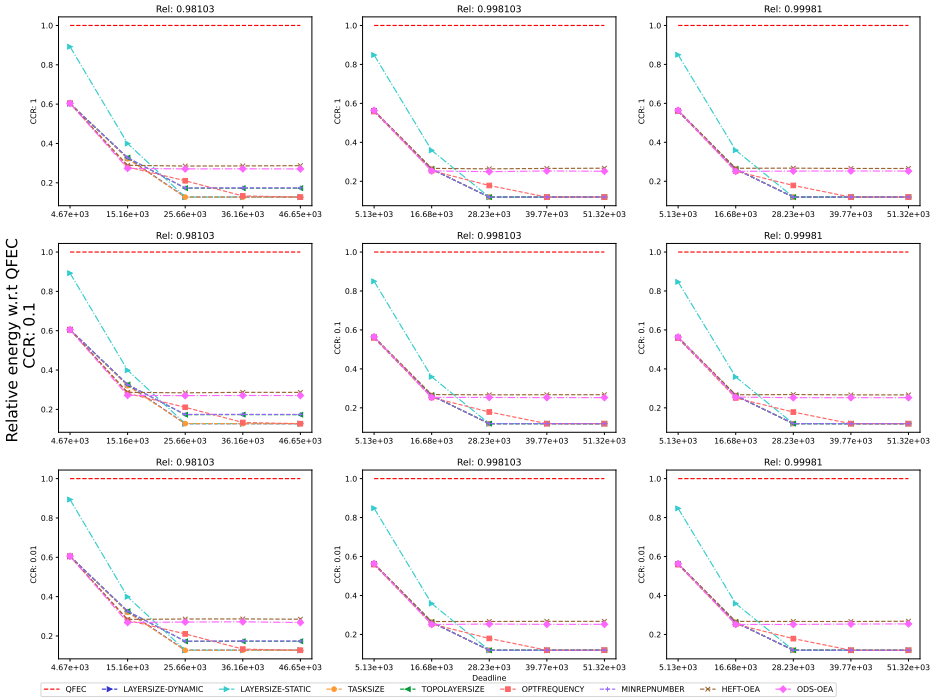

Figure 100: Performance of the different heuristics on the LU workflow.

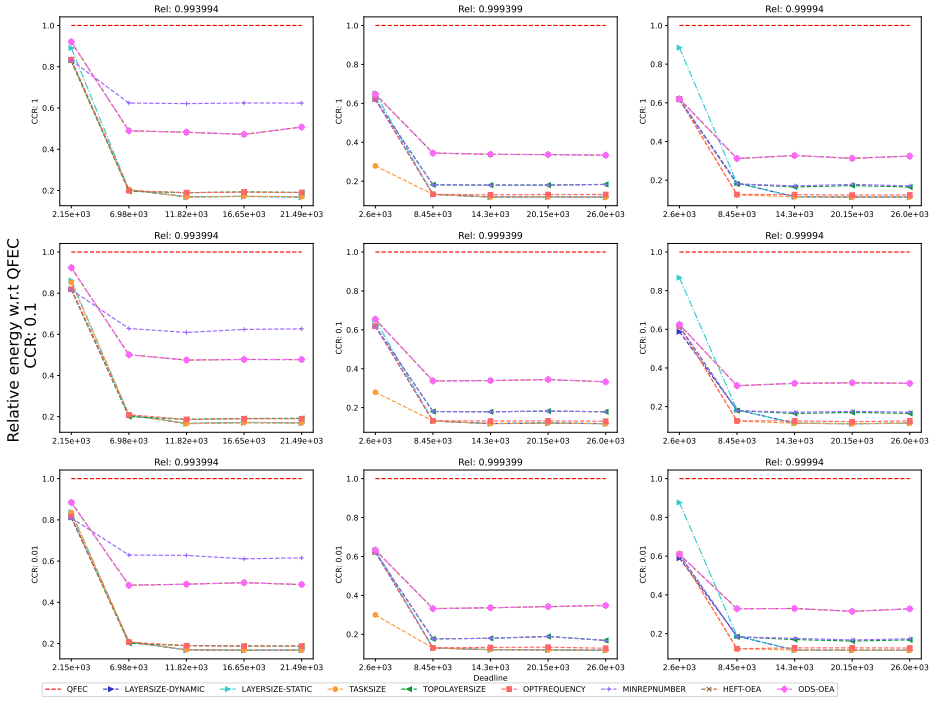

Figure 101: Performance of the different heuristics on the Montage workflow.

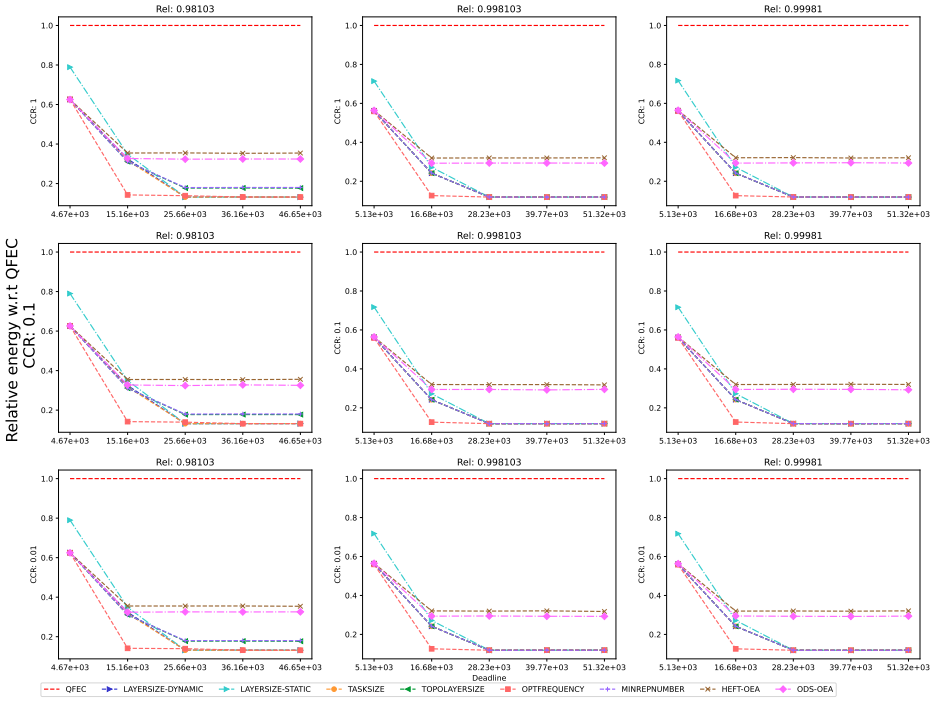

Figure 102: Performance of the different heuristics on the QR workflow.

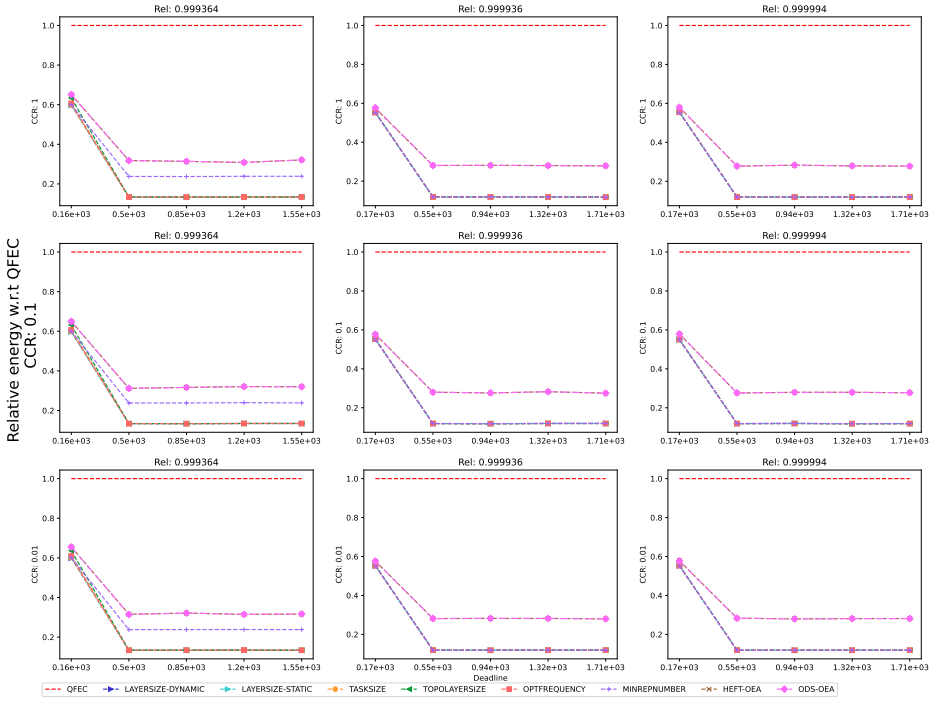

Figure 103: Performance of the different heuristics on the Seismology workflow.

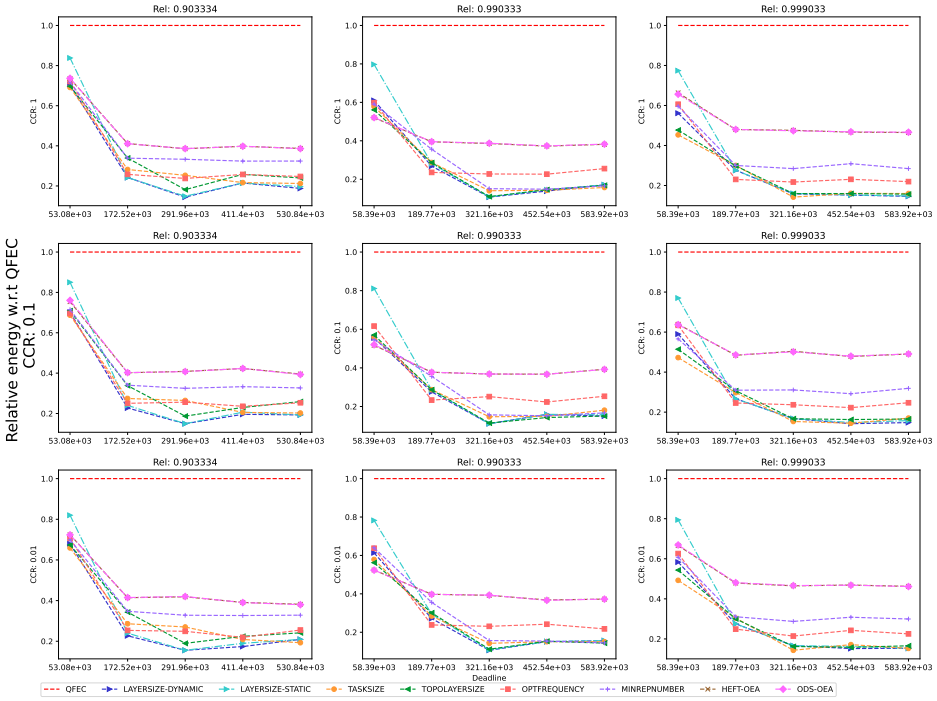

Figure 104: Performance of the different heuristics on the SoyKB workflow.

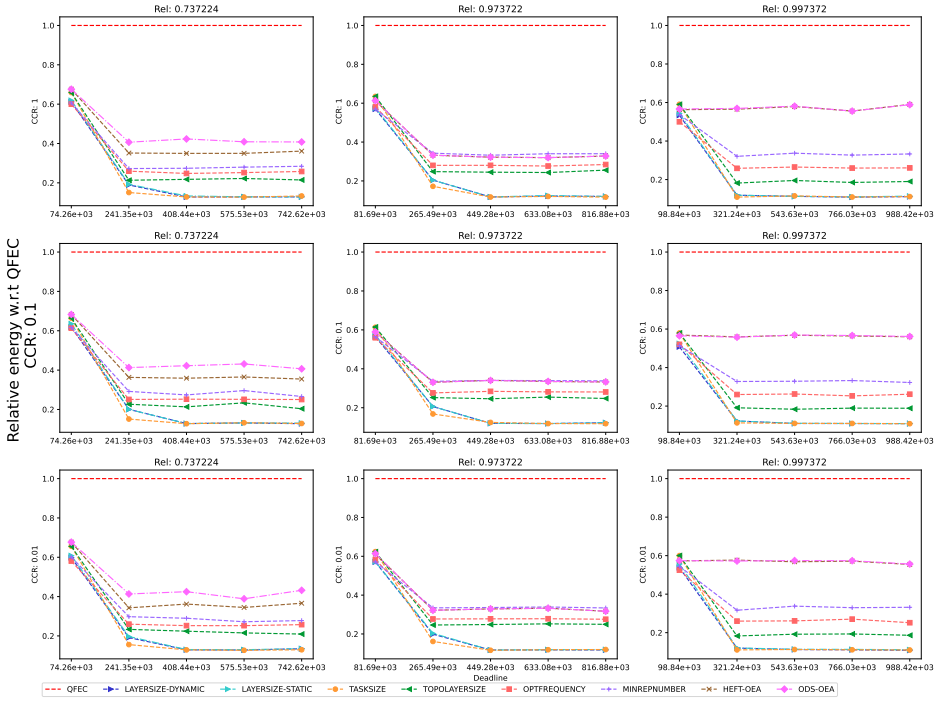

Figure 105: Performance of the different heuristics on the SRASearch.

D.3 $BC/WC = 0.3$

D.3.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

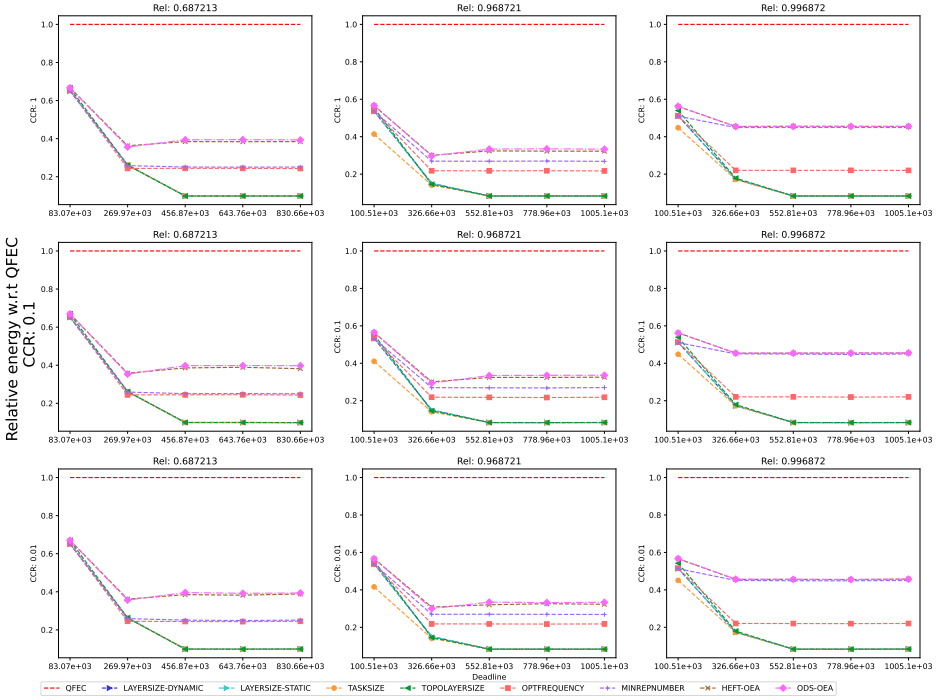

Figure 106: Performance of the different heuristics on the BLAST workflow.

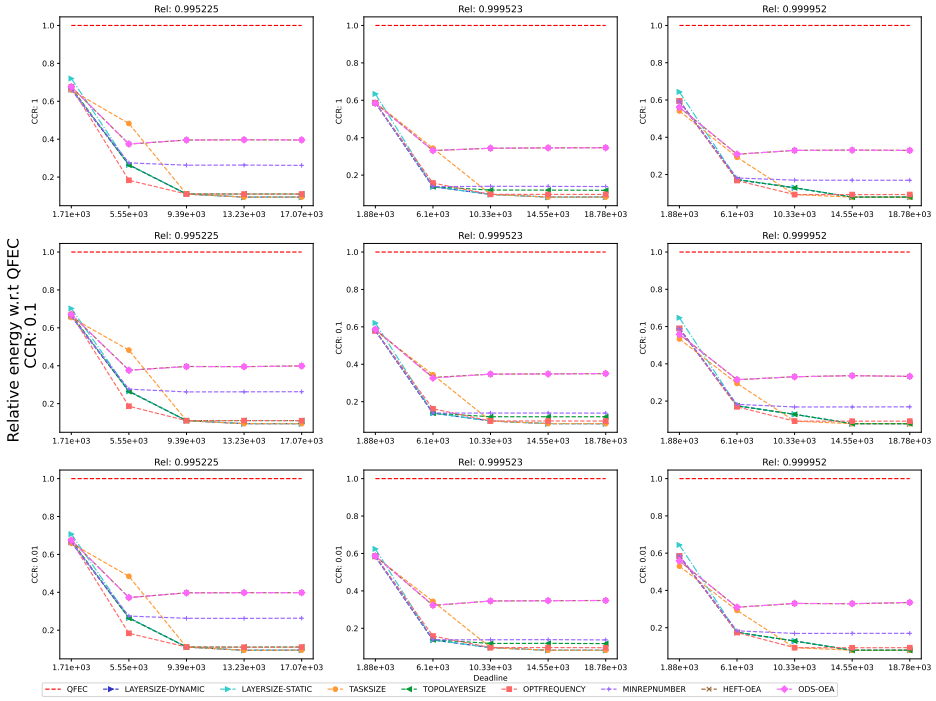

Figure 107: Performance of the different heuristics on the BWA workflow.

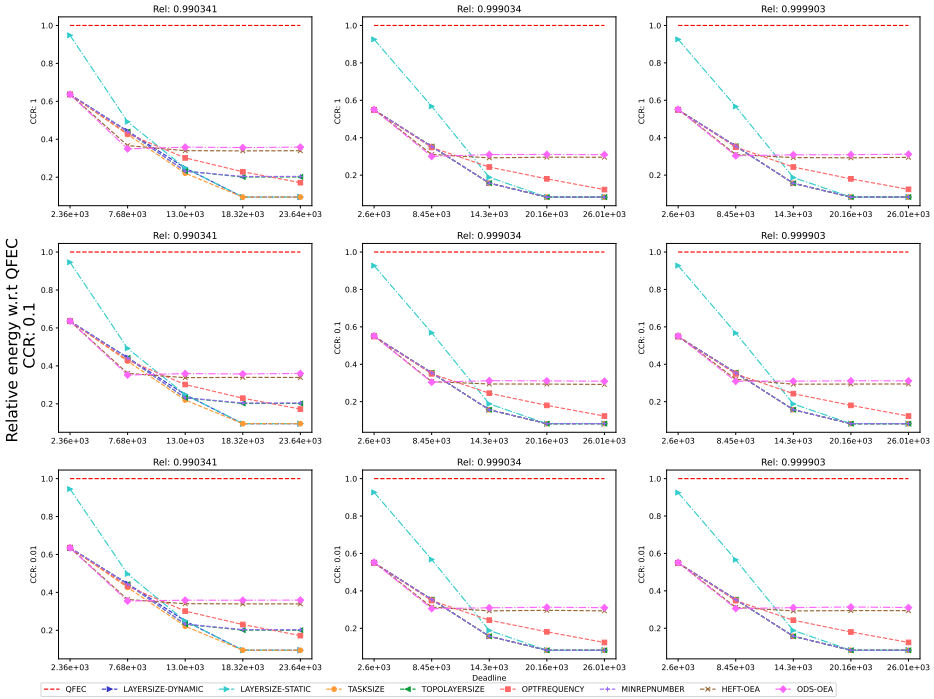

Figure 108: Performance of the different heuristics on the Cholesky workflow.

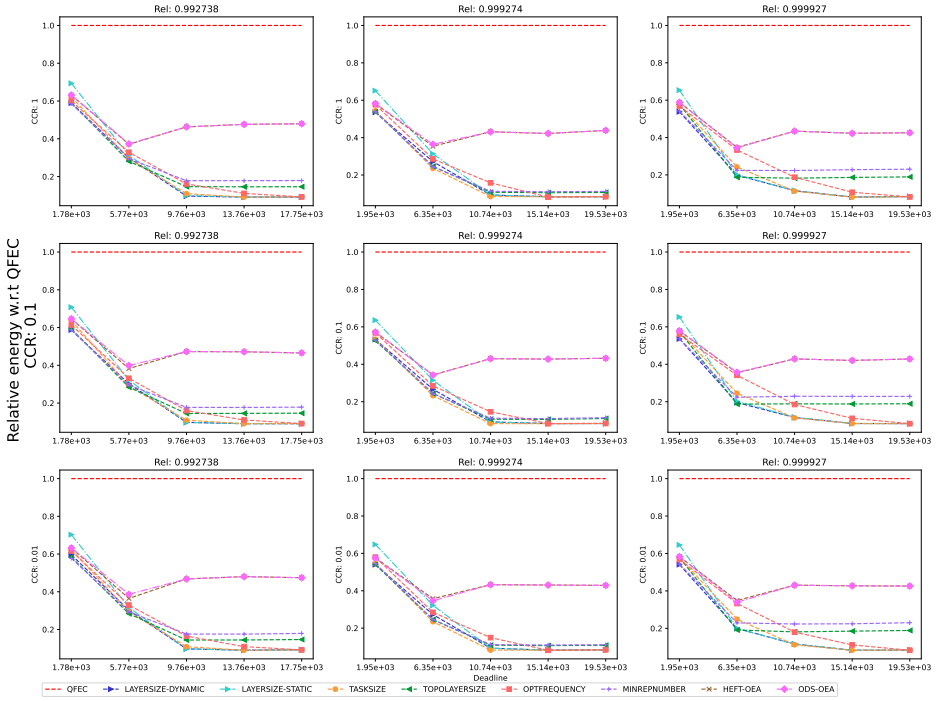

Figure 109: Performance of the different heuristics on the Cycles workflow.

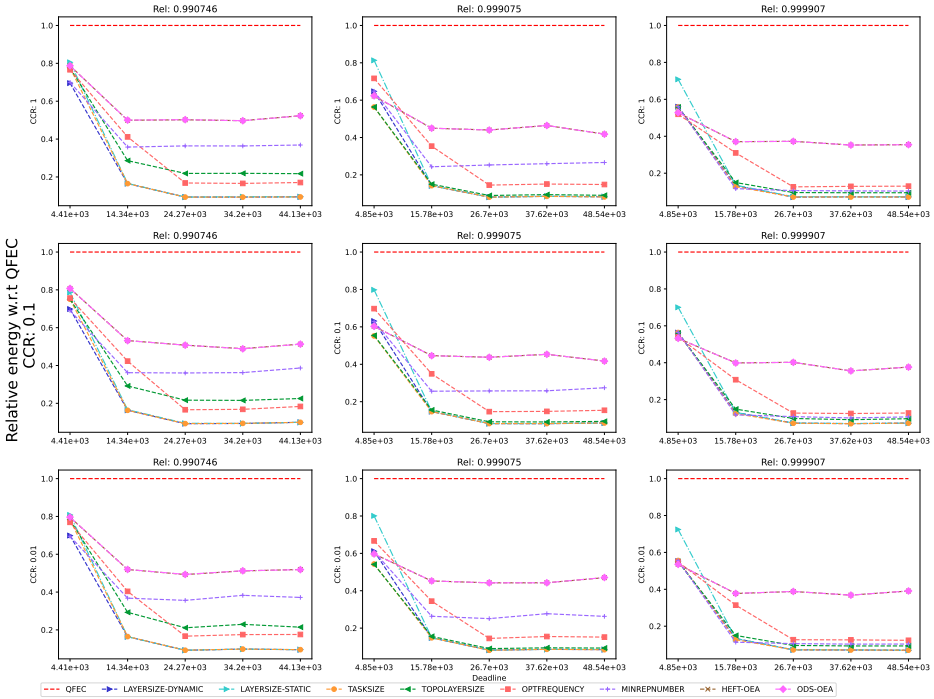

Figure 110: Performance of the different heuristics on the Epigenomics workflow.

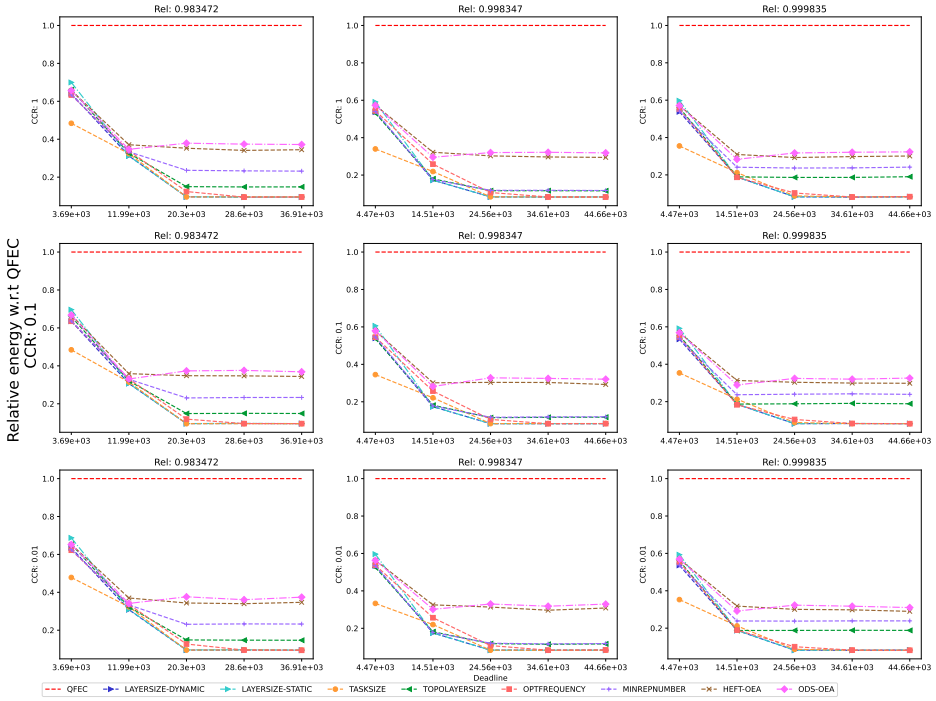

Figure 111: Performance of the different heuristics on the Genome workflow.

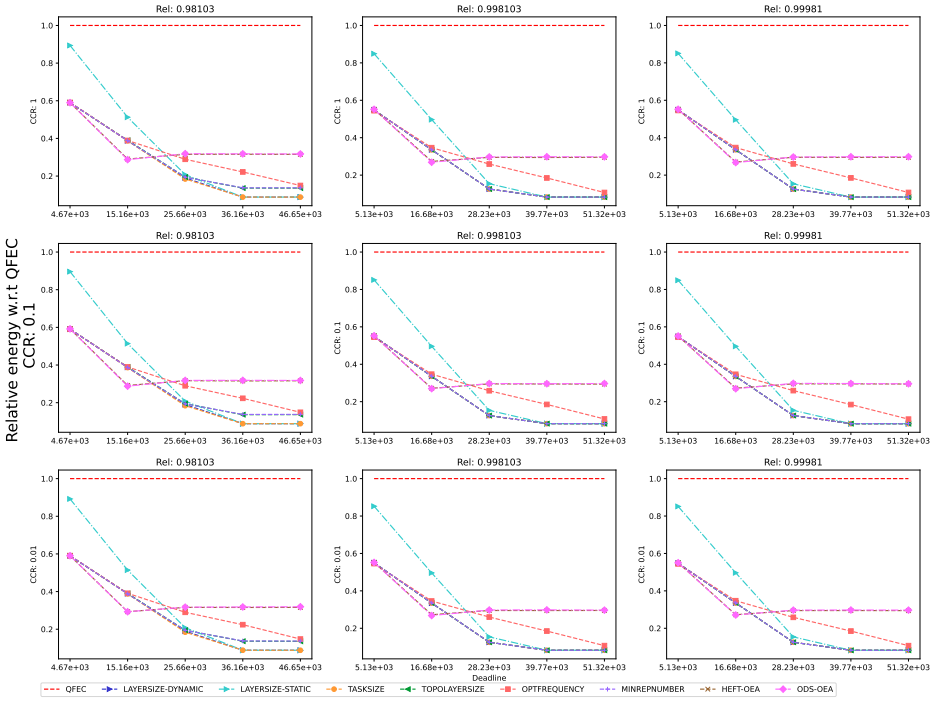

Figure 112: Performance of the different heuristics on the LU workflow.

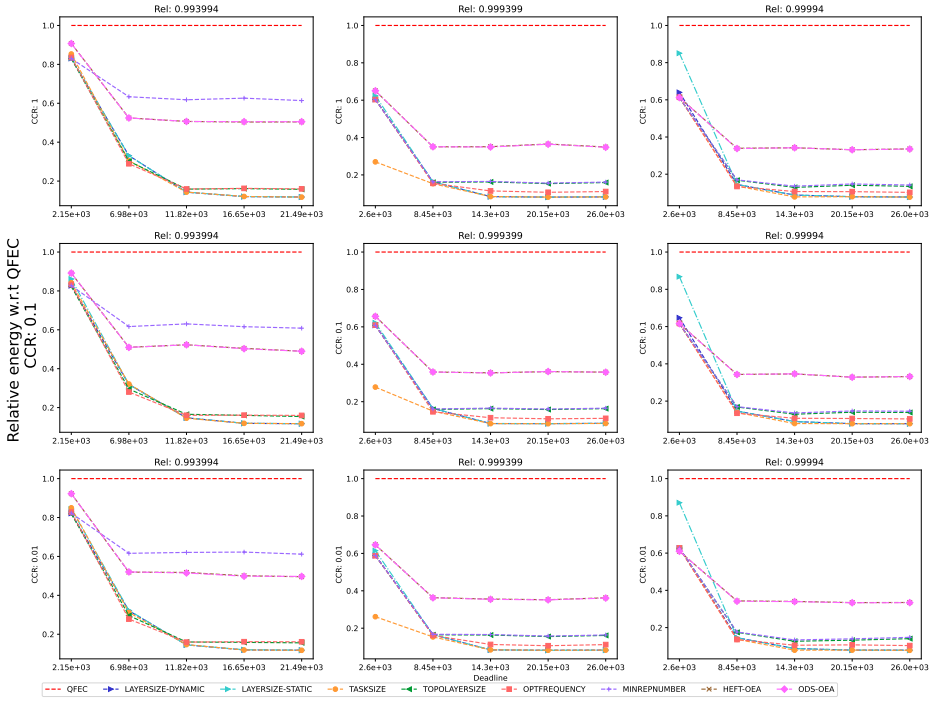

Figure 113: Performance of the different heuristics on the Montage workflow.

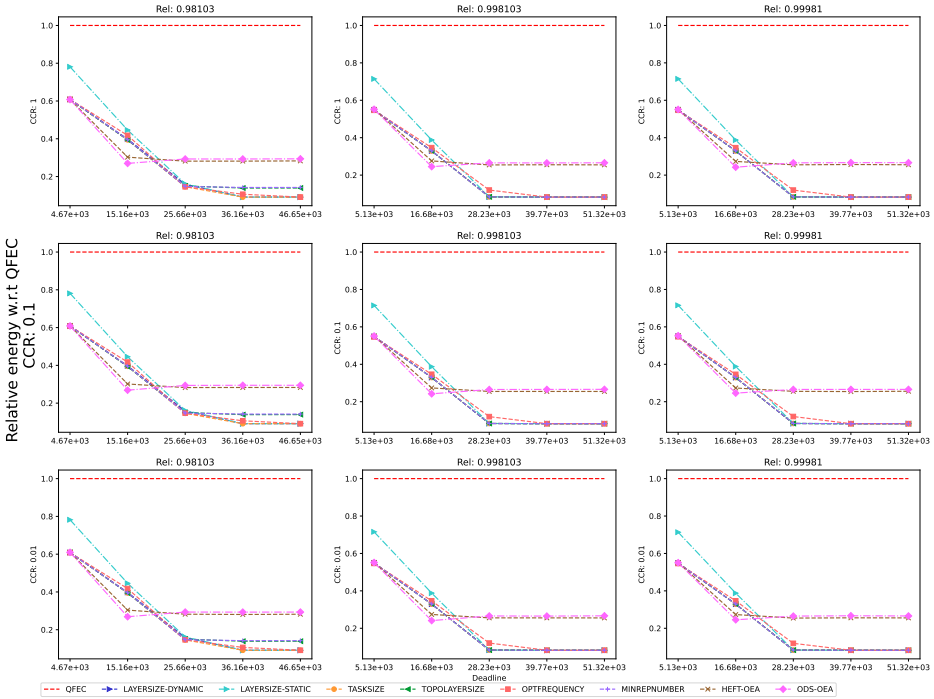

Figure 114: Performance of the different heuristics on the QR workflow.

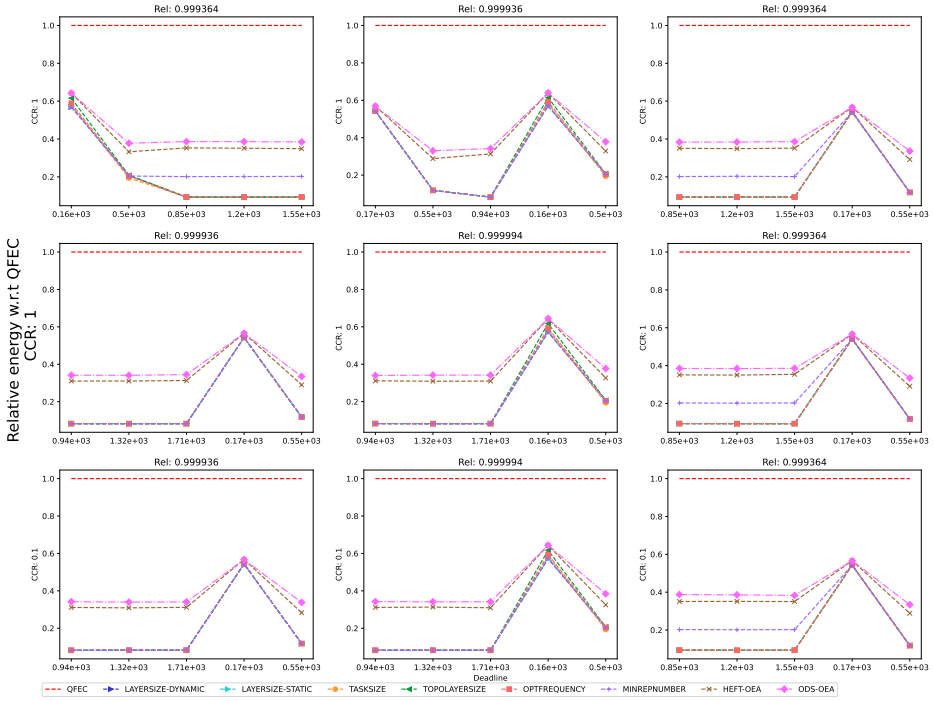

Figure 115: Performance of the different heuristics on the Seismology workflow.

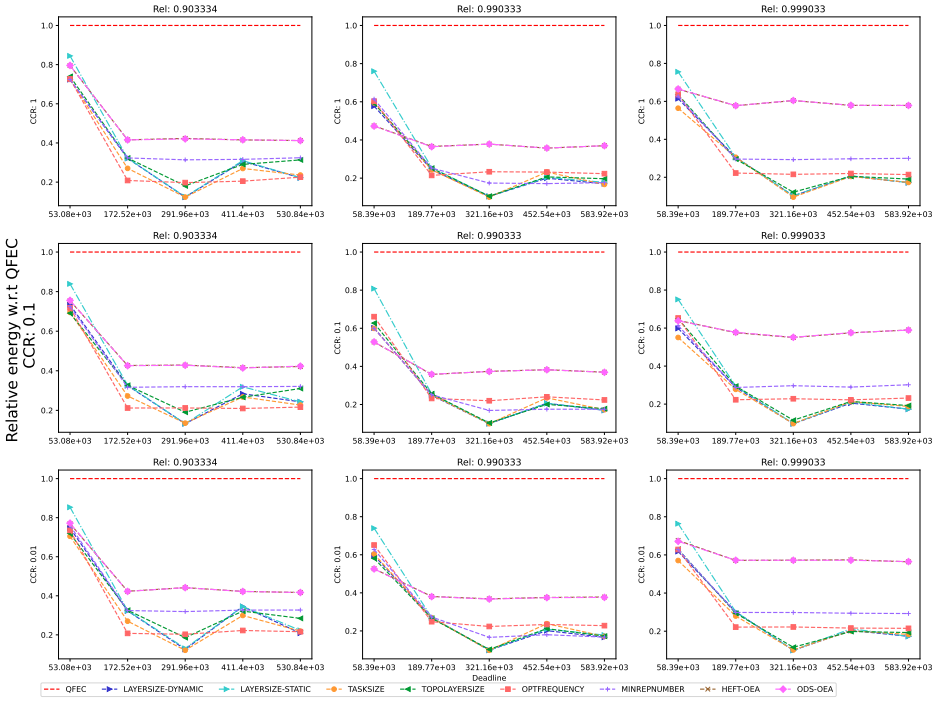

Figure 116: Performance of the different heuristics on the SoyKB workflow.

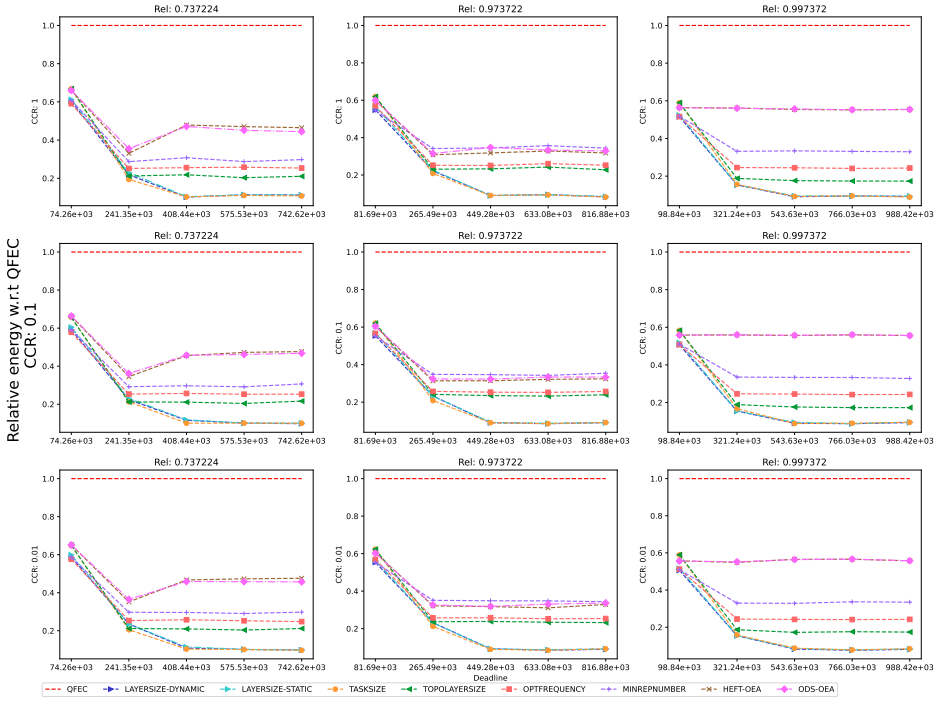

Figure 117: Performance of the different heuristics on the SRASearch.

D.3.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

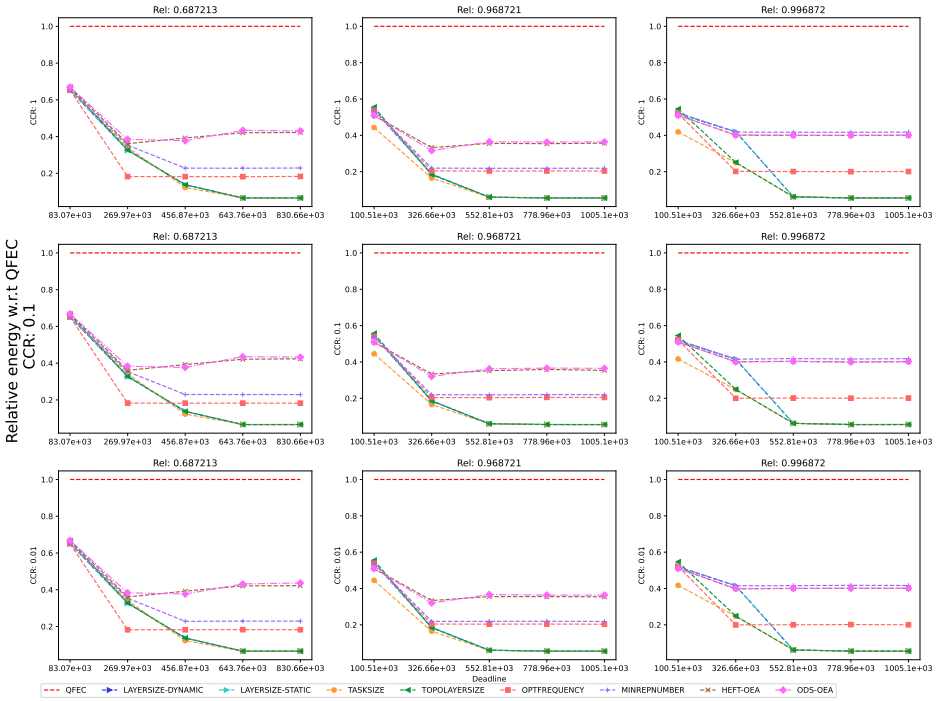

Figure 118: Performance of the different heuristics on the BLAST workflow.

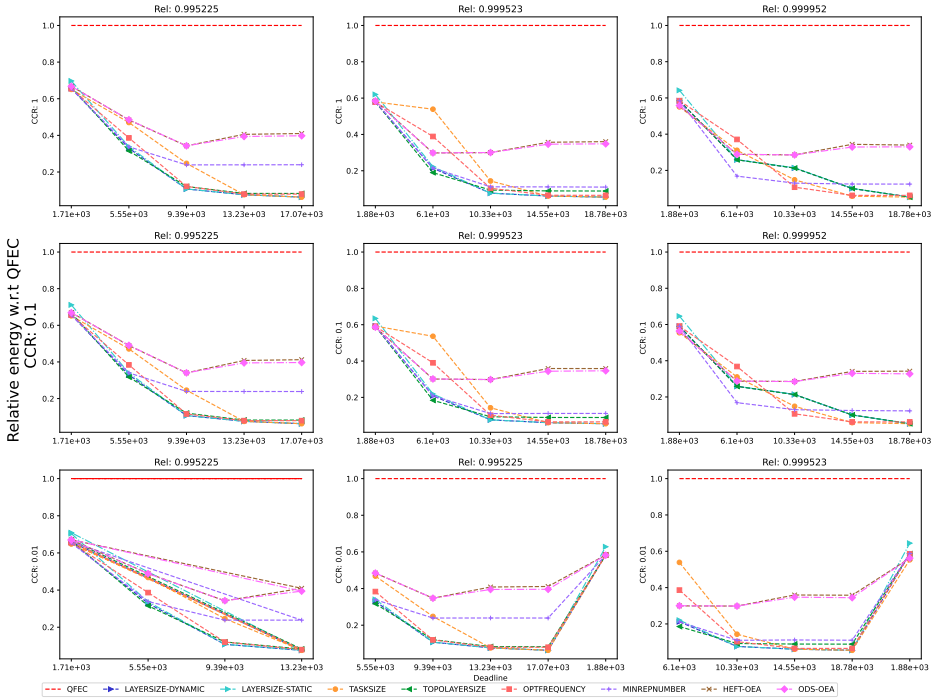

Figure 119: Performance of the different heuristics on the BWA workflow.

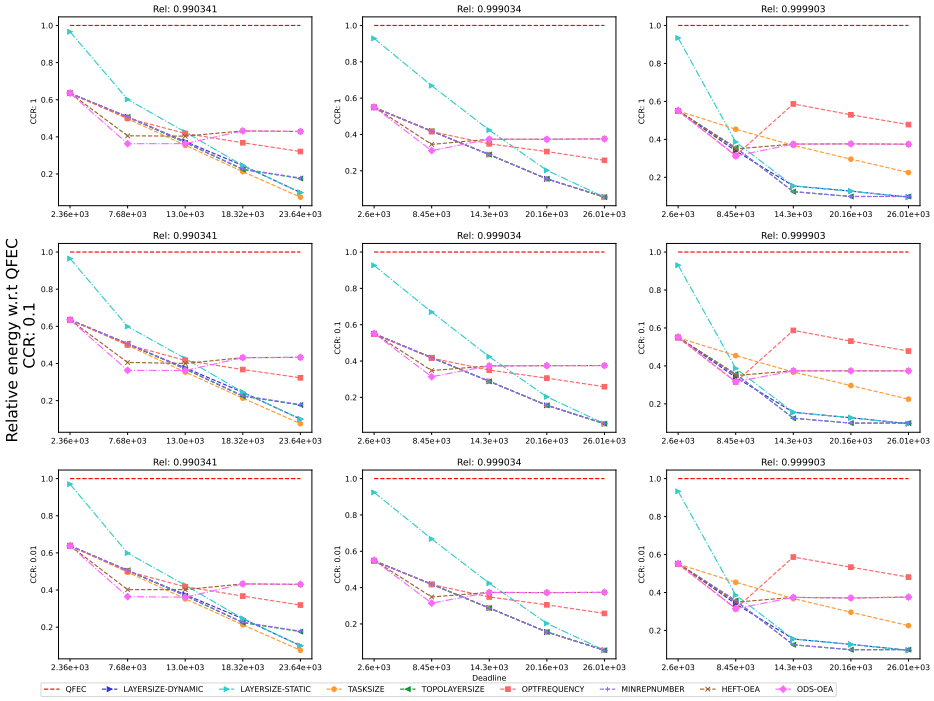

Figure 120: Performance of the different heuristics on the Cholesky workflow.

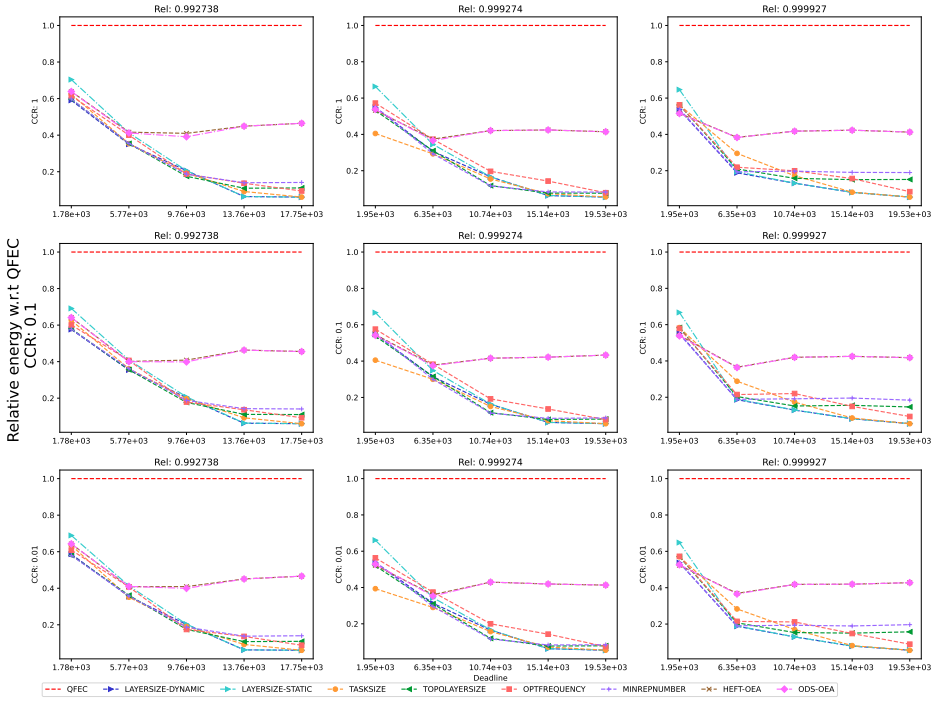

Figure 121: Performance of the different heuristics on the Cycles workflow.

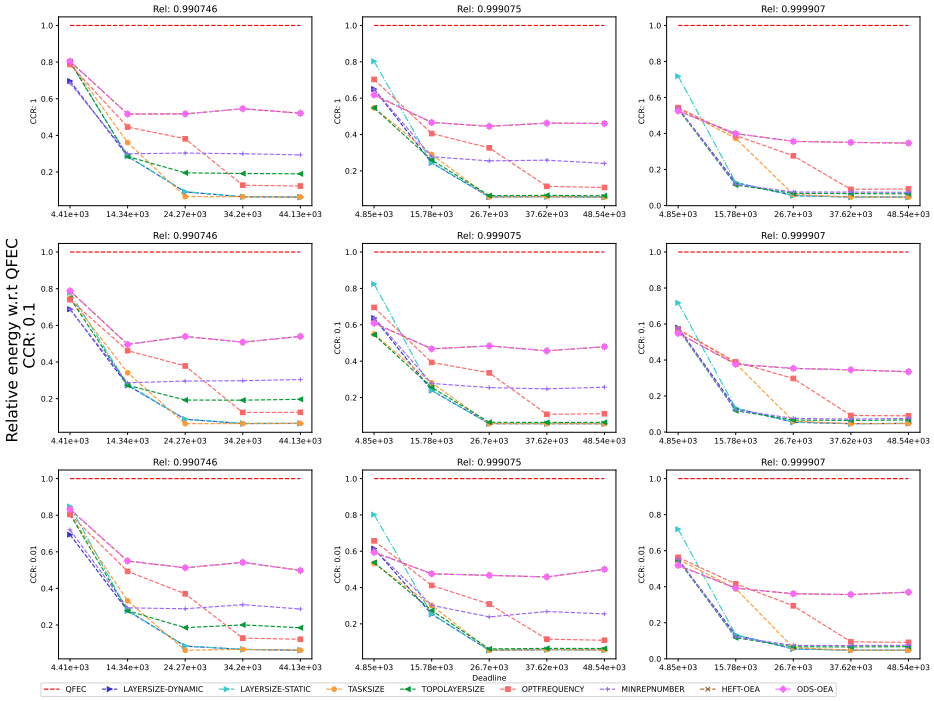

Figure 122: Performance of the different heuristics on the Epigenomics workflow.

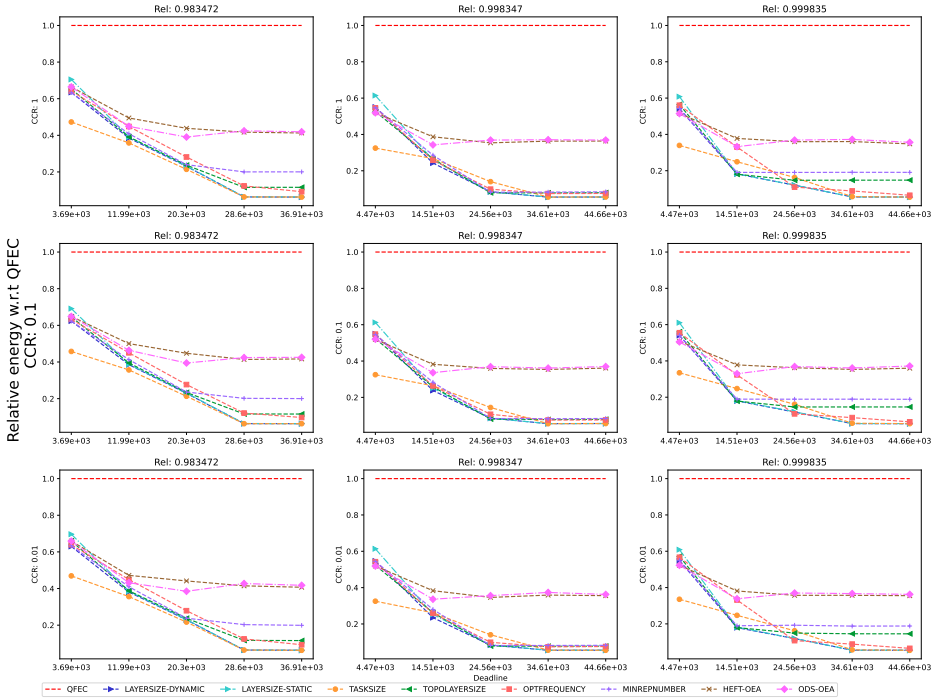

Figure 123: Performance of the different heuristics on the Genome workflow.

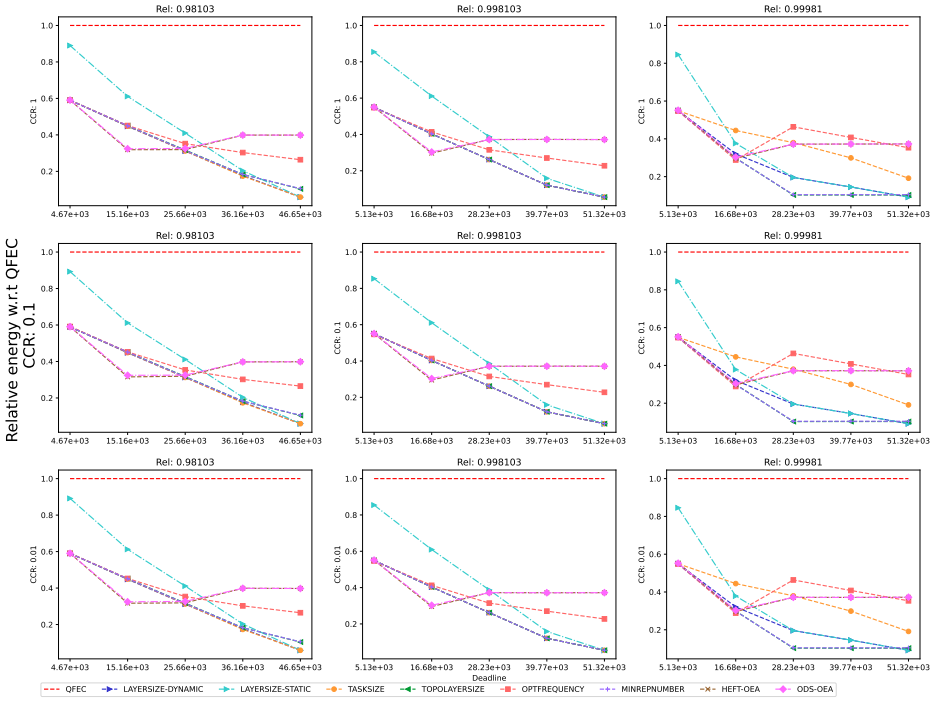

Figure 124: Performance of the different heuristics on the LU workflow.

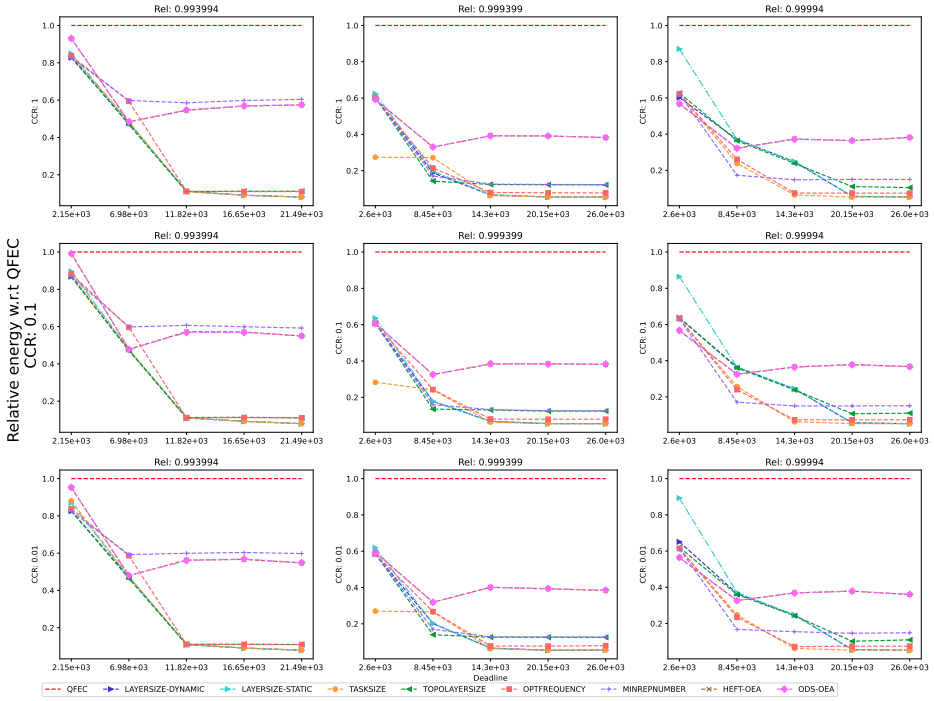

Figure 125: Performance of the different heuristics on the Montage workflow.

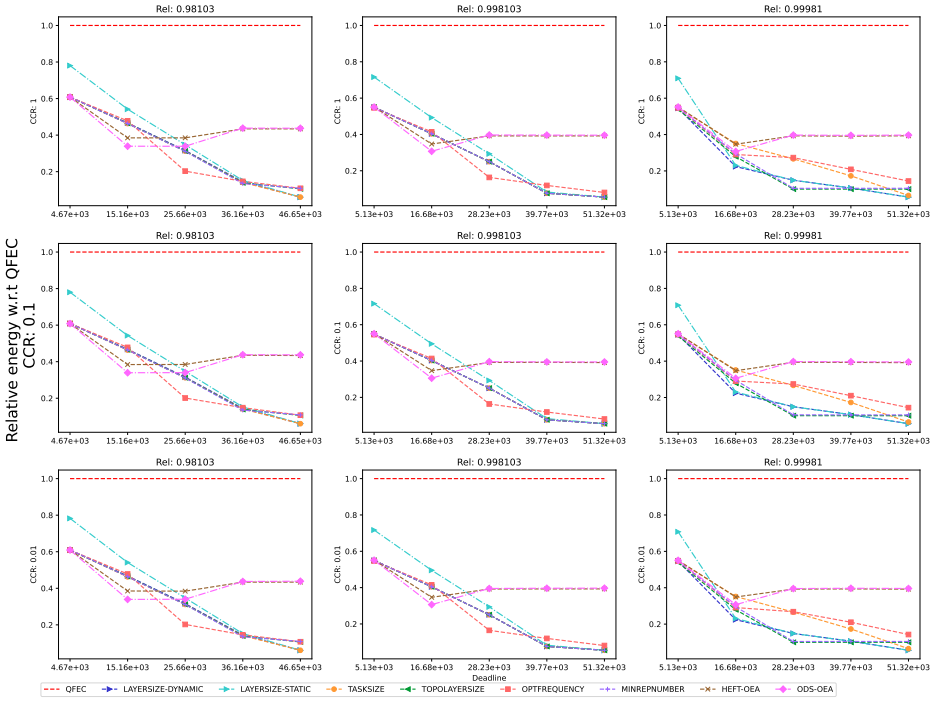

Figure 126: Performance of the different heuristics on the QR workflow.

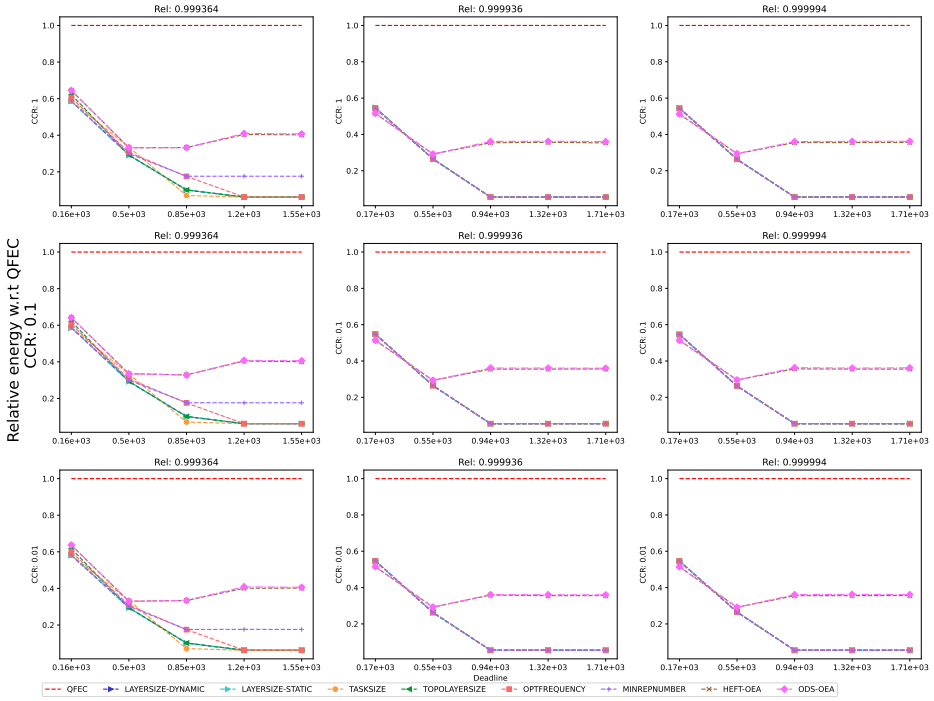

Figure 127: Performance of the different heuristics on the Seismology workflow.

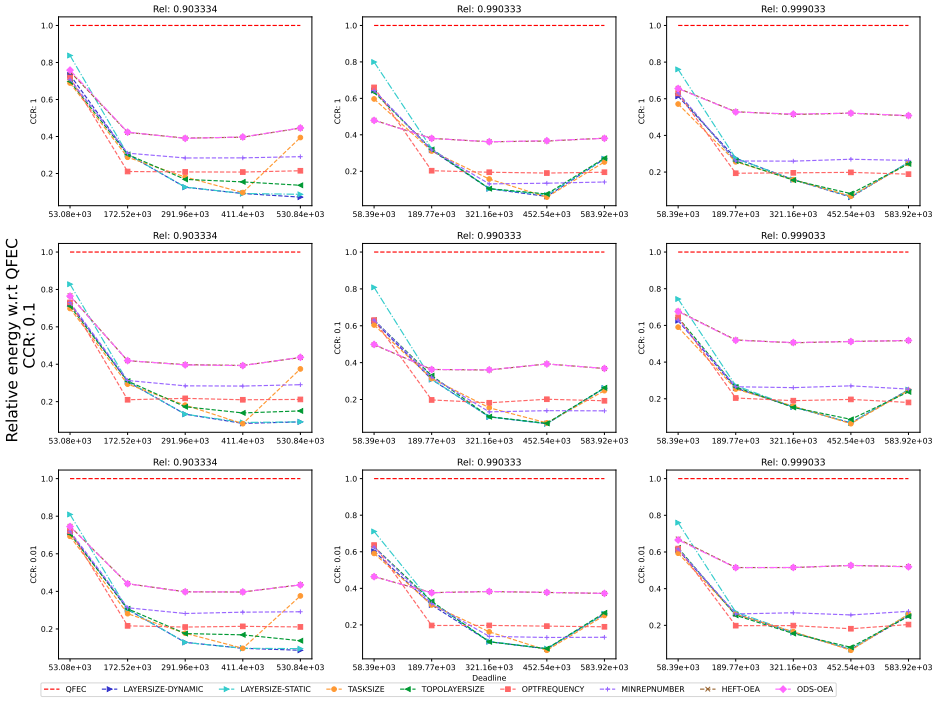

Figure 128: Performance of the different heuristics on the SoyKB workflow.

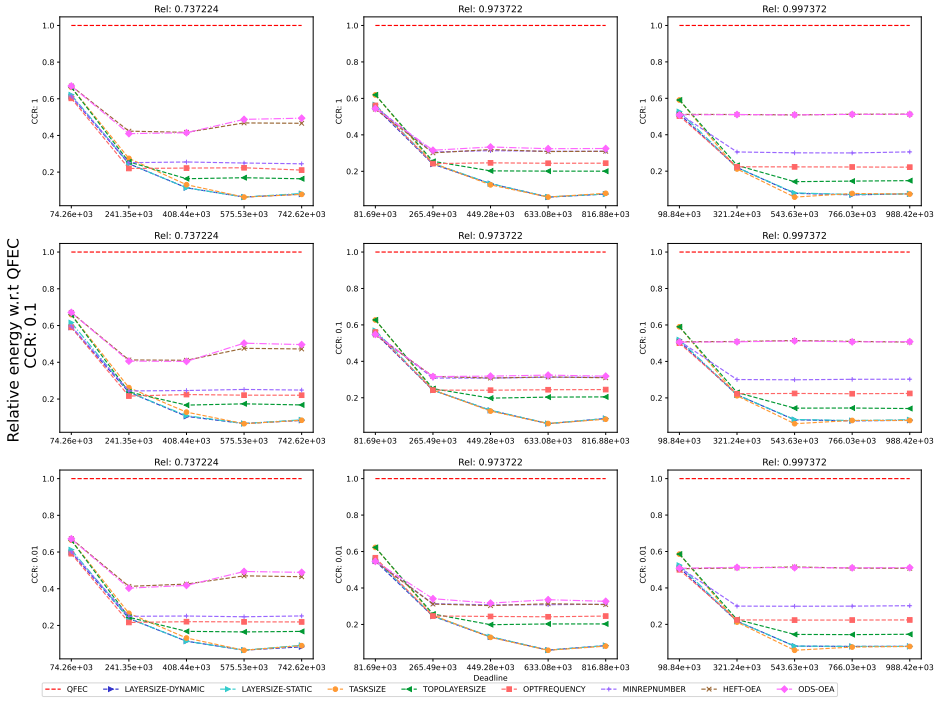

Figure 129: Performance of the different heuristics on the SRASearch.

D.3.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

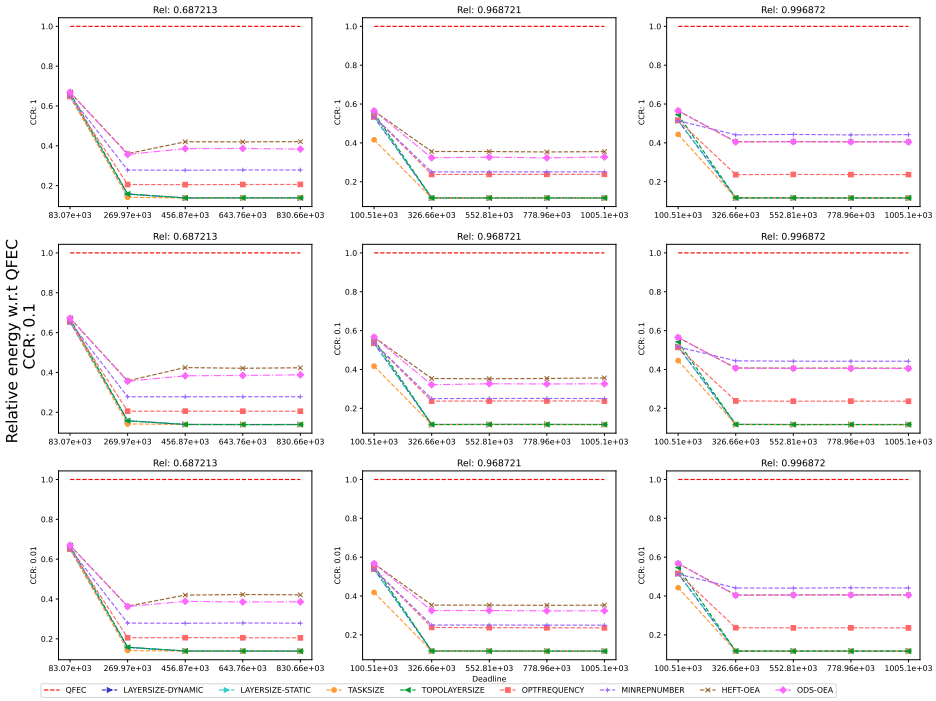

Figure 130: Performance of the different heuristics on the BLAST workflow.

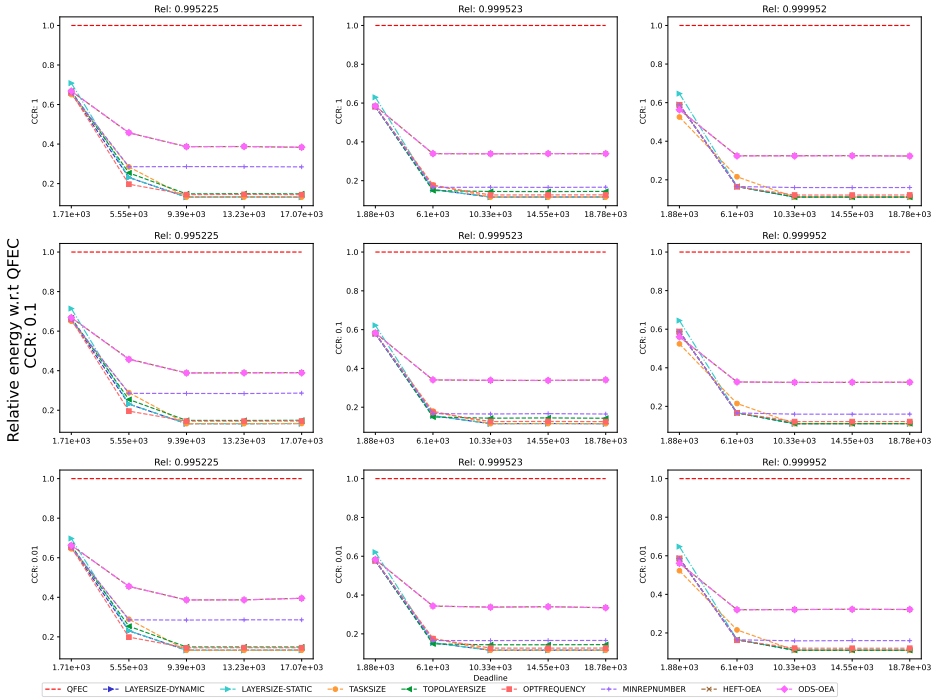

Figure 131: Performance of the different heuristics on the BWA workflow.

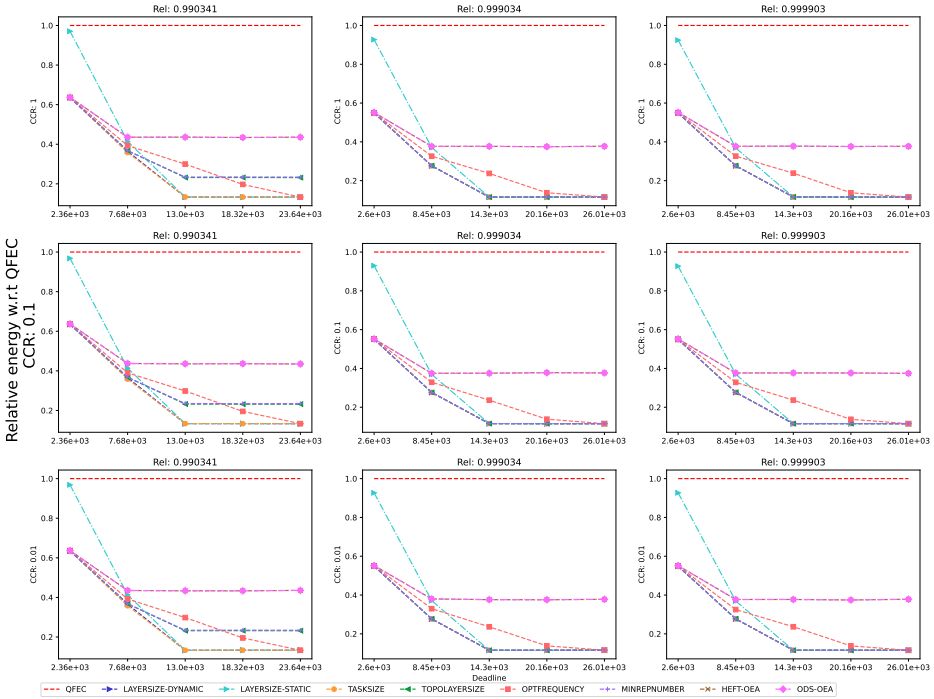

Figure 132: Performance of the different heuristics on the Cholesky workflow.

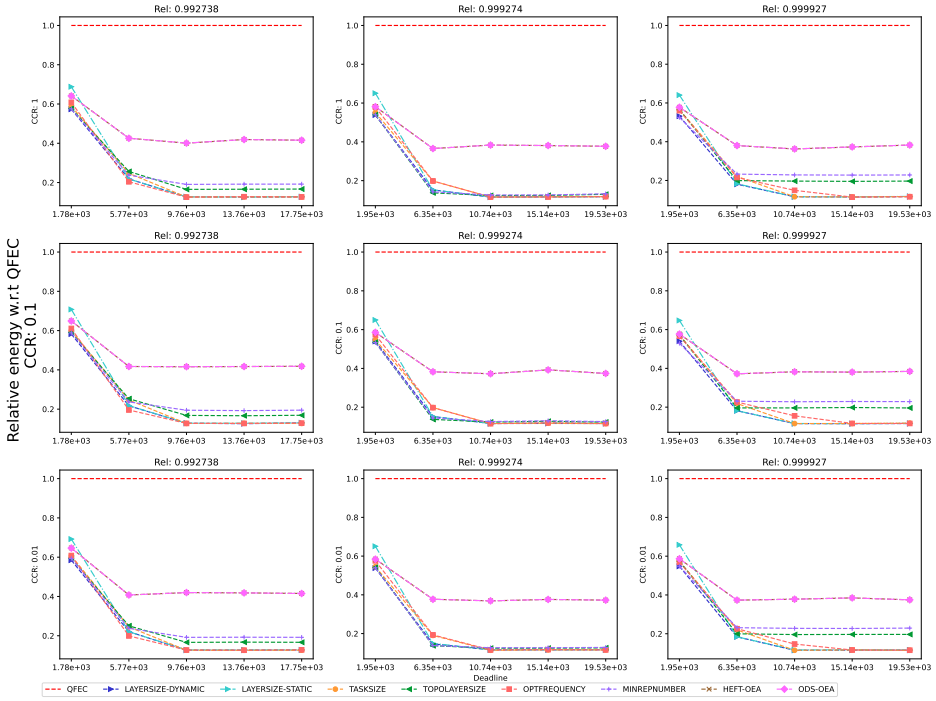

Figure 133: Performance of the different heuristics on the Cycles workflow.

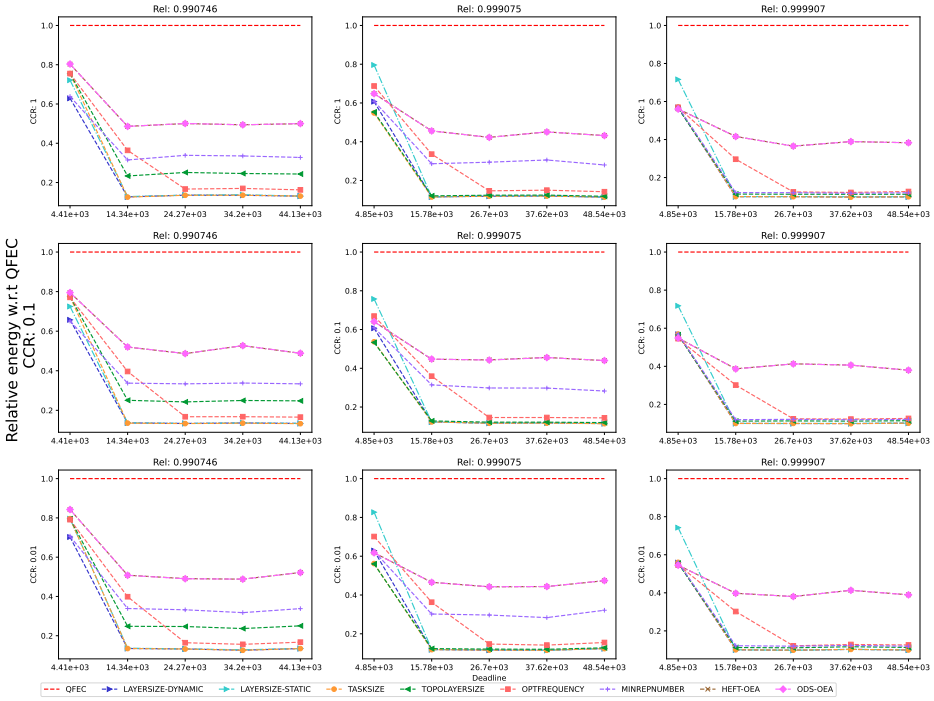

Figure 134: Performance of the different heuristics on the Epigenomics workflow.

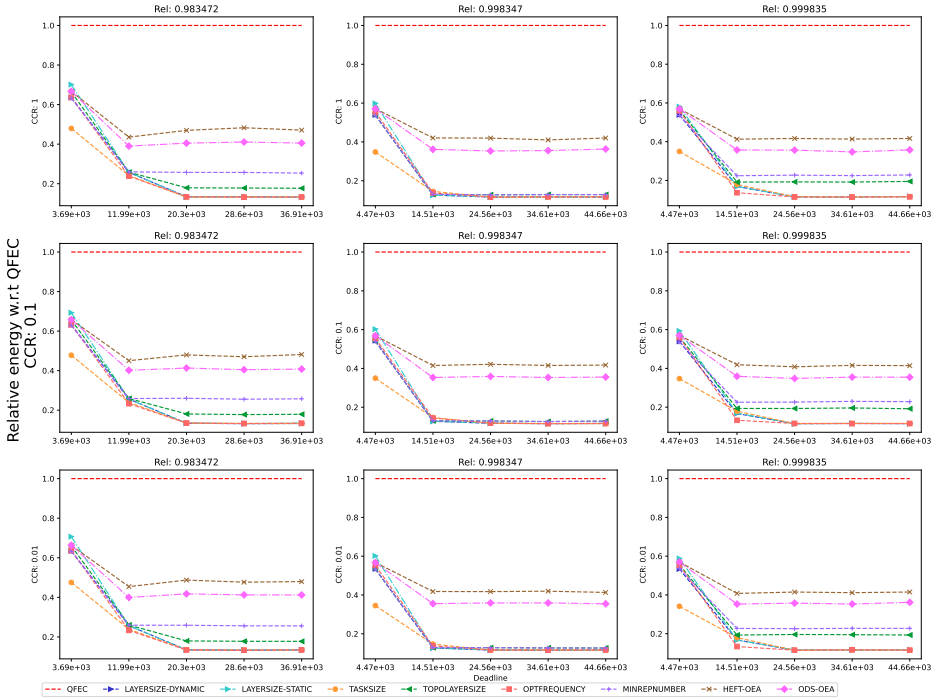

Figure 135: Performance of the different heuristics on the Genome workflow.

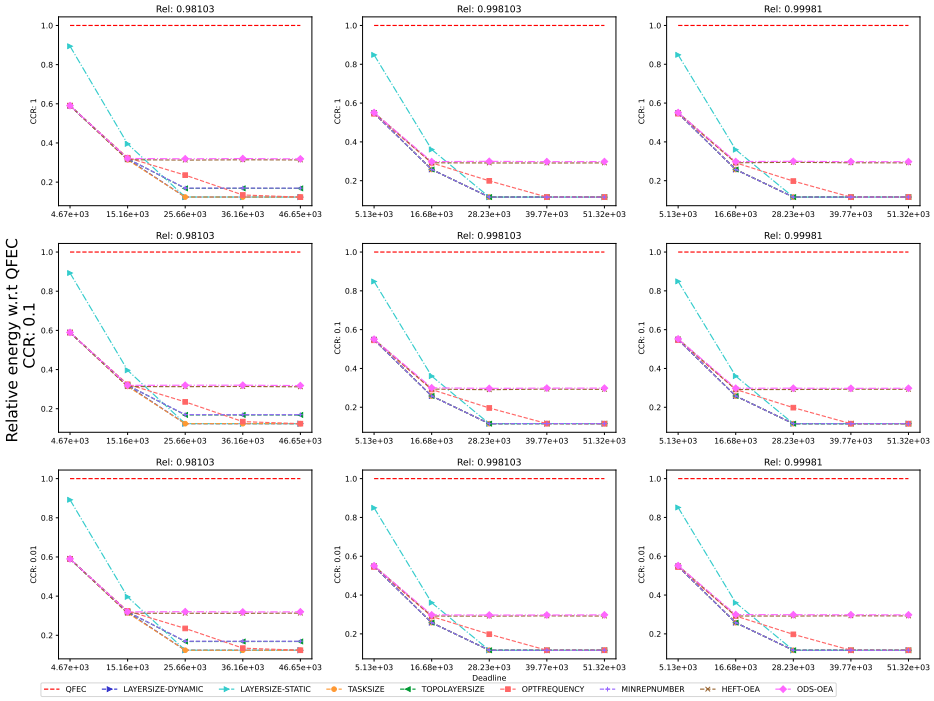

Figure 136: Performance of the different heuristics on the LU workflow.

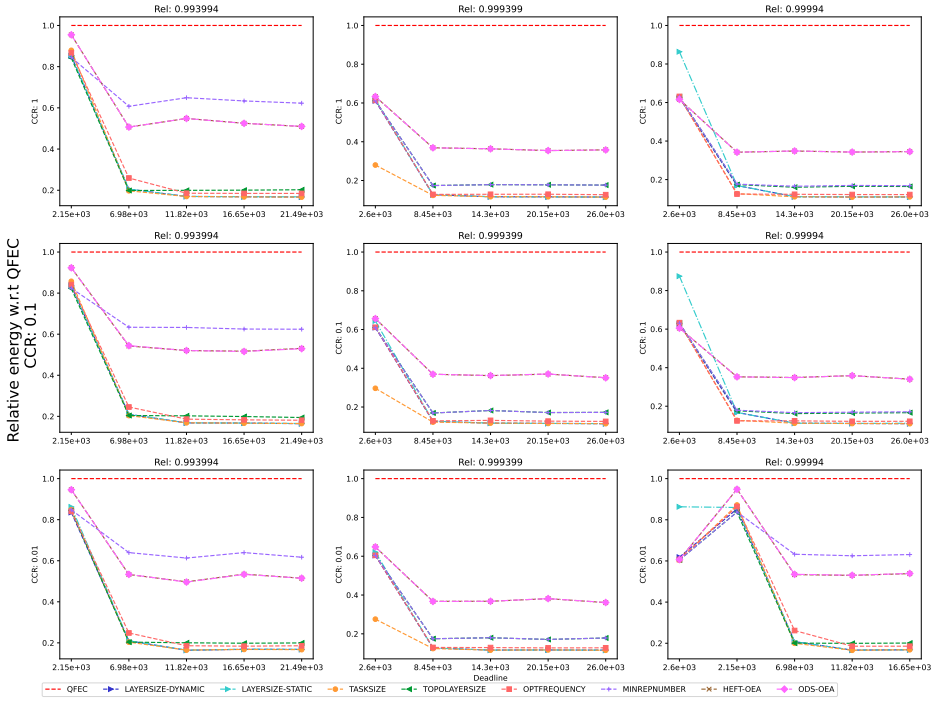

Figure 137: Performance of the different heuristics on the Montage workflow.

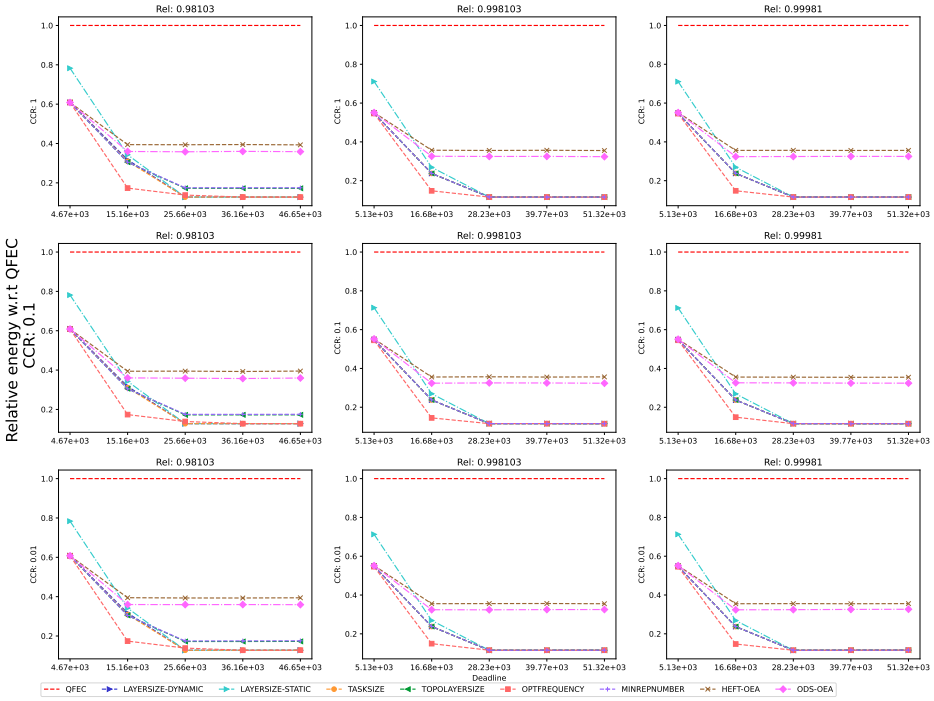

Figure 138: Performance of the different heuristics on the QR workflow.

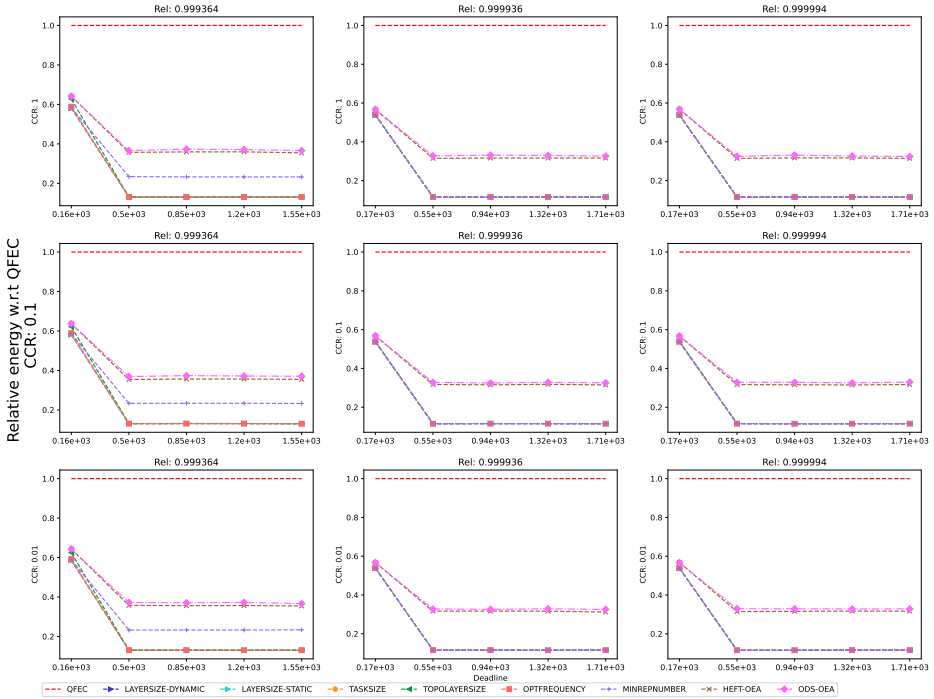

Figure 139: Performance of the different heuristics on the Seismology workflow.

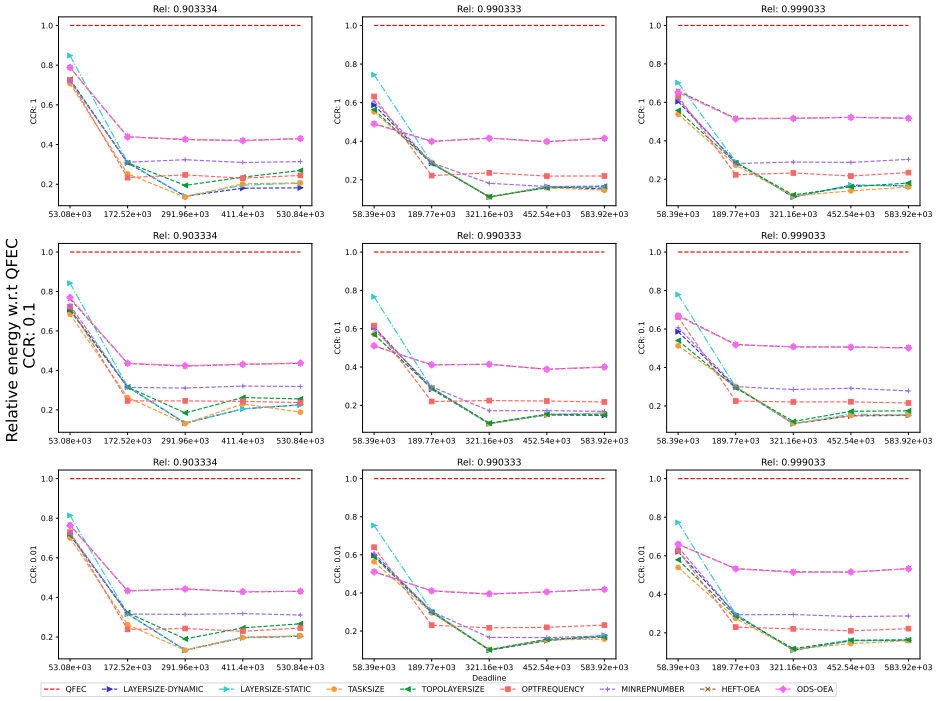

Figure 140: Performance of the different heuristics on the SoyKB workflow.

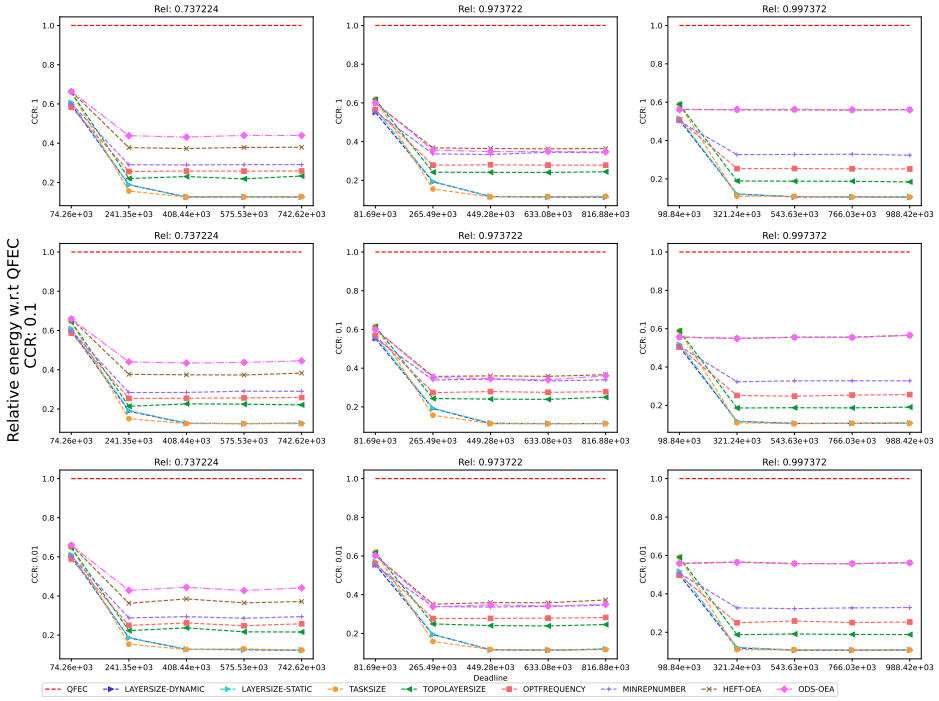

Figure 141: Performance of the different heuristics on the SRASearch.

D.3.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

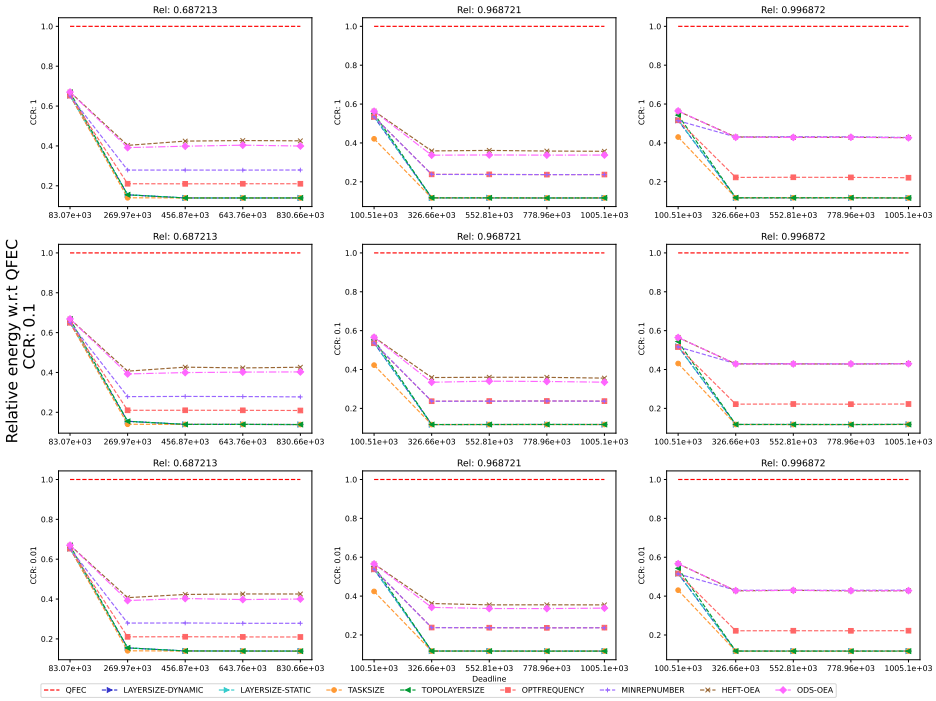

Figure 142: Performance of the different heuristics on the BLAST workflow.

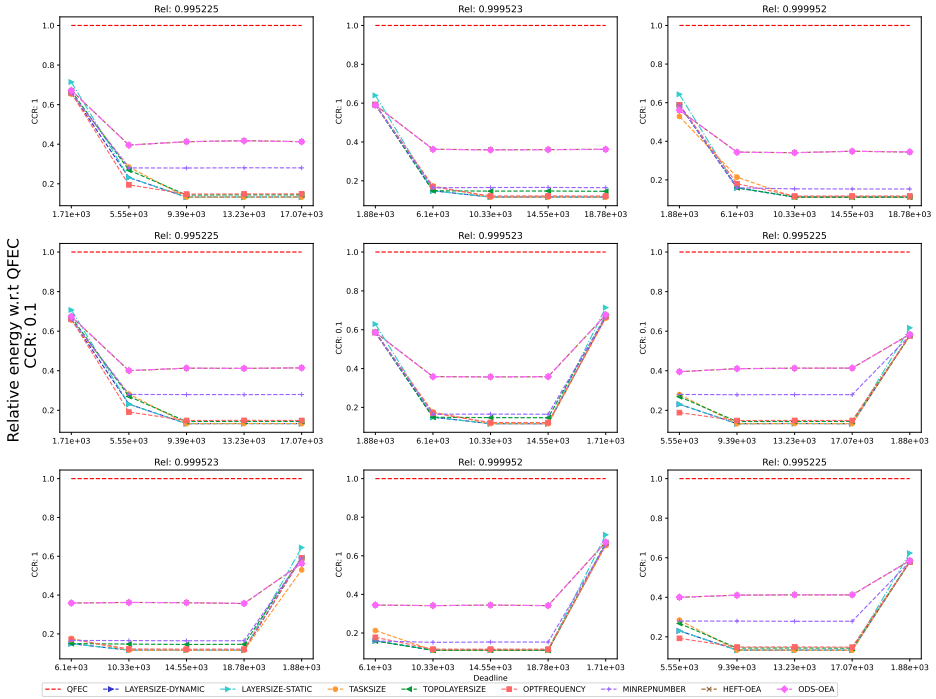

Figure 143: Performance of the different heuristics on the BWA workflow.

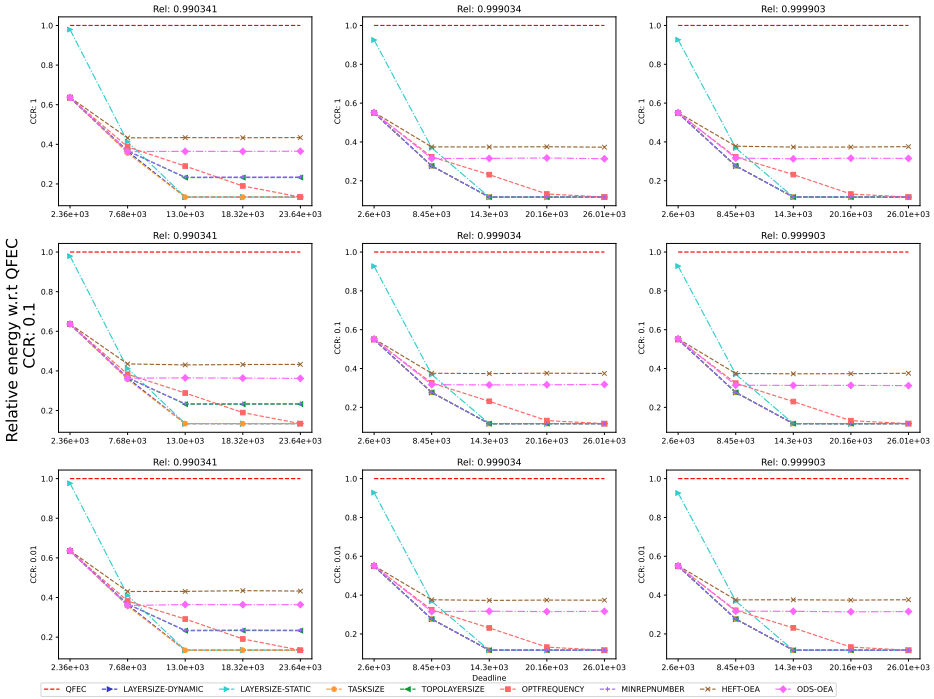

Figure 144: Performance of the different heuristics on the Cholesky workflow.

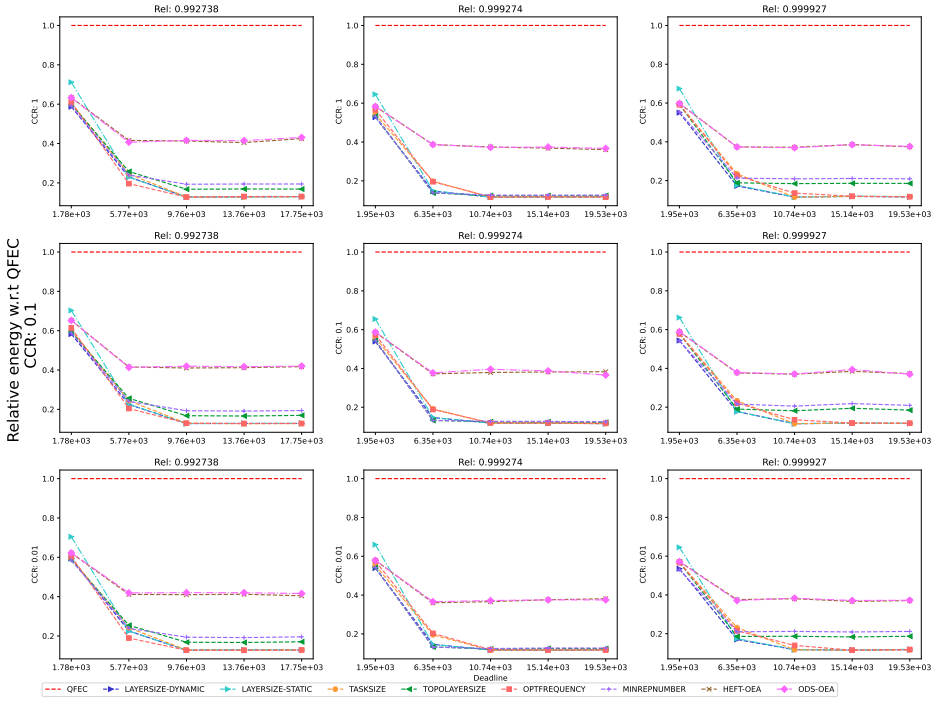

Figure 145: Performance of the different heuristics on the Cycles workflow.

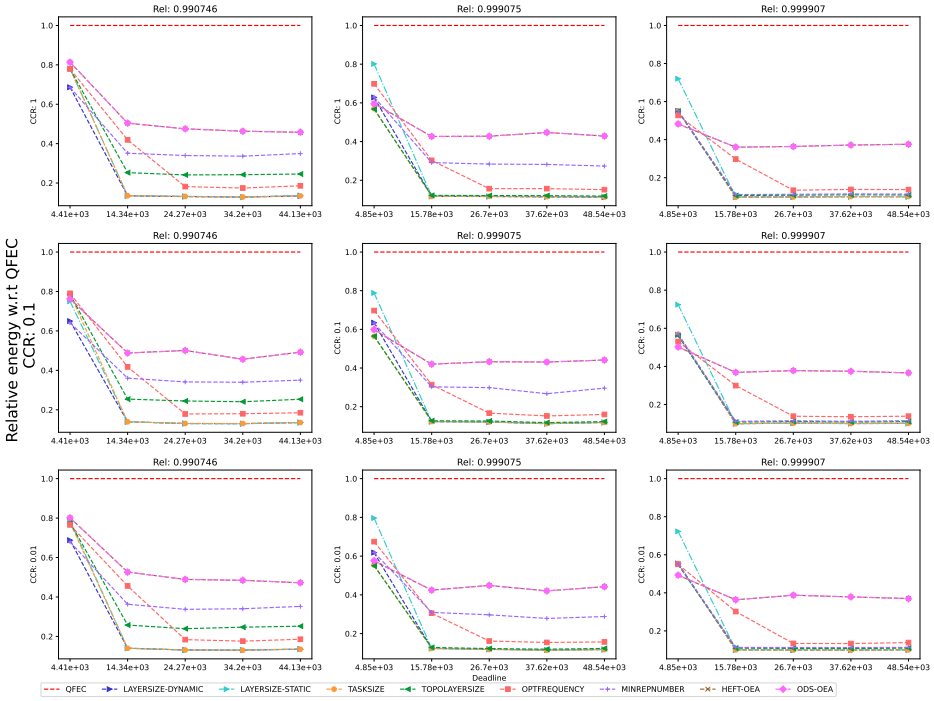

Figure 146: Performance of the different heuristics on the Epigenomics workflow.

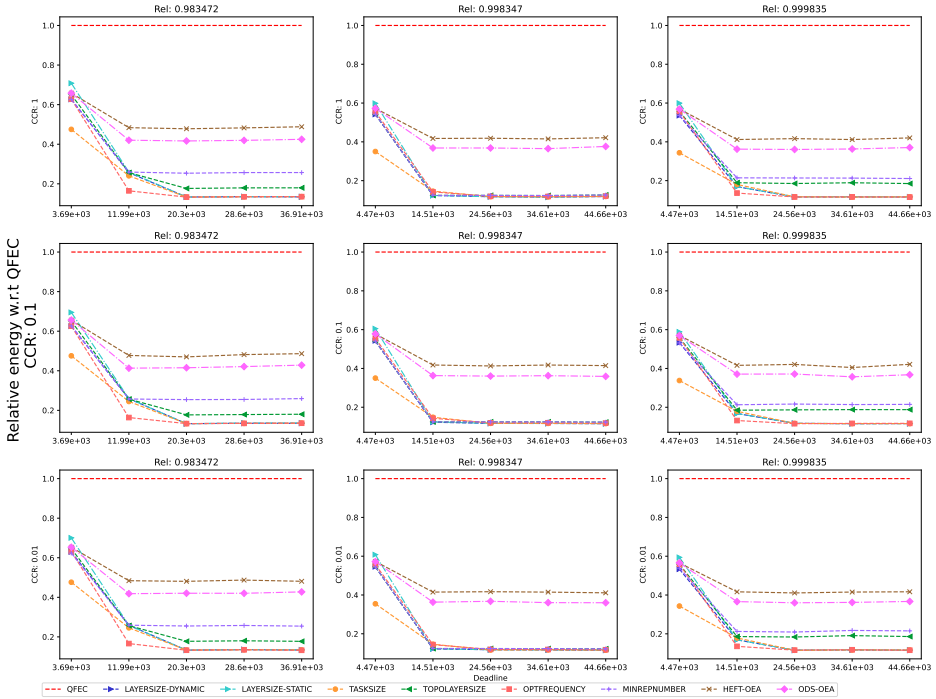

Figure 147: Performance of the different heuristics on the Genome workflow.

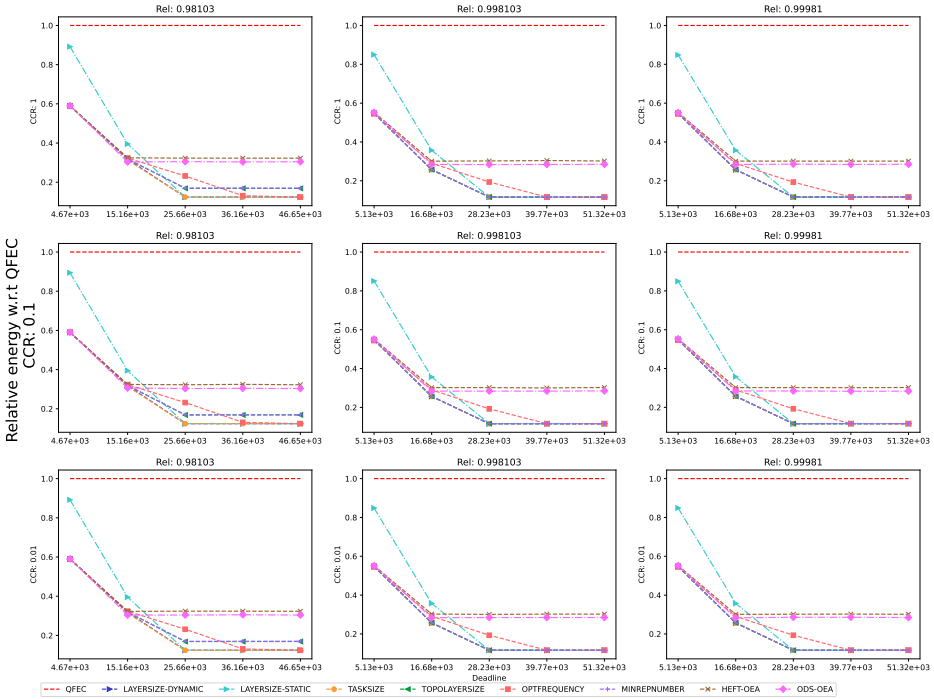

Figure 148: Performance of the different heuristics on the LU workflow.

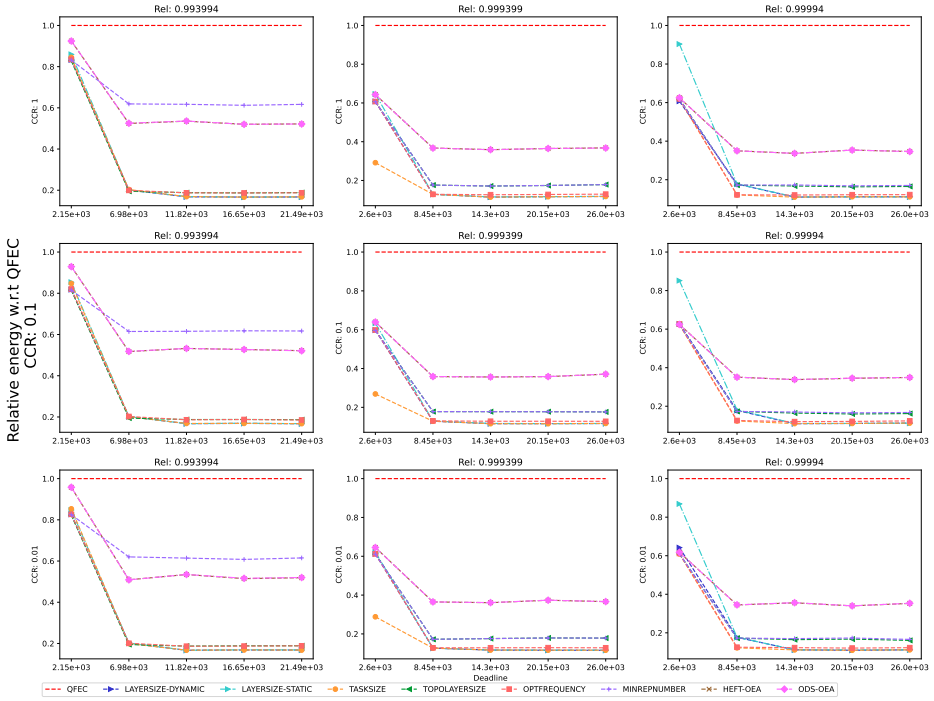

Figure 149: Performance of the different heuristics on the Montage workflow.

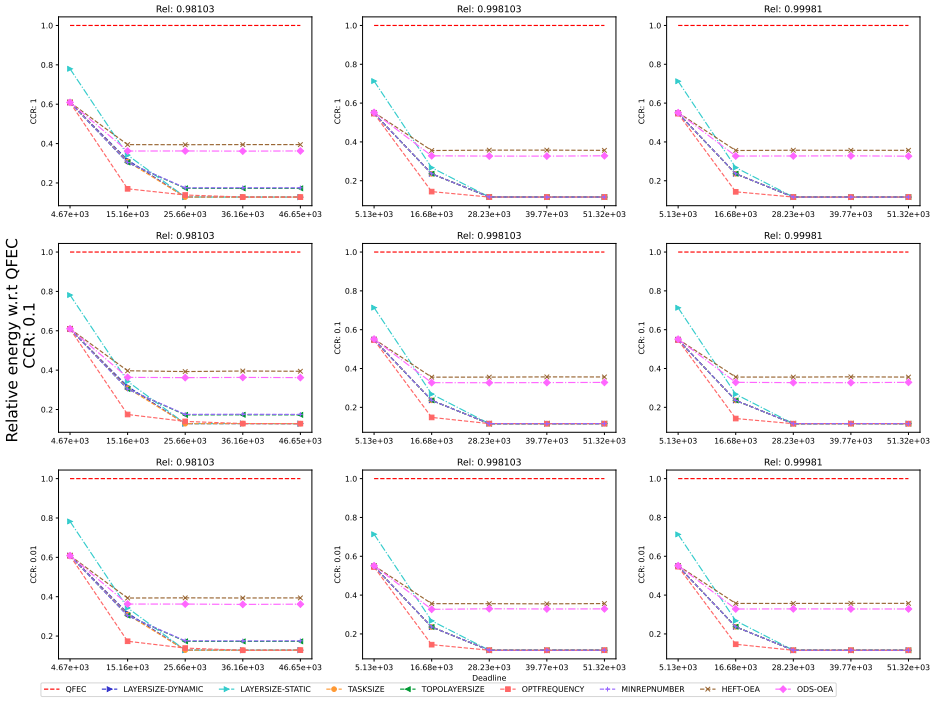

Figure 150: Performance of the different heuristics on the QR workflow.

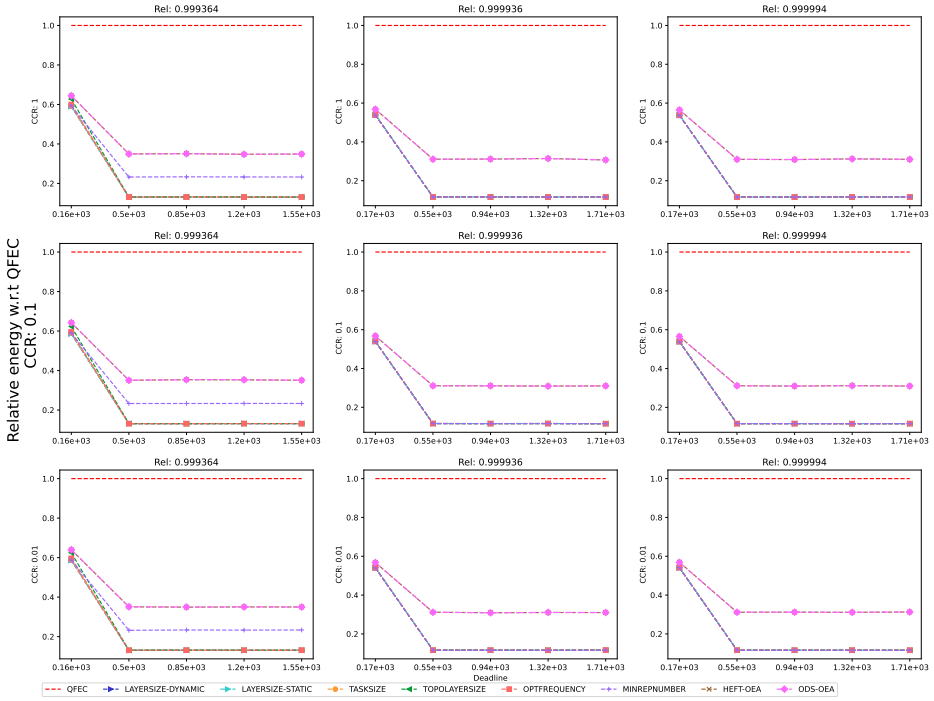

Figure 151: Performance of the different heuristics on the Seismology workflow.

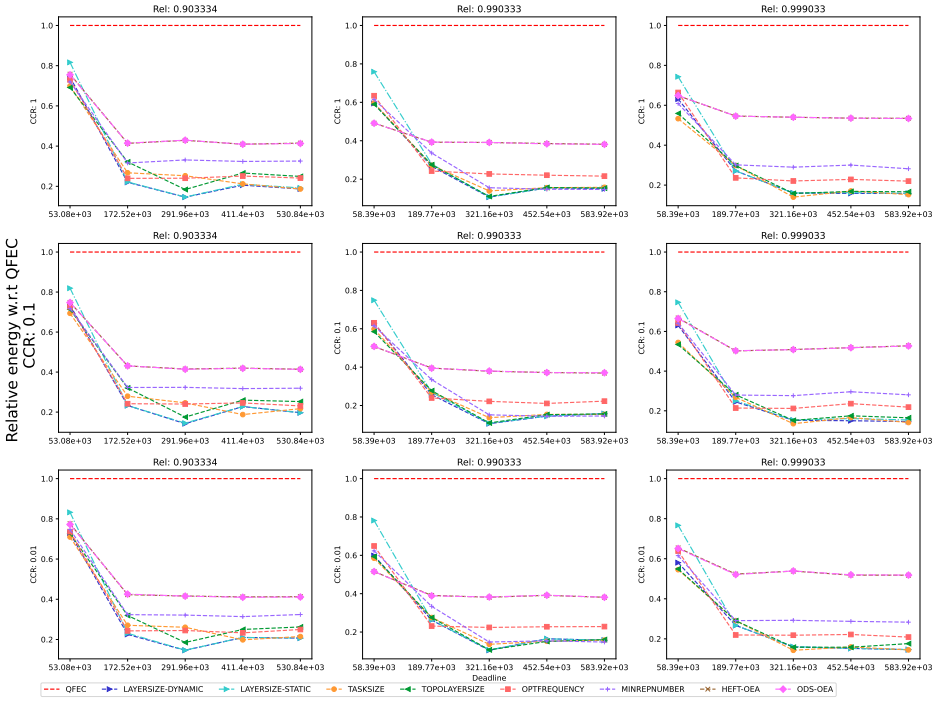

Figure 152: Performance of the different heuristics on the SoyKB workflow.

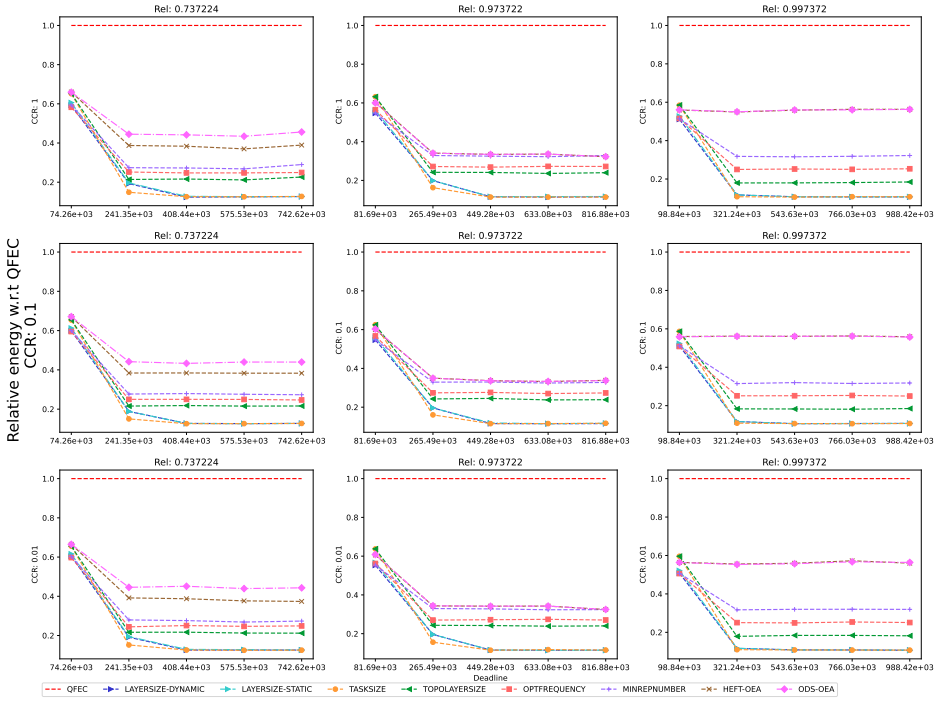

Figure 153: Performance of the different heuristics on the SRASearch.

D.4 $BC/WC = 0.4$

D.4.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

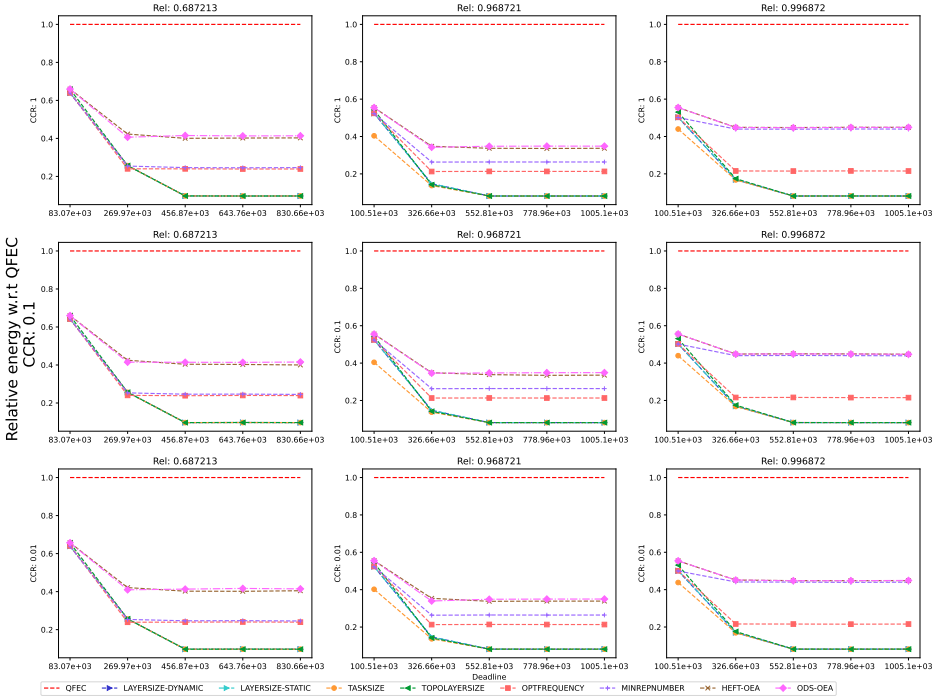

Figure 154: Performance of the different heuristics on the BLAST workflow.

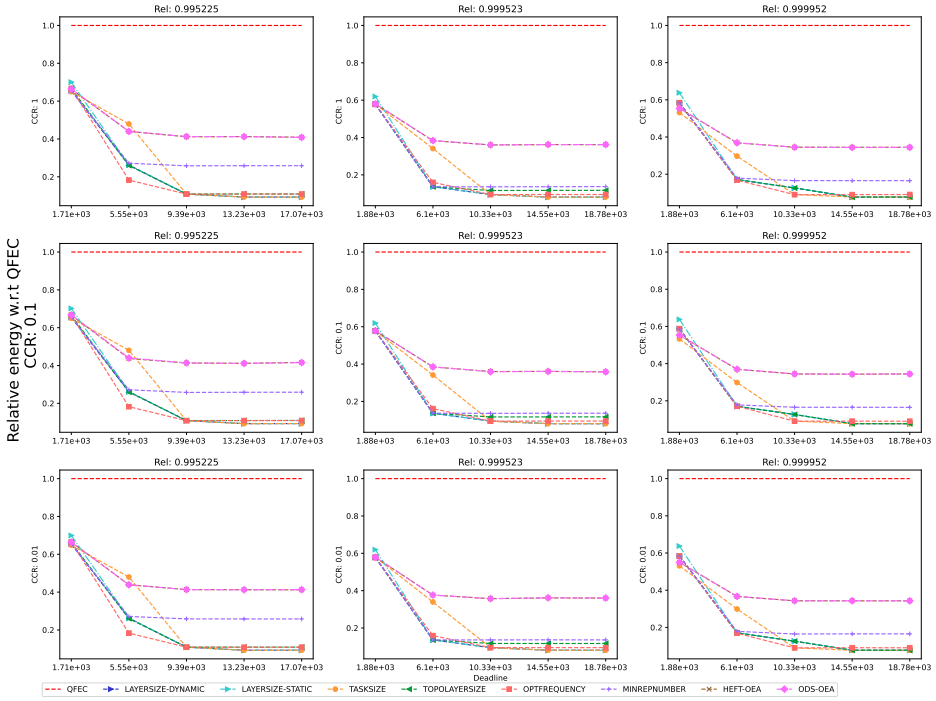

Figure 155: Performance of the different heuristics on the BWA workflow.

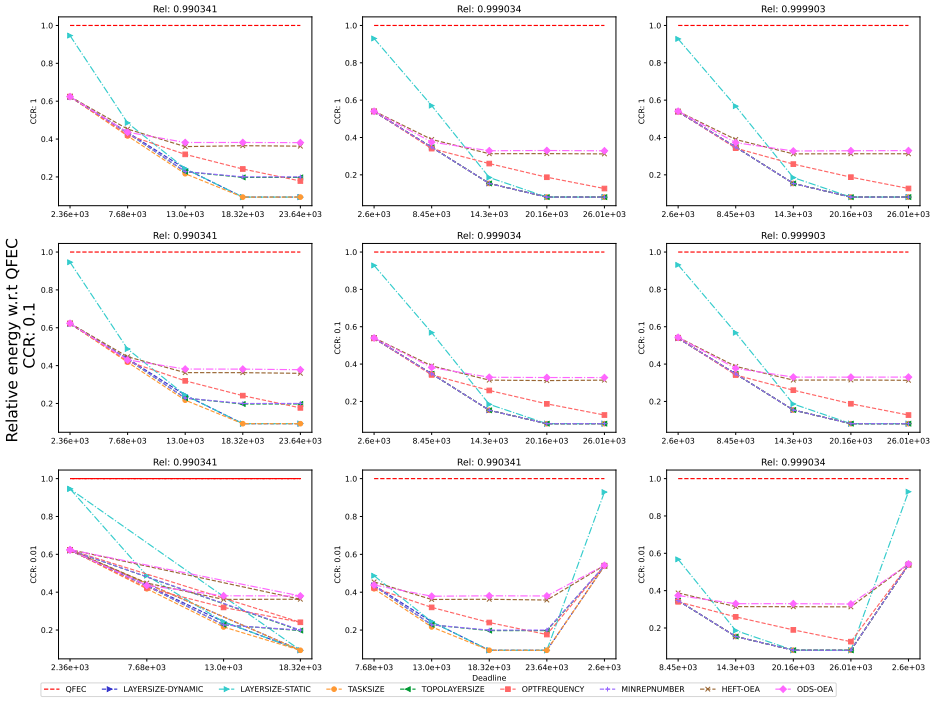

Figure 156: Performance of the different heuristics on the Cholesky workflow.

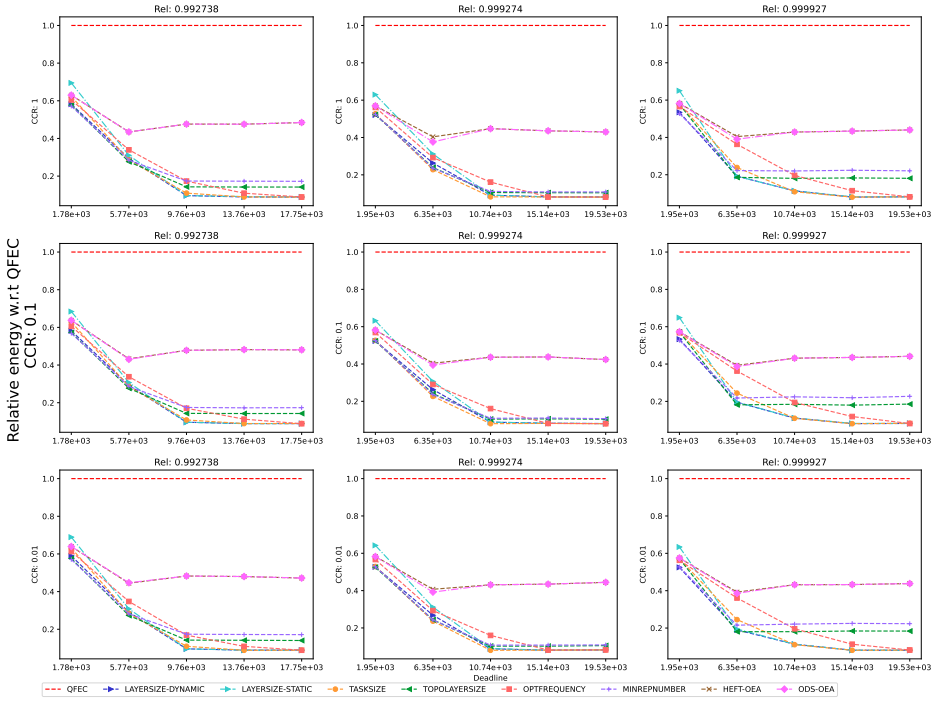

Figure 157: Performance of the different heuristics on the Cycles workflow.

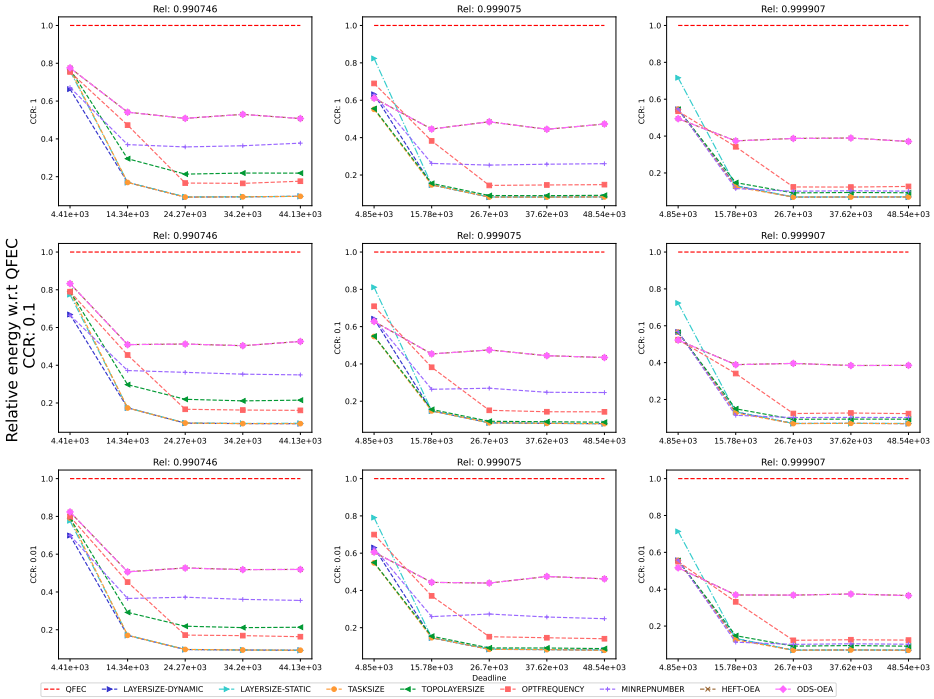

Figure 158: Performance of the different heuristics on the Epigenomics workflow.

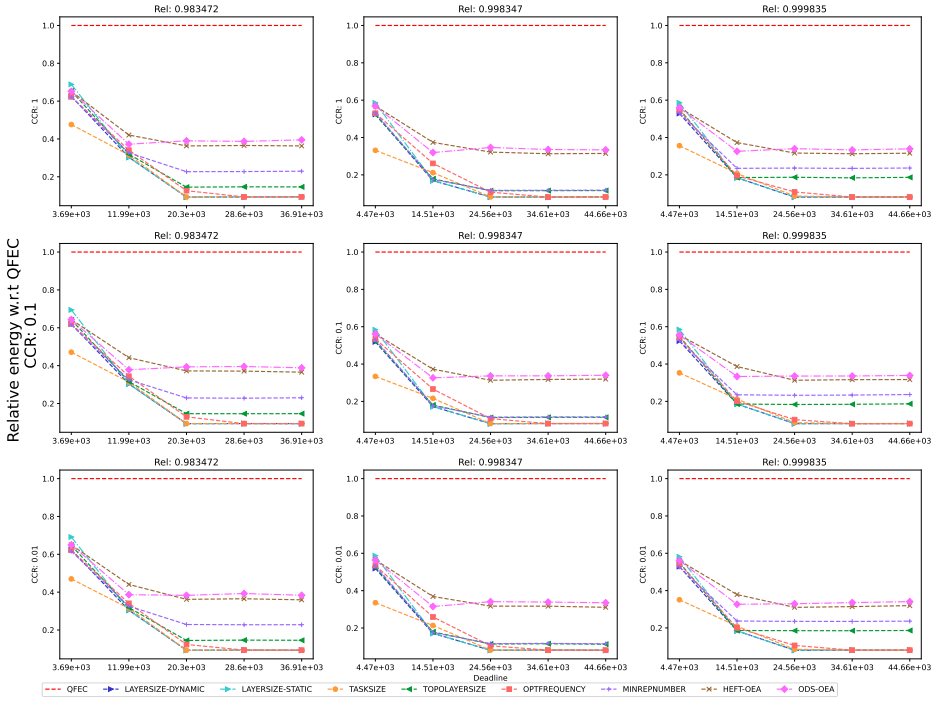

Figure 159: Performance of the different heuristics on the Genome workflow.

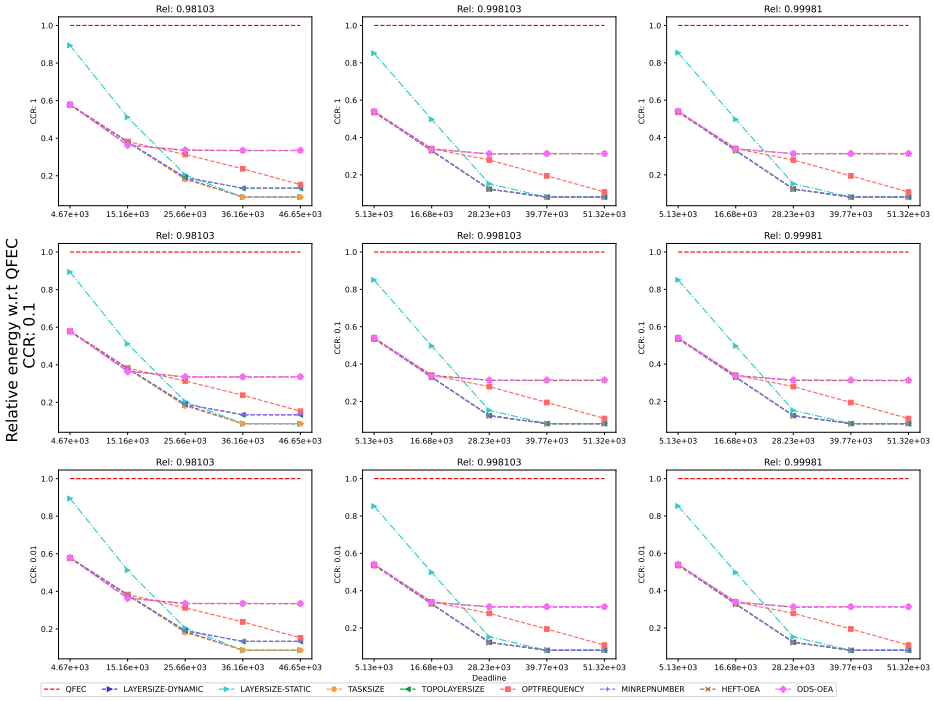

Figure 160: Performance of the different heuristics on the LU workflow.

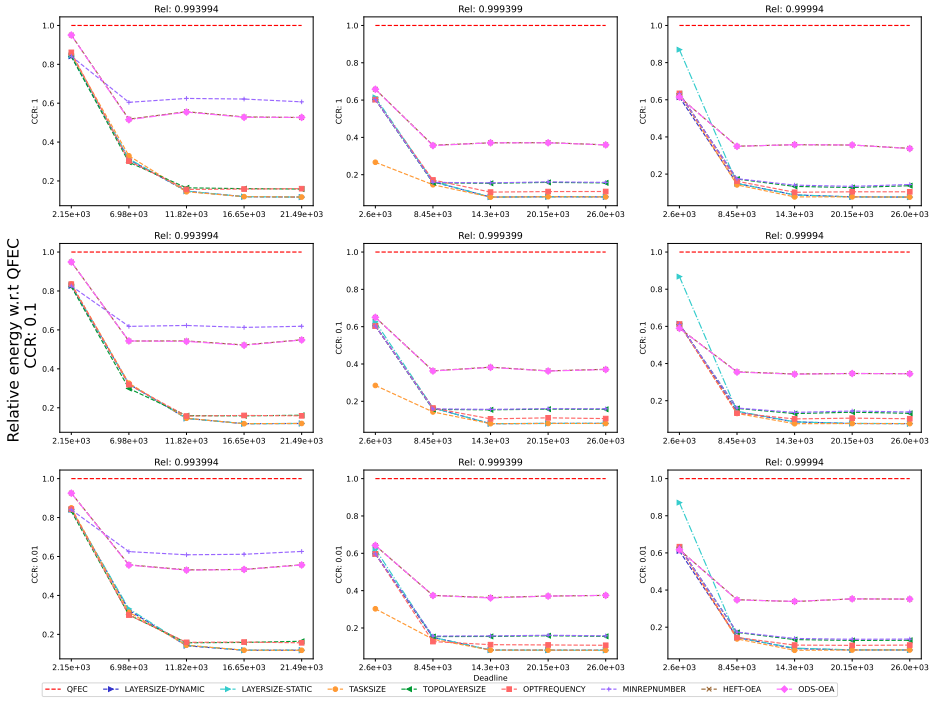

Figure 161: Performance of the different heuristics on the Montage workflow.

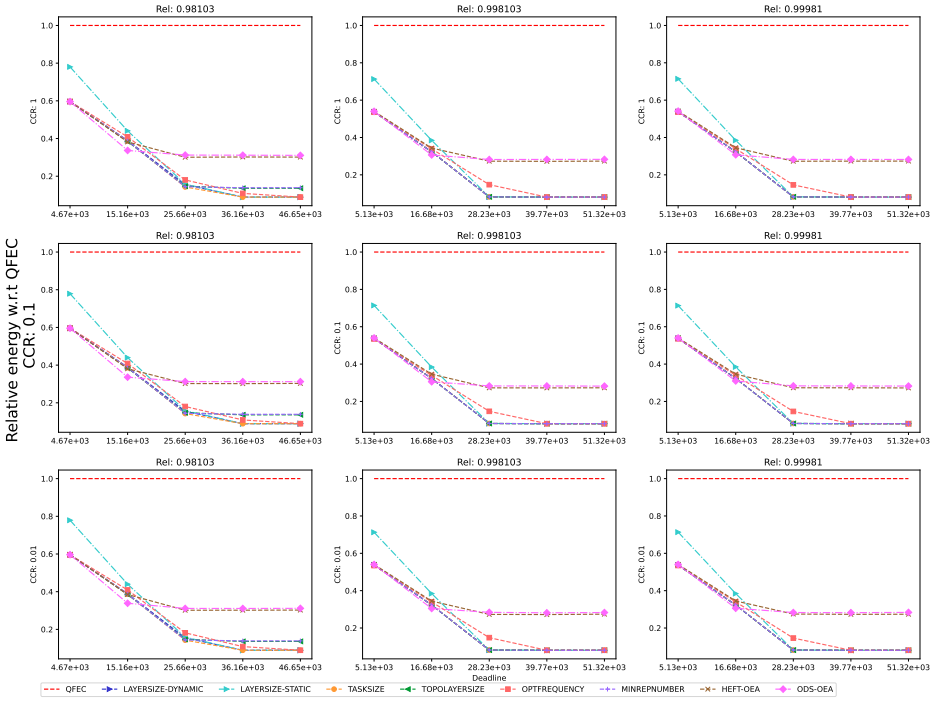

Figure 162: Performance of the different heuristics on the QR workflow.

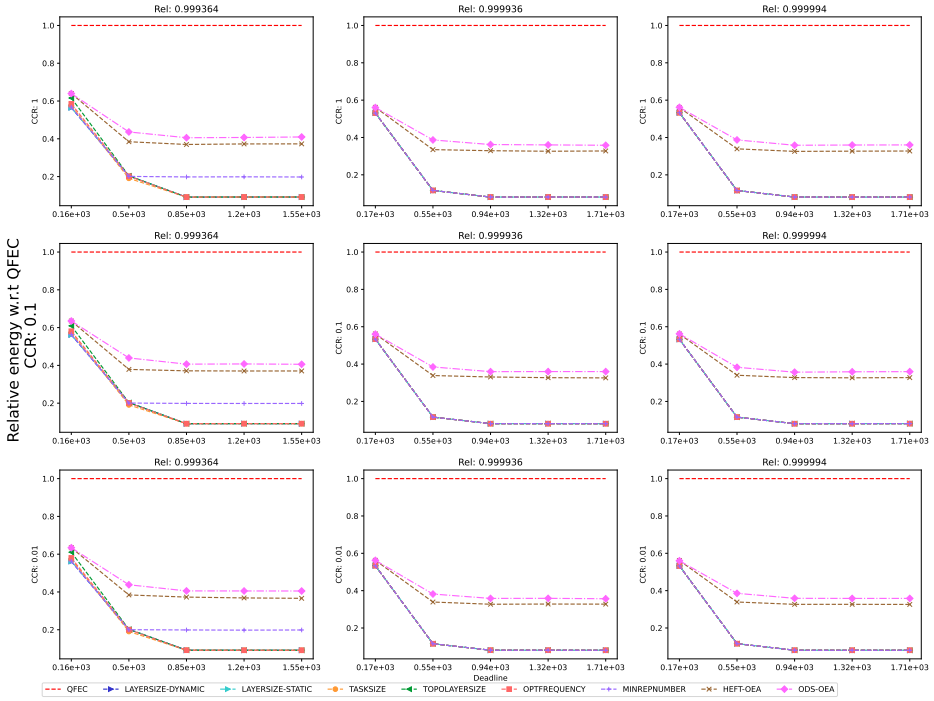

Figure 163: Performance of the different heuristics on the Seismology workflow.

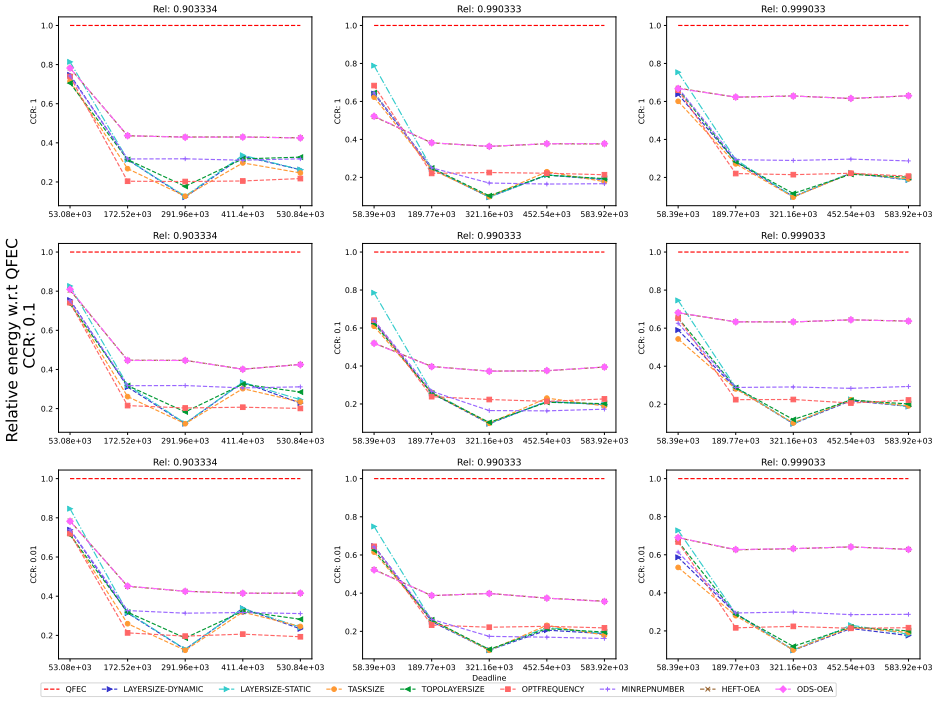

Figure 164: Performance of the different heuristics on the SoyKB workflow.

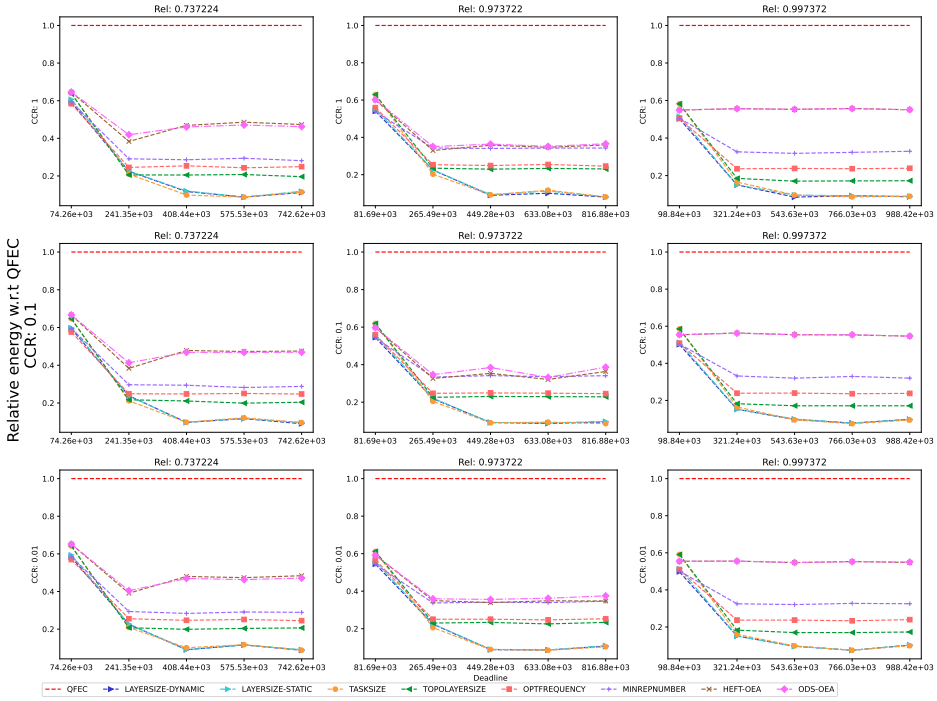

Figure 165: Performance of the different heuristics on the SRASearch.

D.4.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

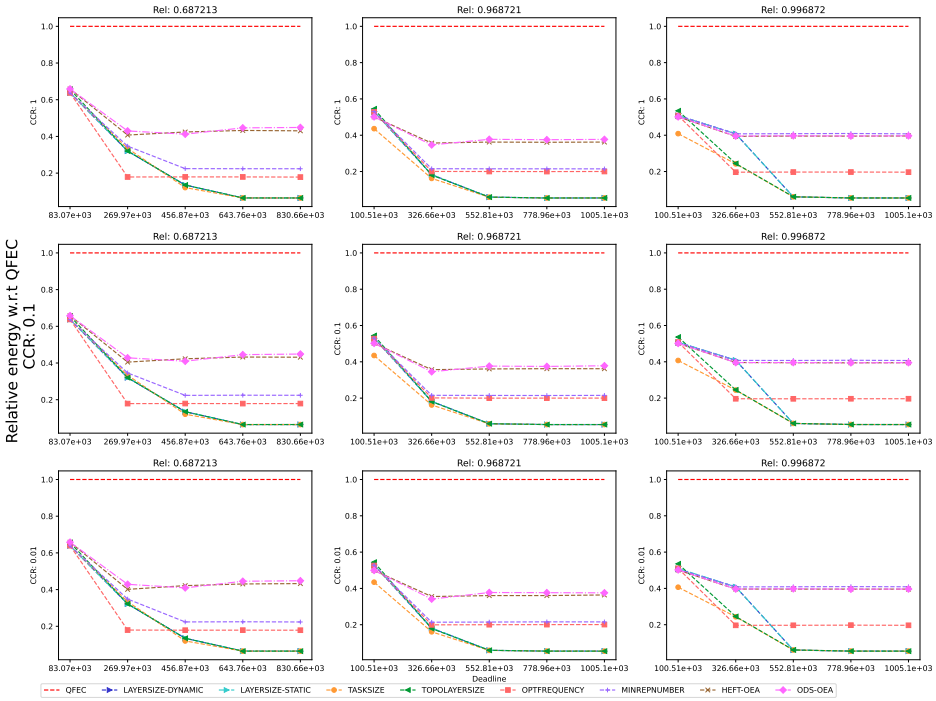

Figure 166: Performance of the different heuristics on the BLAST workflow.

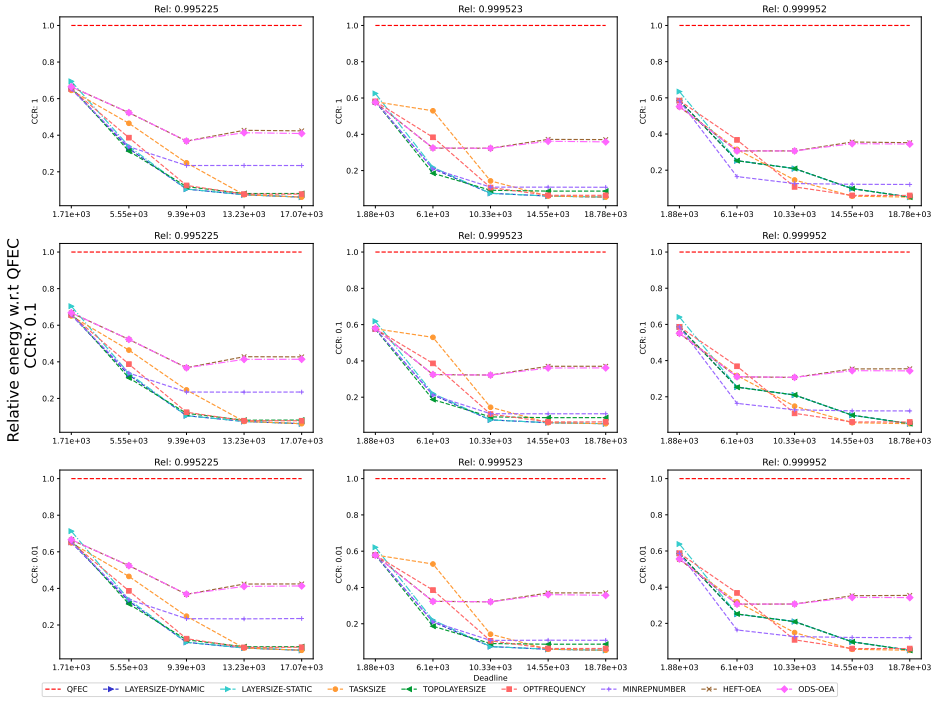

Figure 167: Performance of the different heuristics on the BWA workflow.

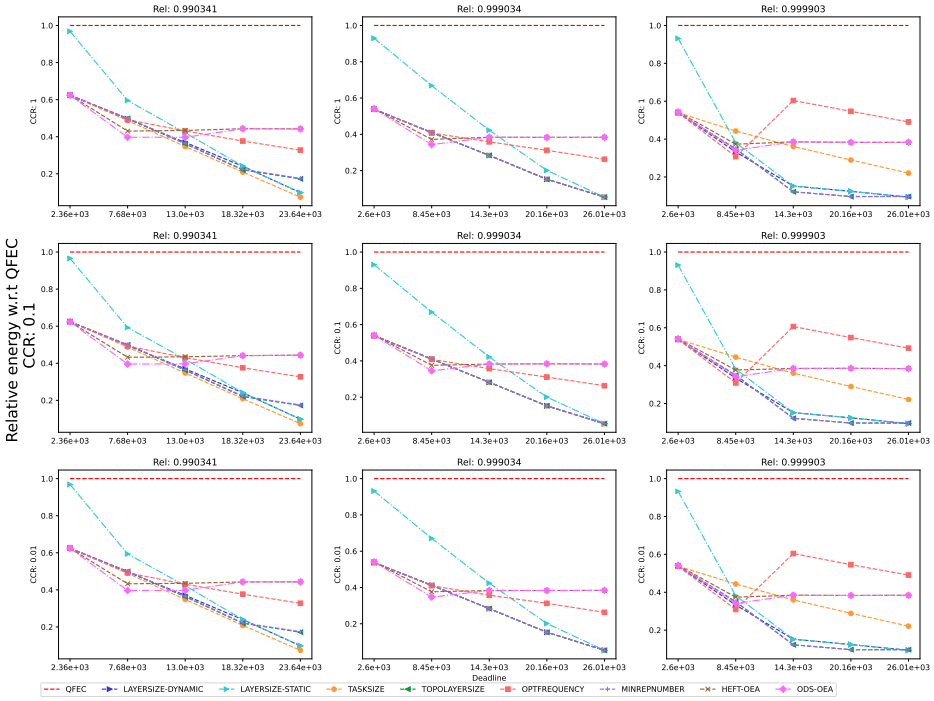

Figure 168: Performance of the different heuristics on the Cholesky workflow.

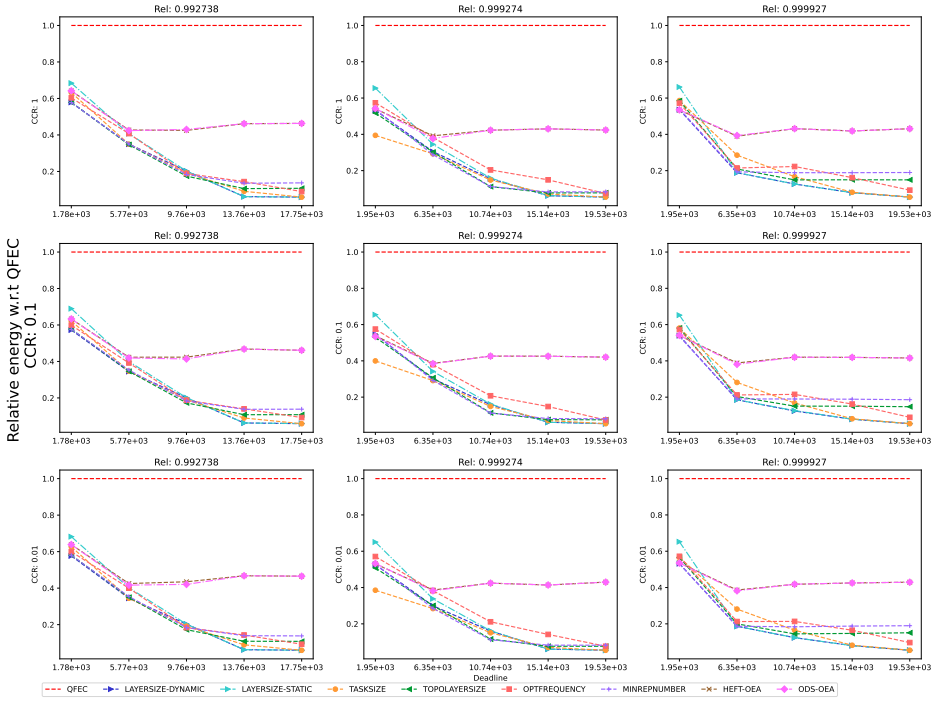

Figure 169: Performance of the different heuristics on the Cycles workflow.

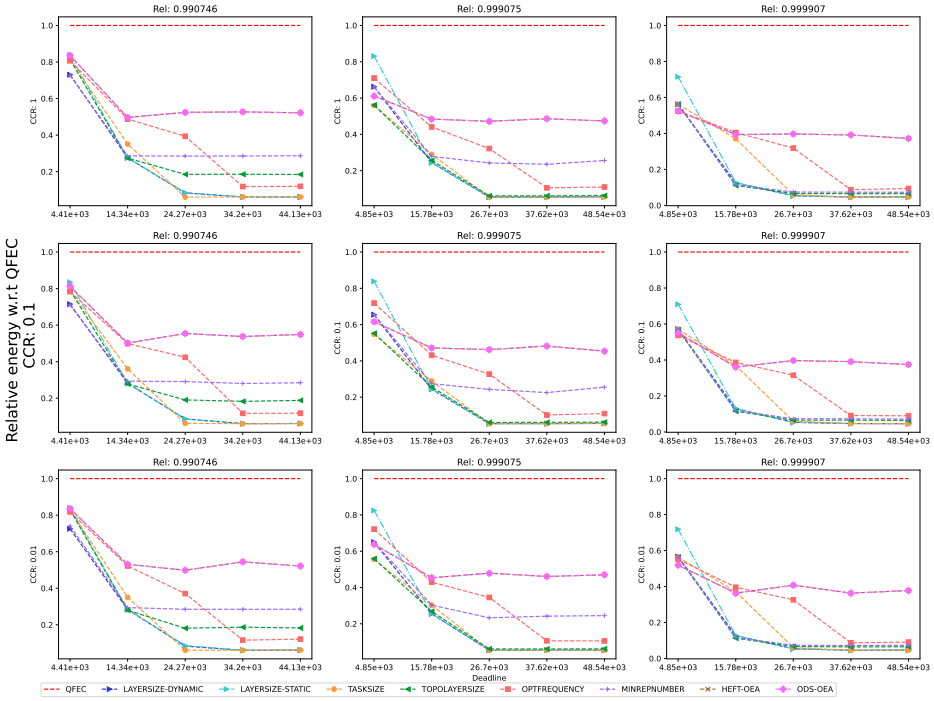

Figure 170: Performance of the different heuristics on the Epigenomics workflow.

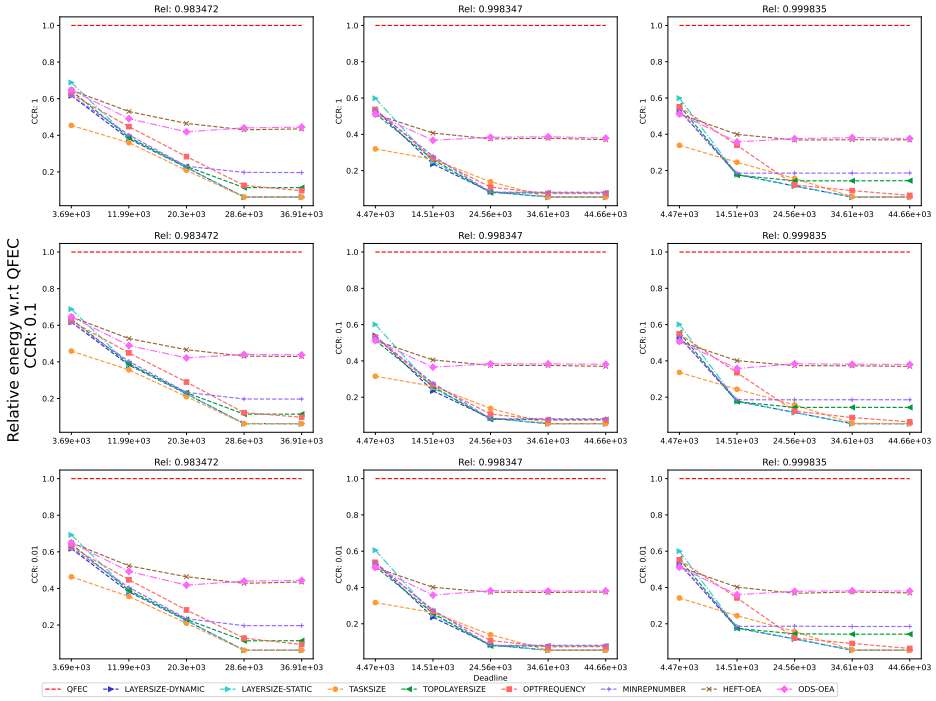

Figure 171: Performance of the different heuristics on the Genome workflow.

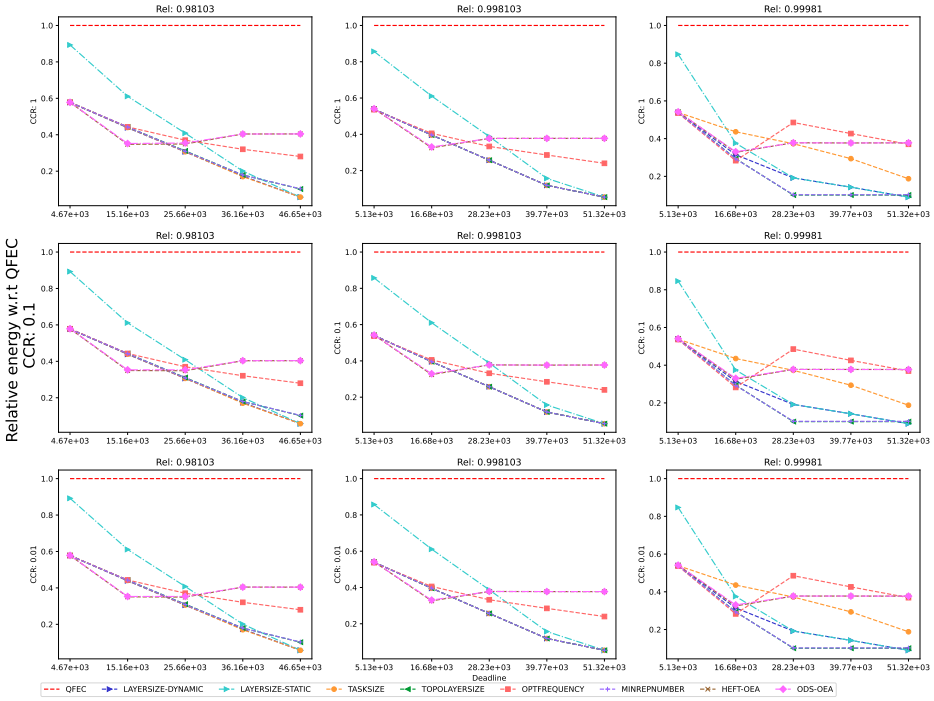

Figure 172: Performance of the different heuristics on the LU workflow.

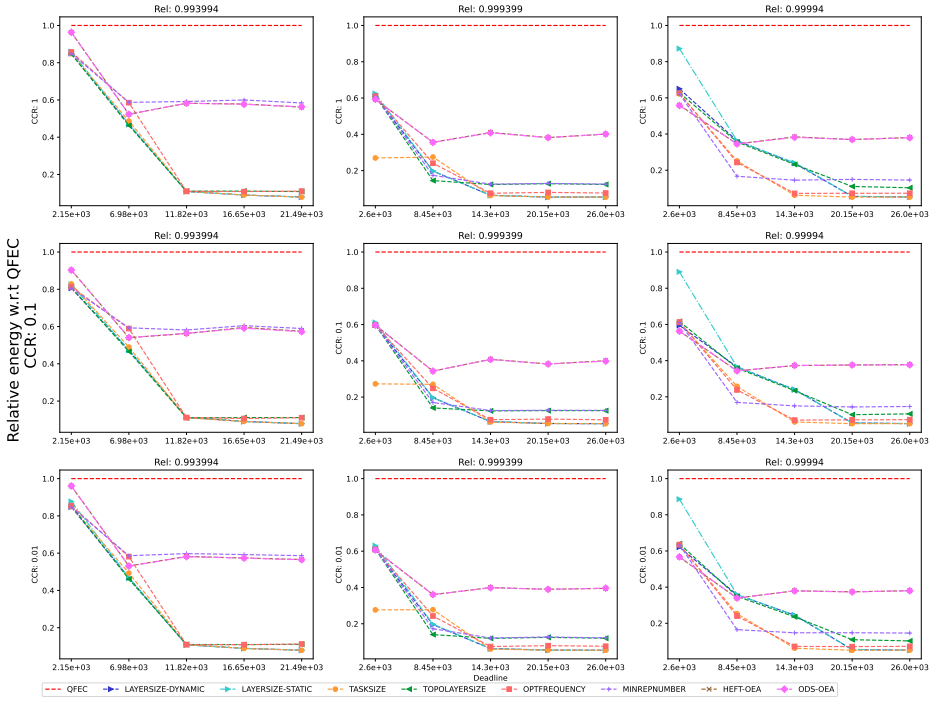

Figure 173: Performance of the different heuristics on the Montage workflow.

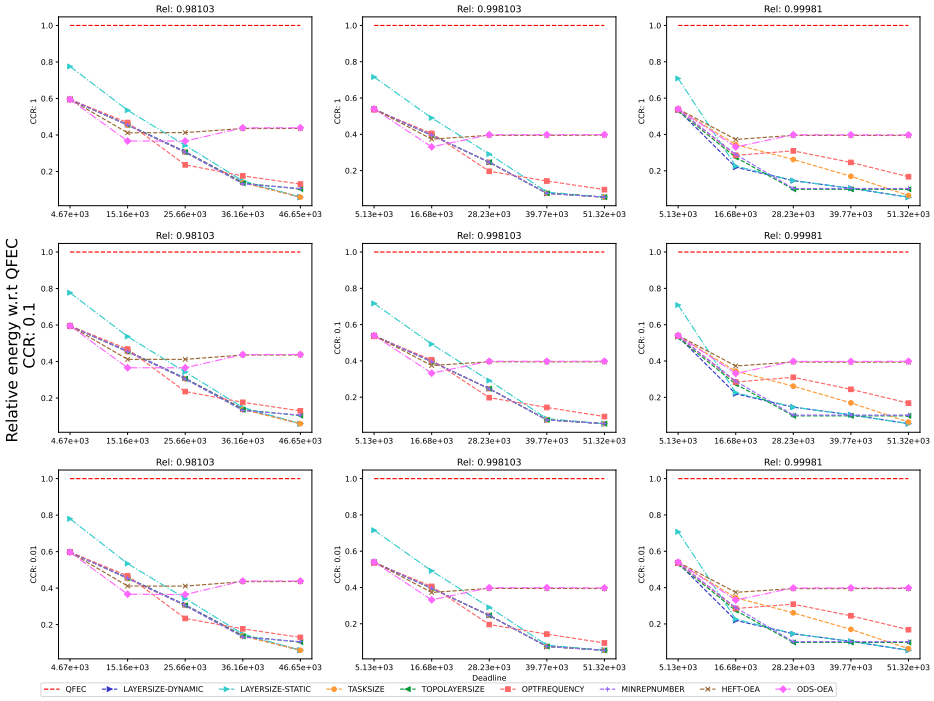

Figure 174: Performance of the different heuristics on the QR workflow.

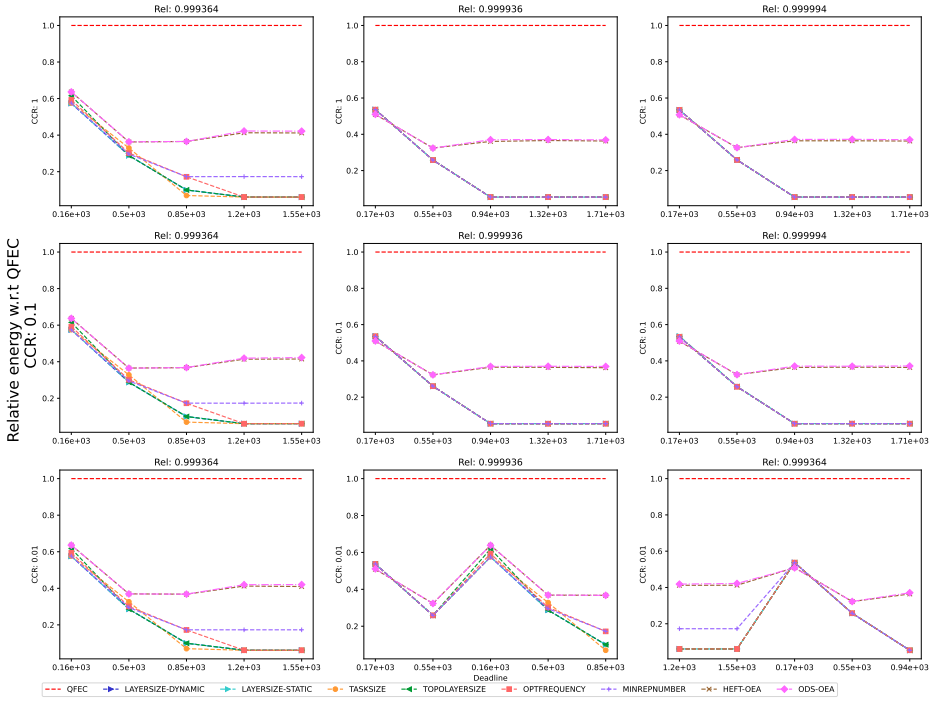

Figure 175: Performance of the different heuristics on the Seismology workflow.

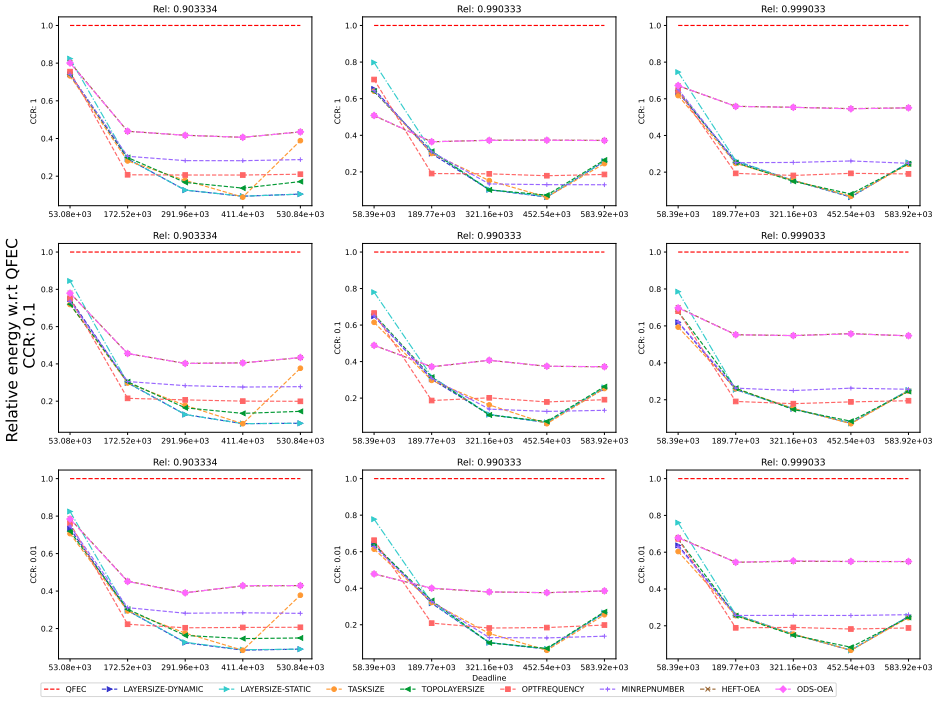

Figure 176: Performance of the different heuristics on the SoyKB workflow.

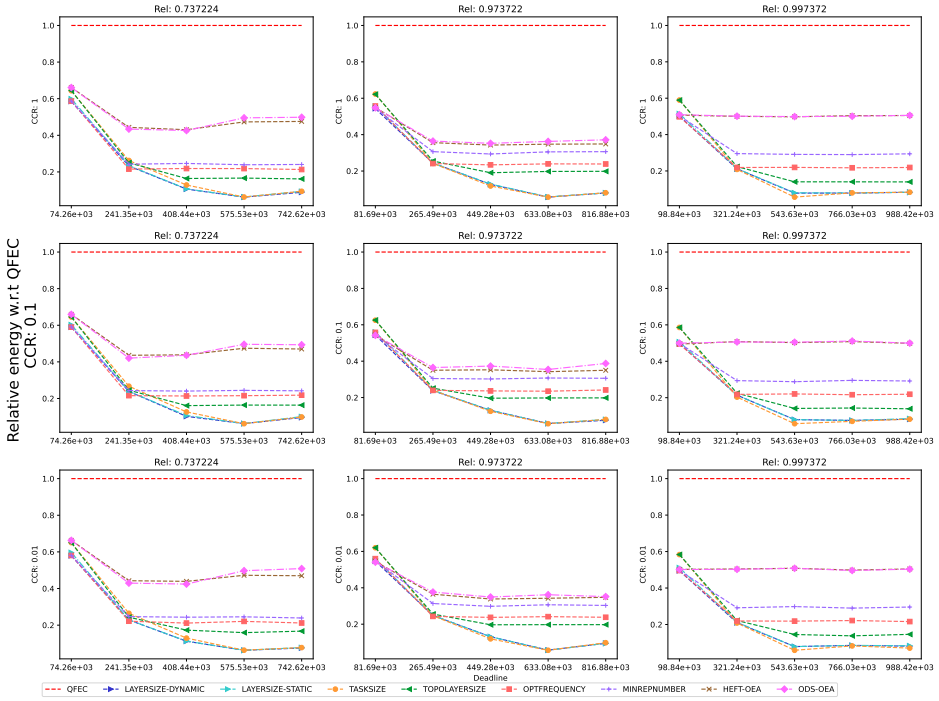

Figure 177: Performance of the different heuristics on the SRASearch.

D.4.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

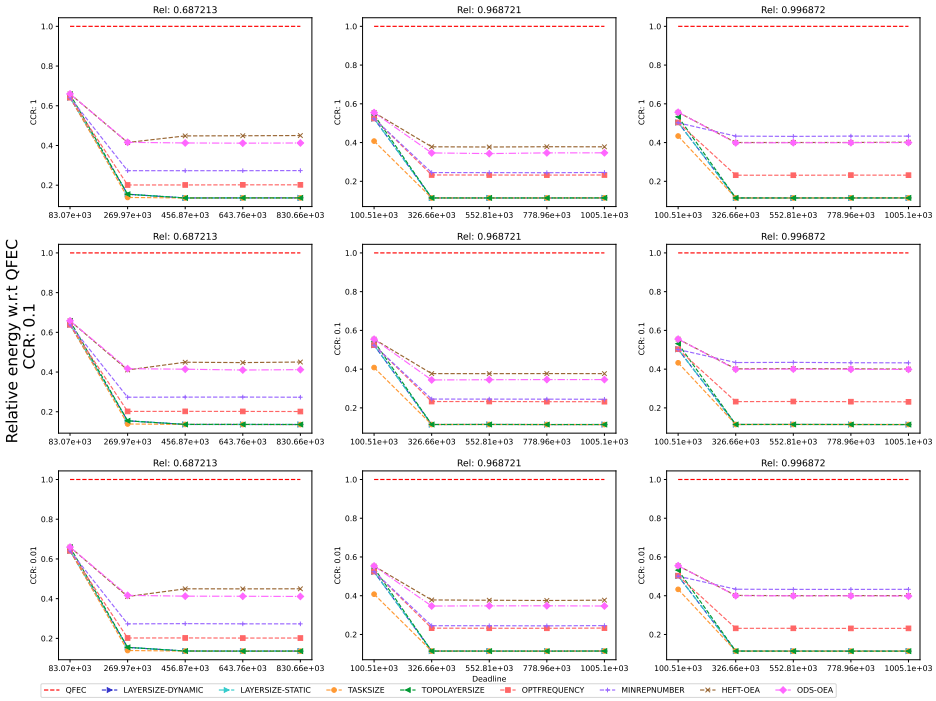

Figure 178: Performance of the different heuristics on the BLAST workflow.

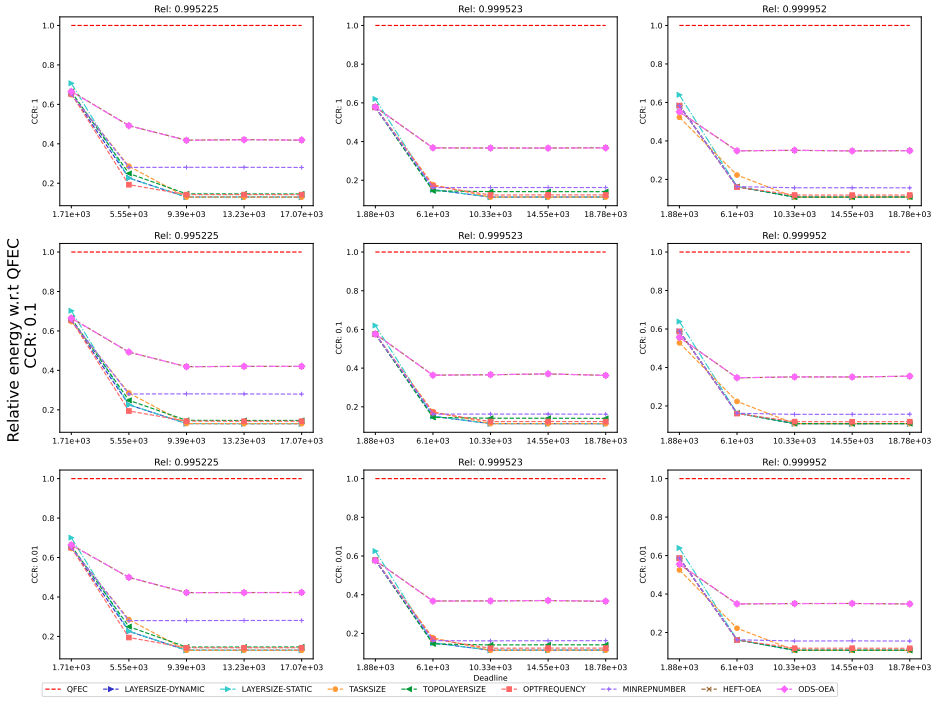

Figure 179: Performance of the different heuristics on the BWA workflow.

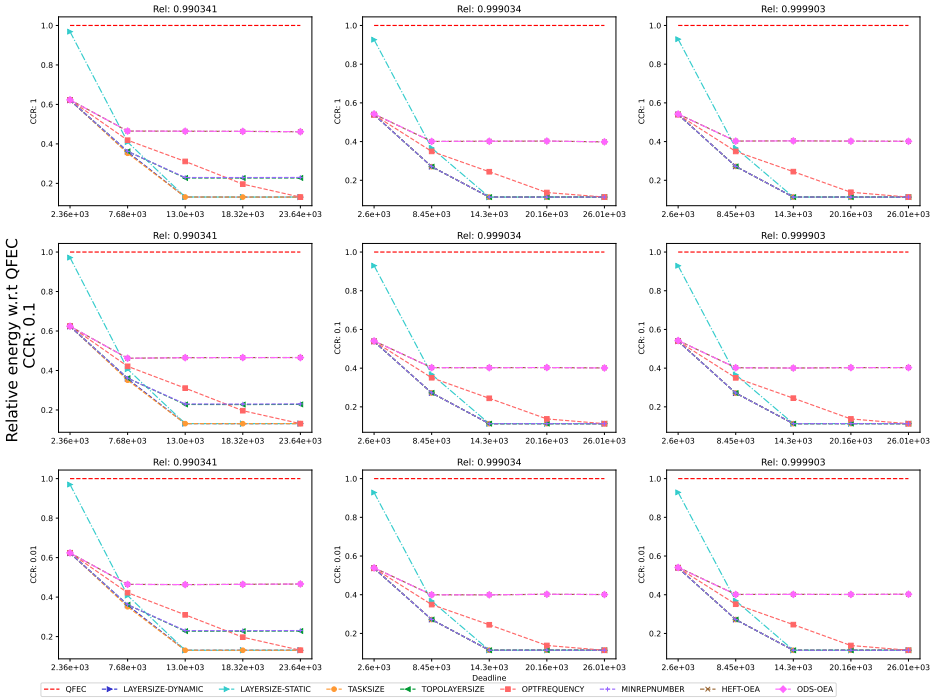

Figure 180: Performance of the different heuristics on the Cholesky workflow.

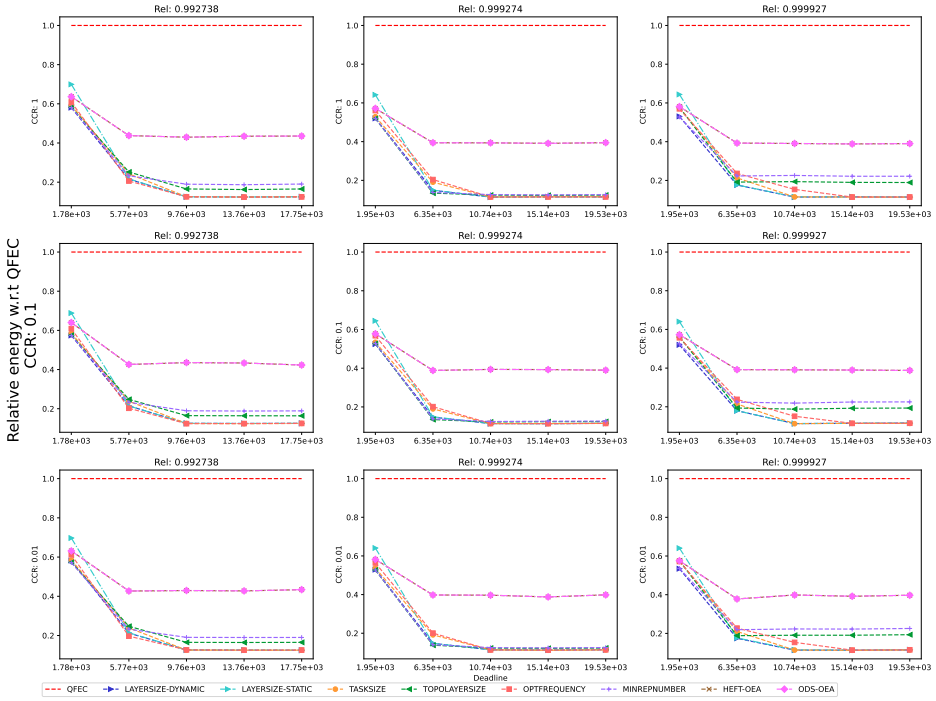

Figure 181: Performance of the different heuristics on the Cycles workflow.

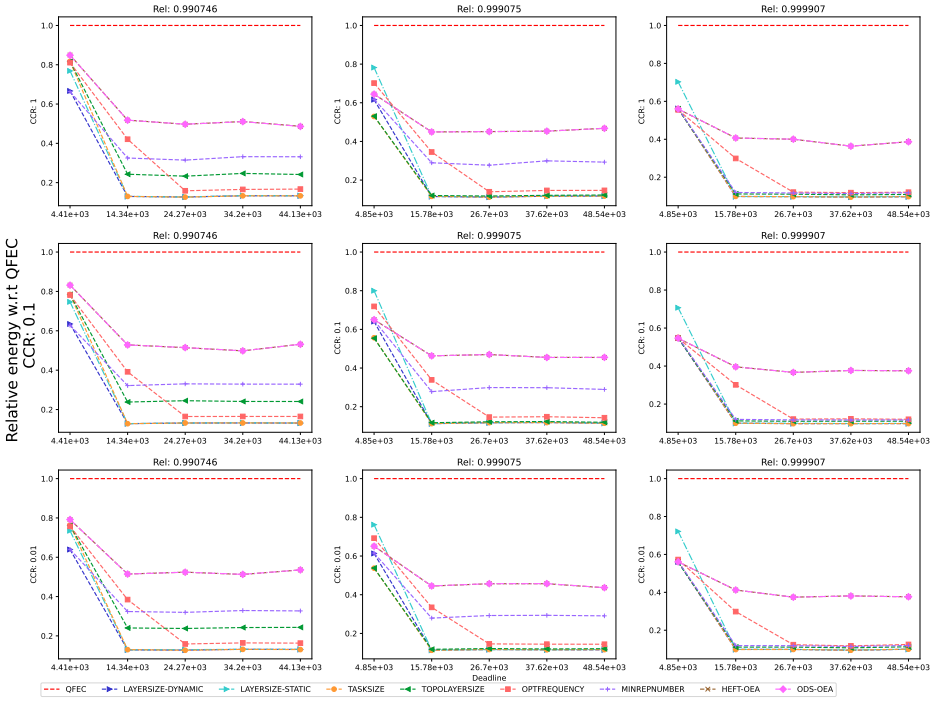

Figure 182: Performance of the different heuristics on the Epigenomics workflow.

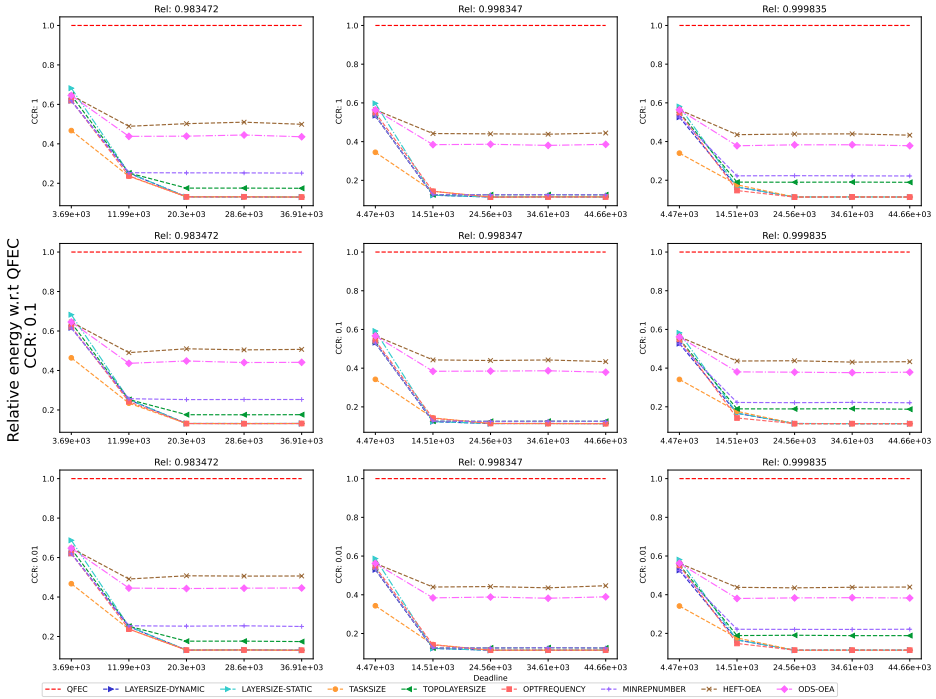

Figure 183: Performance of the different heuristics on the Genome workflow.

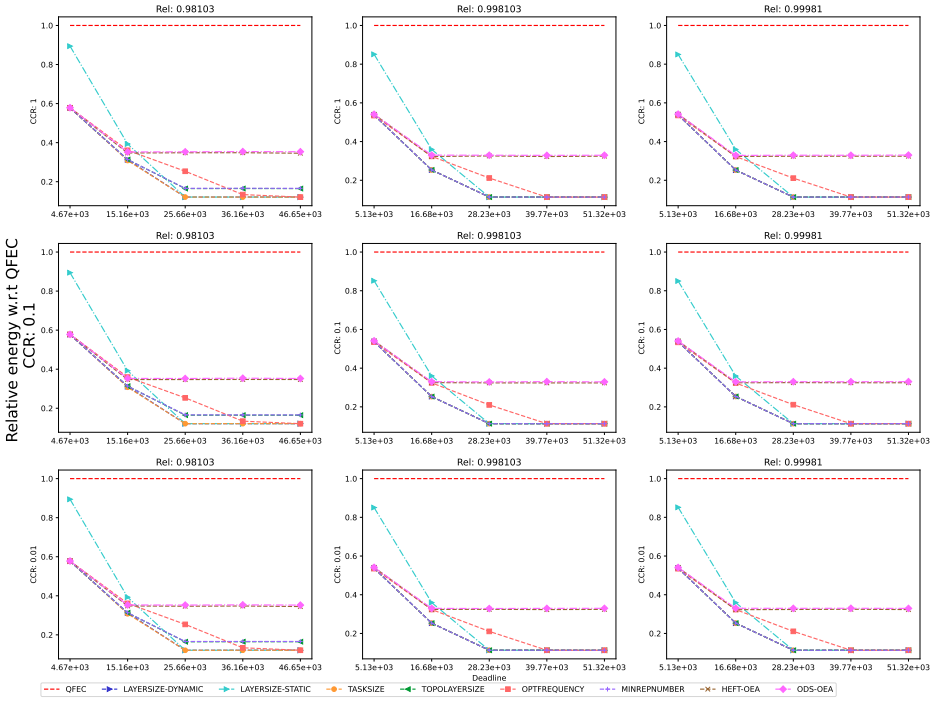

Figure 184: Performance of the different heuristics on the LU workflow.

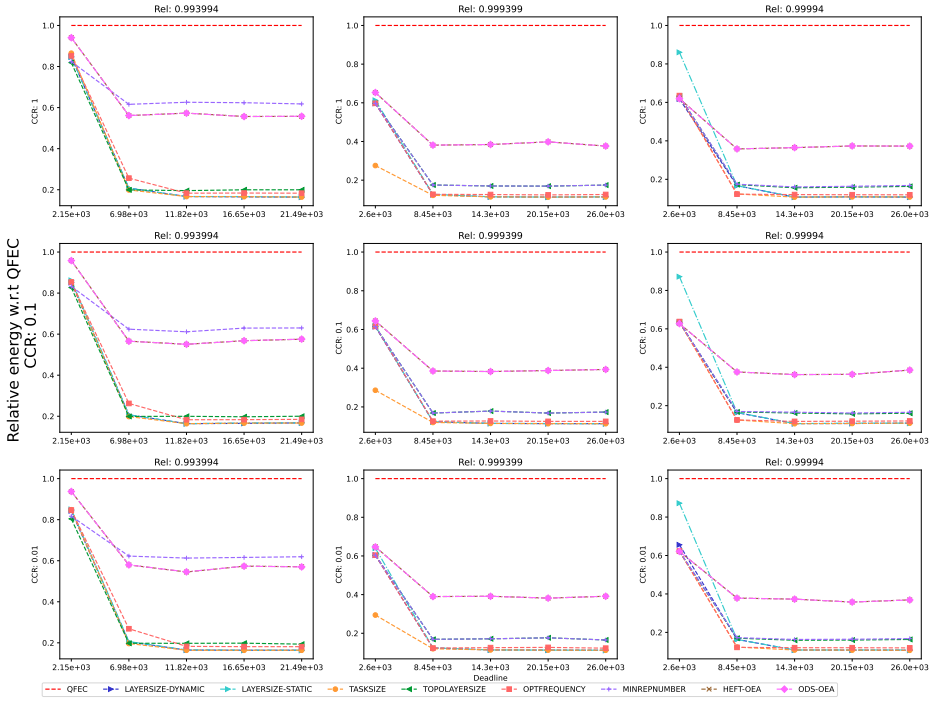

Figure 185: Performance of the different heuristics on the Montage workflow.

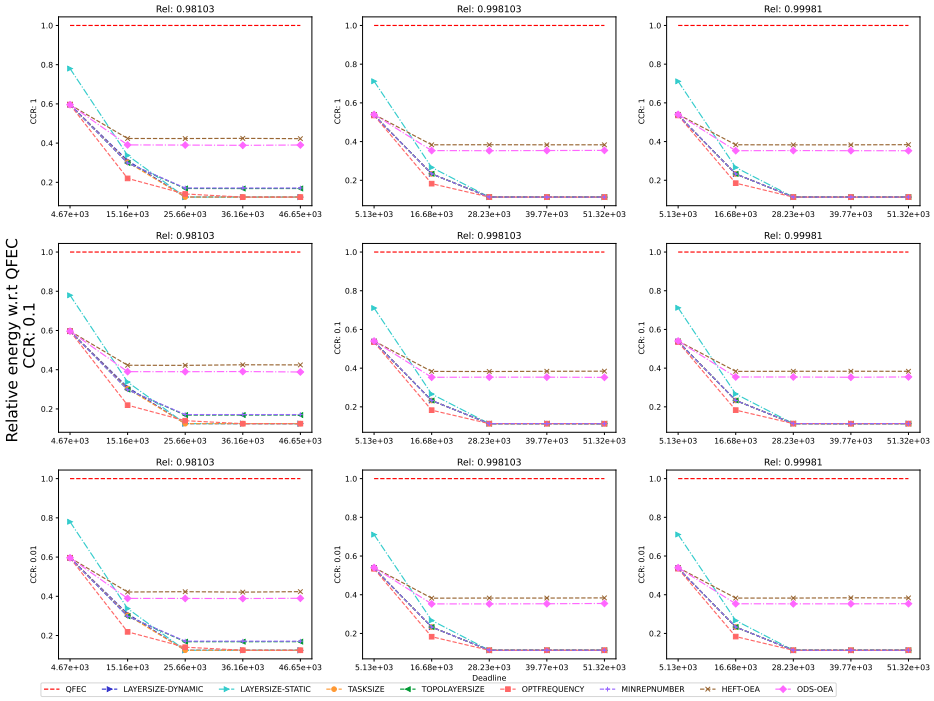

Figure 186: Performance of the different heuristics on the QR workflow.

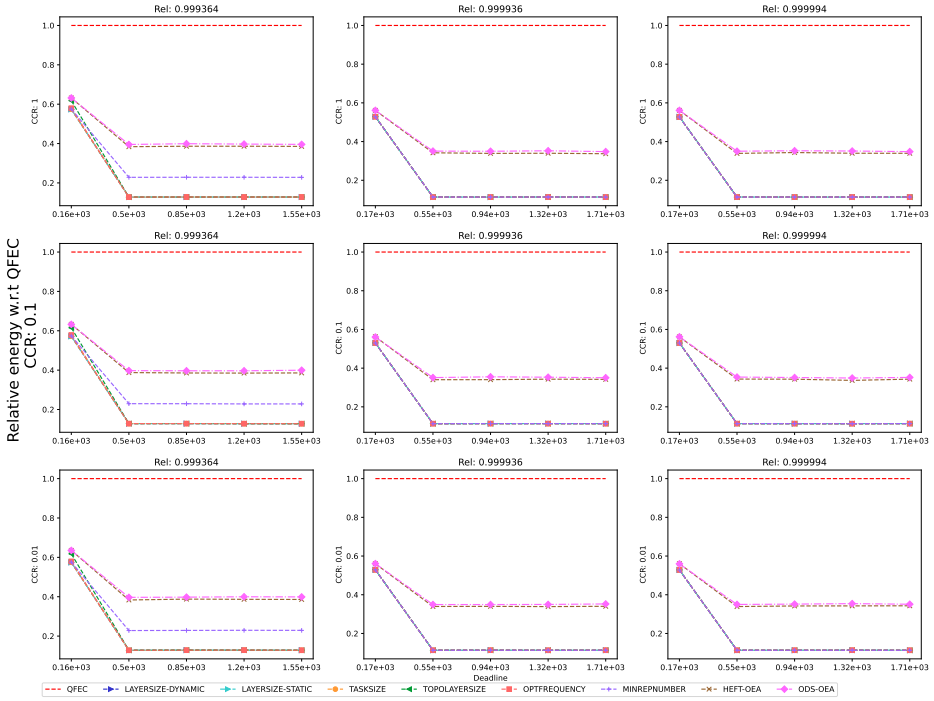

Figure 187: Performance of the different heuristics on the Seismology workflow.

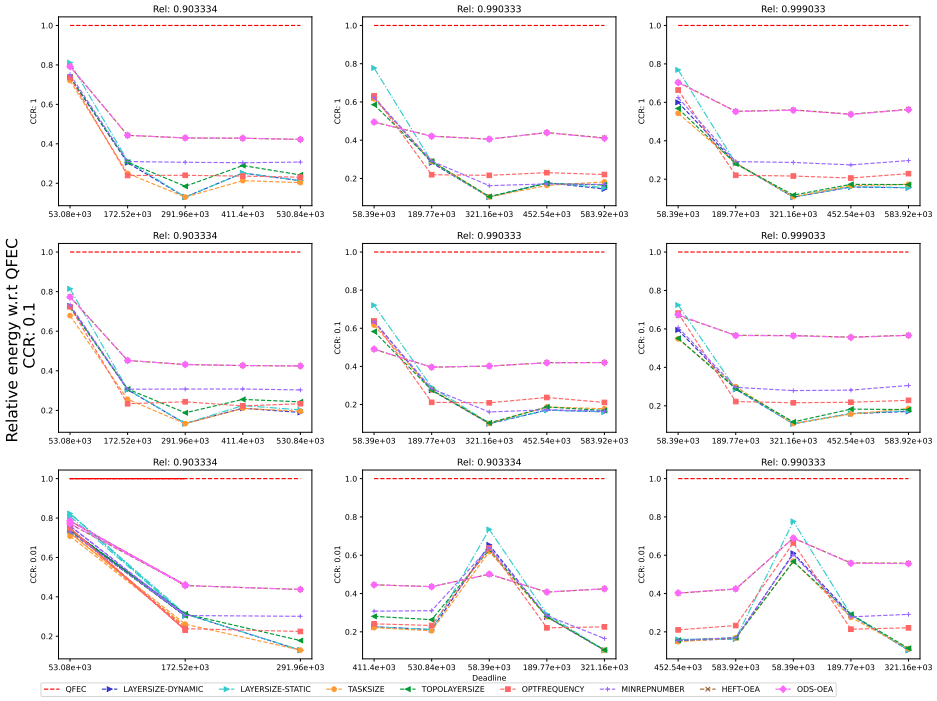

Figure 188: Performance of the different heuristics on the SoyKB workflow.

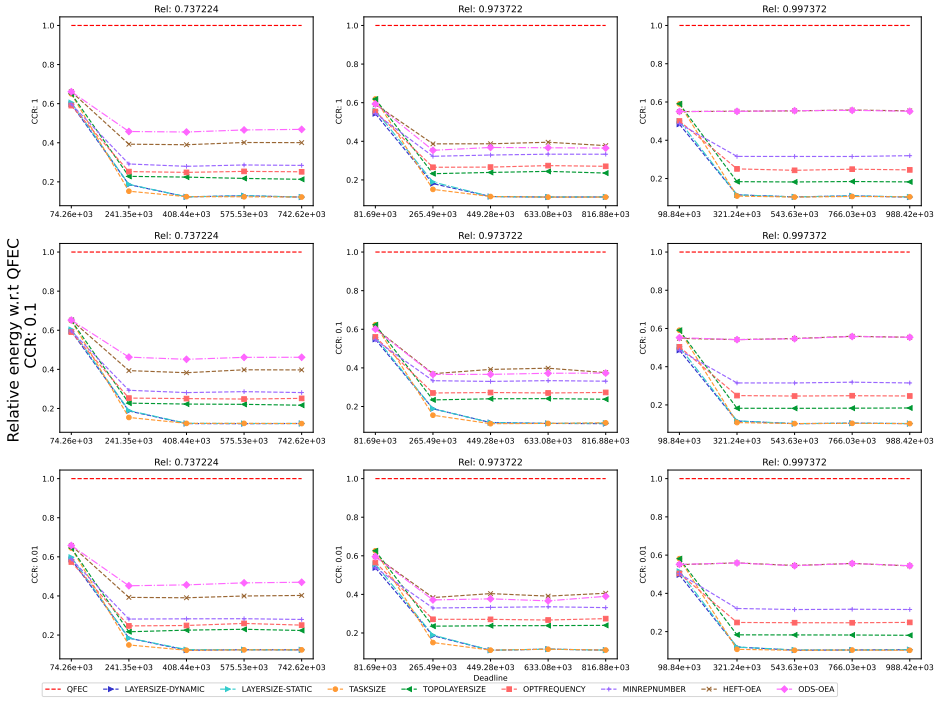

Figure 189: Performance of the different heuristics on the SRASearch.

D.4.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

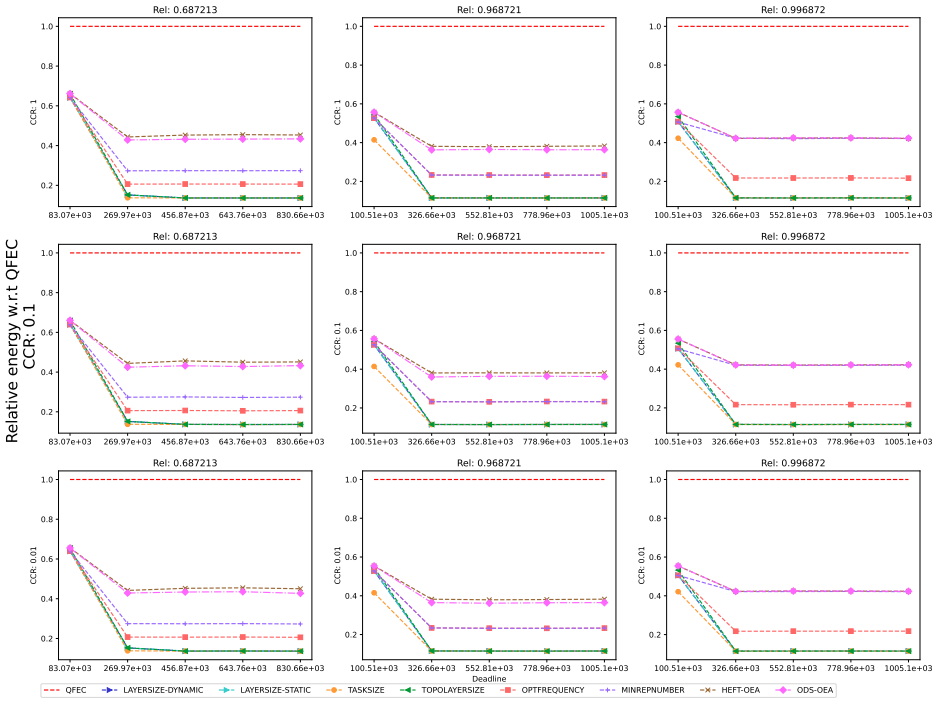

Figure 190: Performance of the different heuristics on the BLAST workflow.

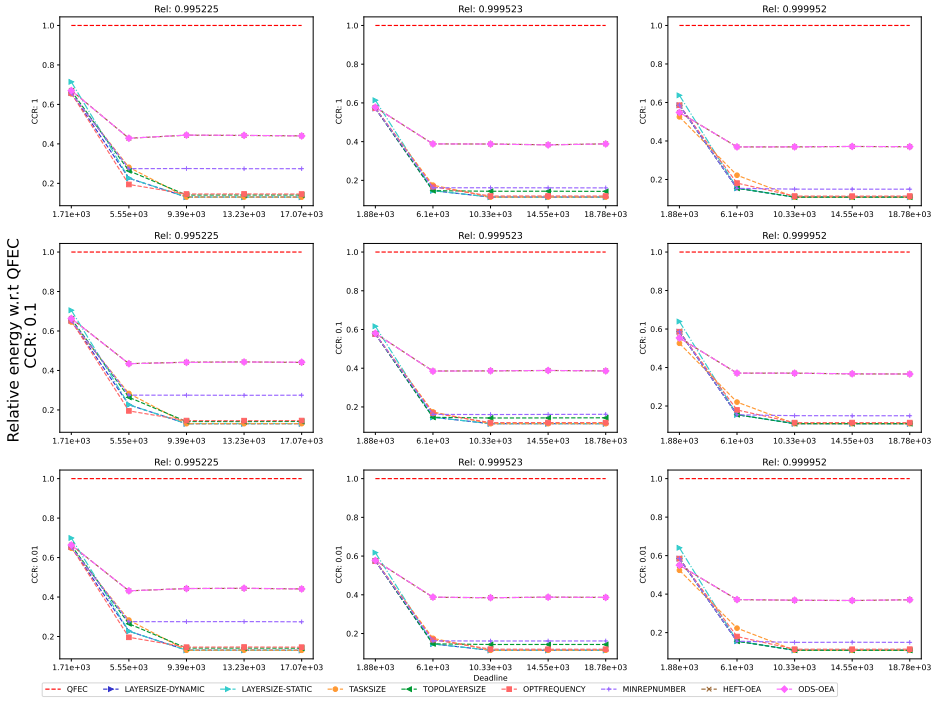

Figure 191: Performance of the different heuristics on the BWA workflow.

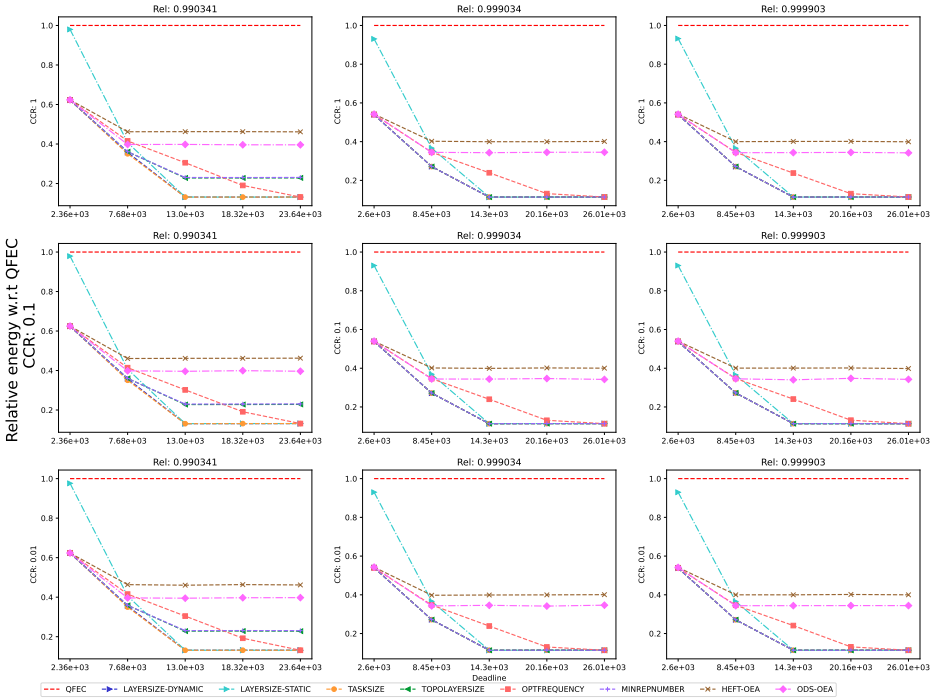

Figure 192: Performance of the different heuristics on the Cholesky workflow.

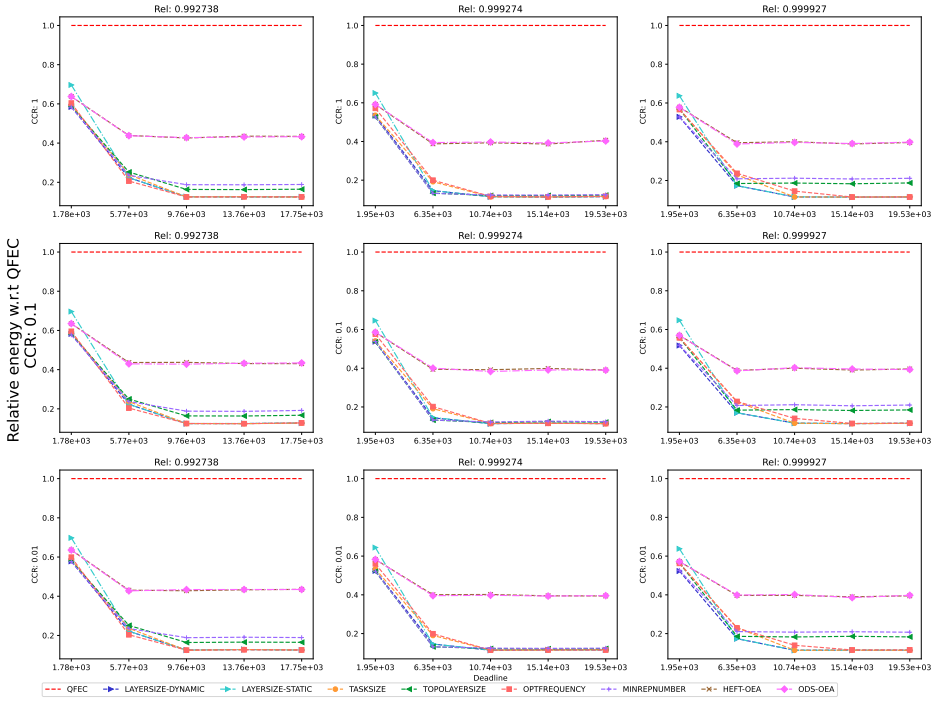

Figure 193: Performance of the different heuristics on the Cycles workflow.

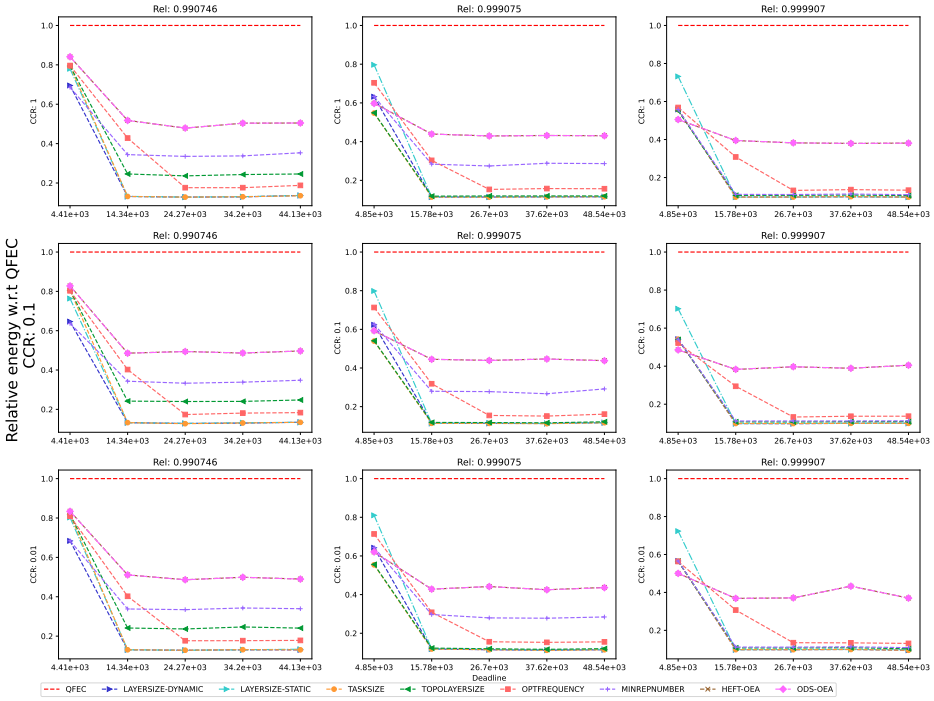

Figure 194: Performance of the different heuristics on the Epigenomics workflow.

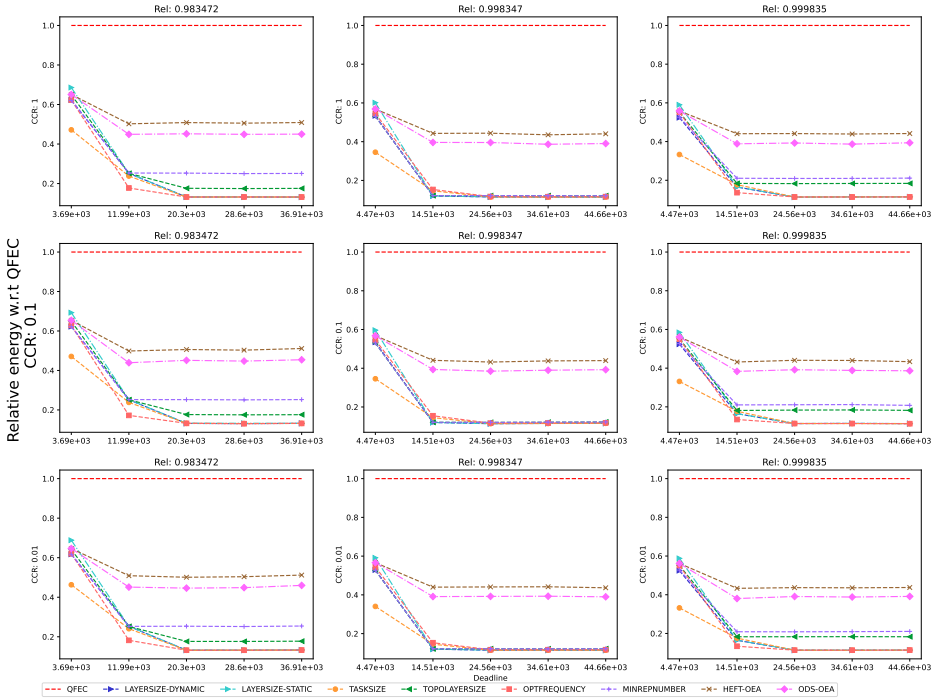

Figure 195: Performance of the different heuristics on the Genome workflow.

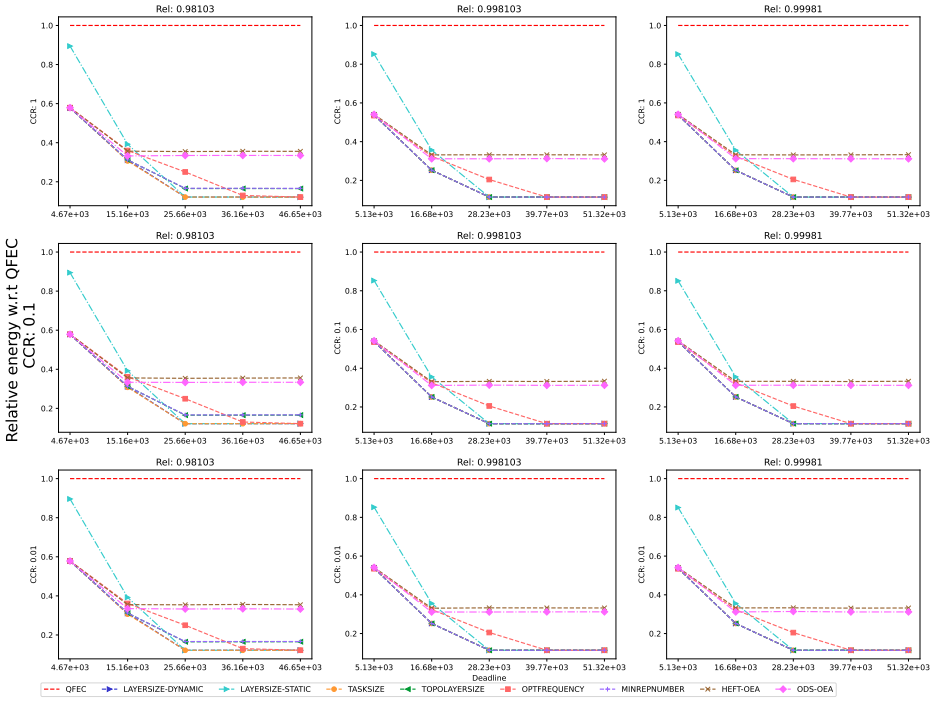

Figure 196: Performance of the different heuristics on the LU workflow.

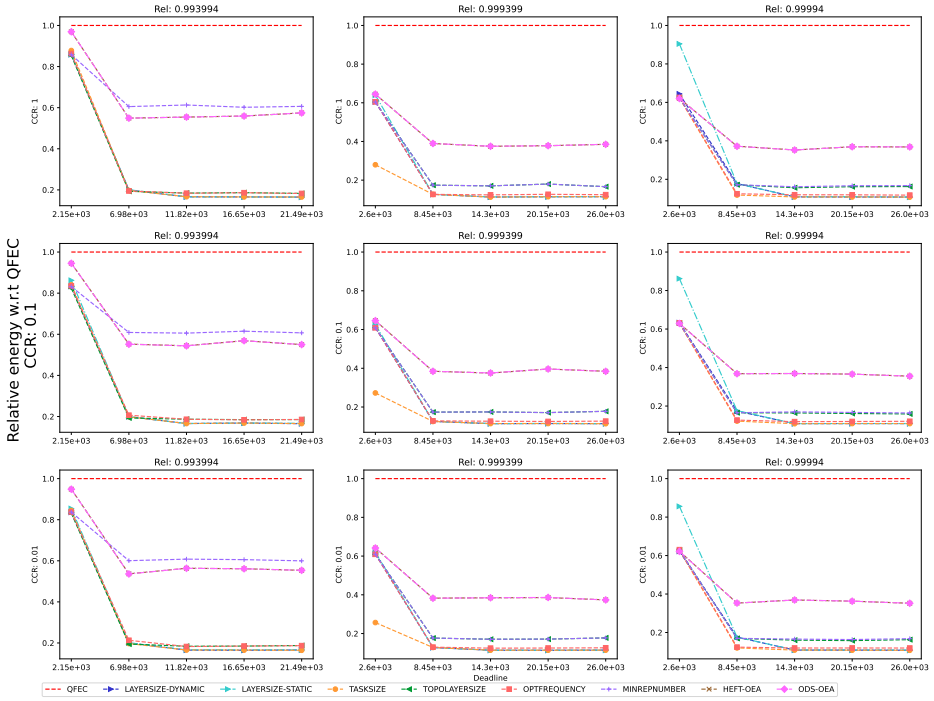

Figure 197: Performance of the different heuristics on the Montage workflow.

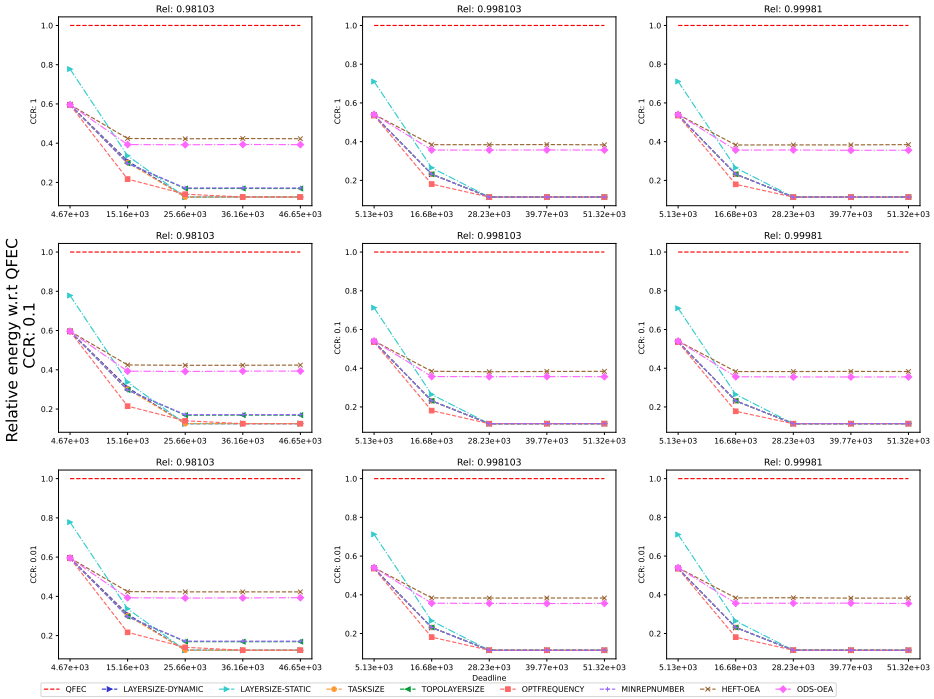

Figure 198: Performance of the different heuristics on the QR workflow.

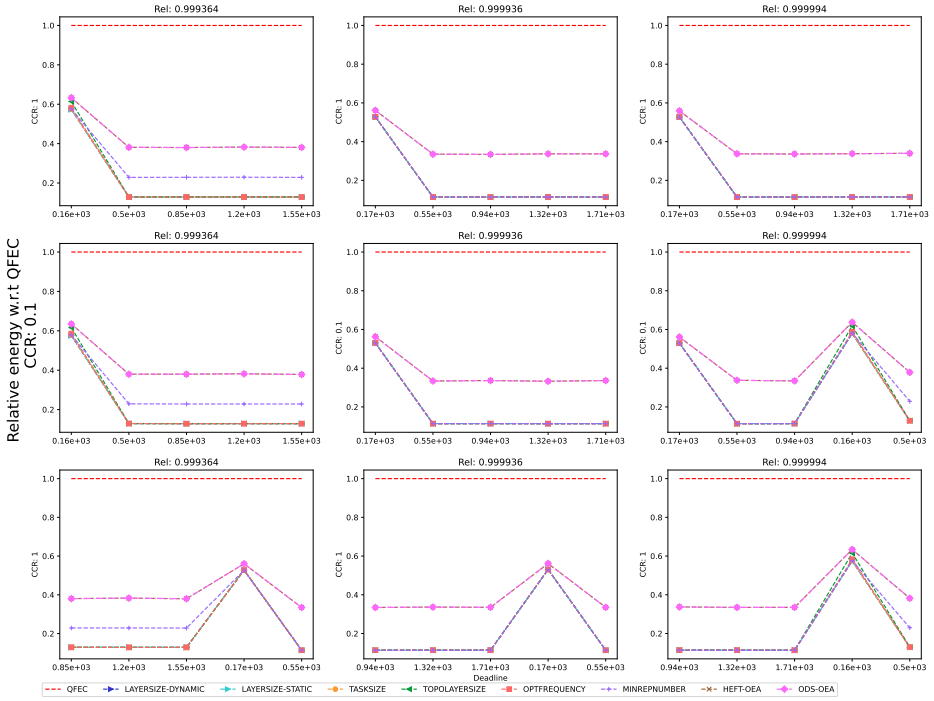

Figure 199: Performance of the different heuristics on the Seismology workflow.

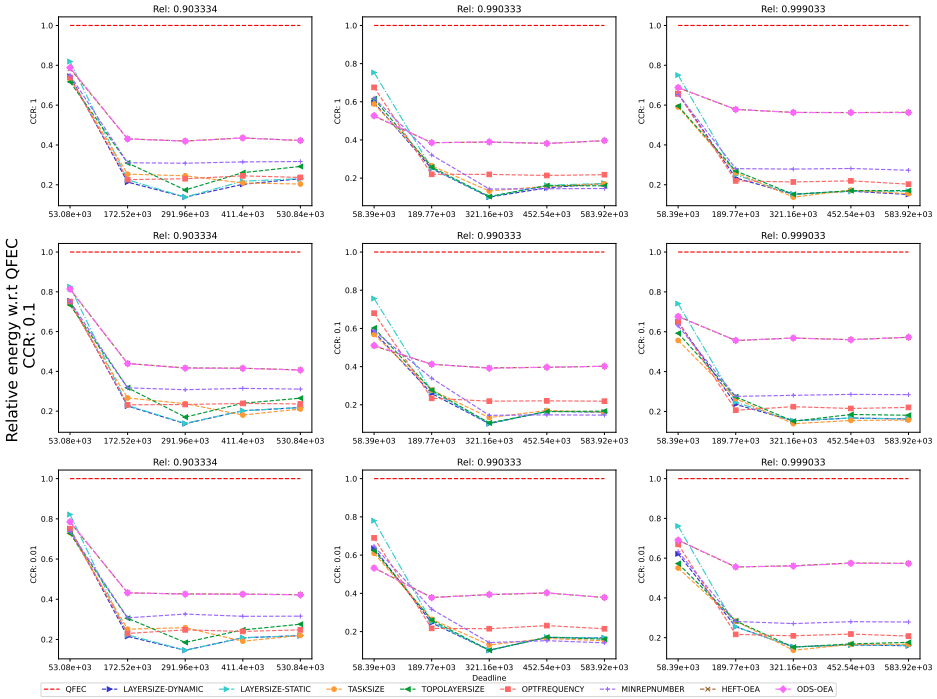

Figure 200: Performance of the different heuristics on the SoyKB workflow.

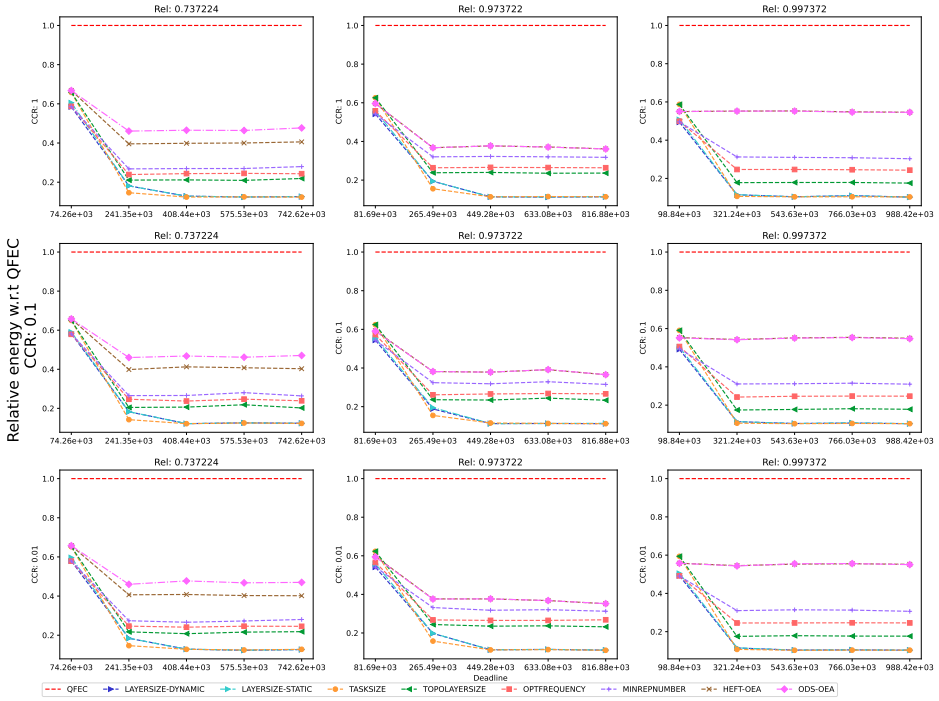

Figure 201: Performance of the different heuristics on the SRASearch.

D.5 $BC/WC = 0.5$

D.5.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

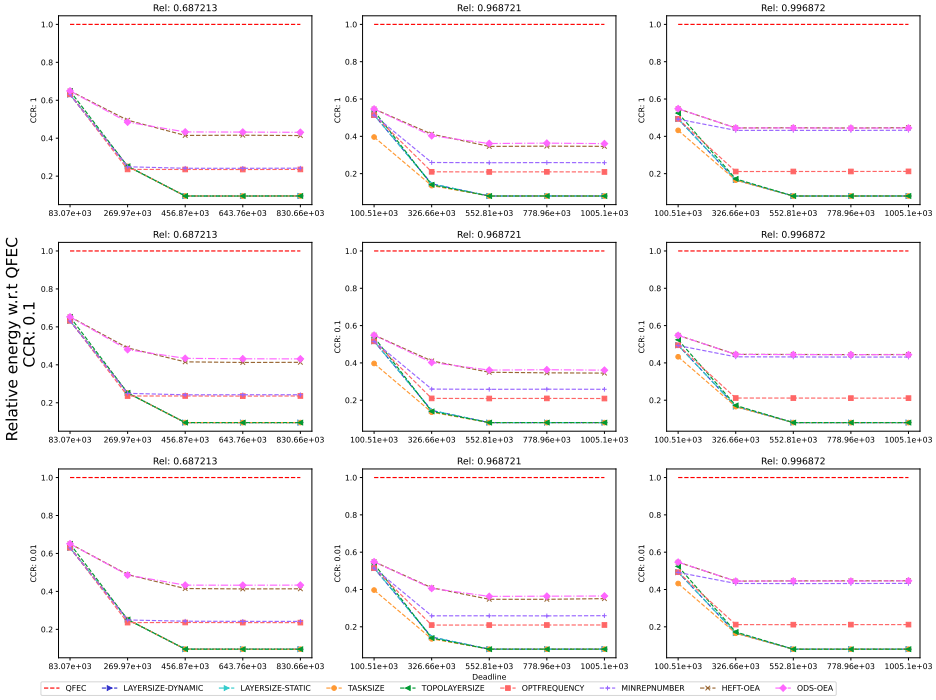

Figure 202: Performance of the different heuristics on the BLAST workflow.

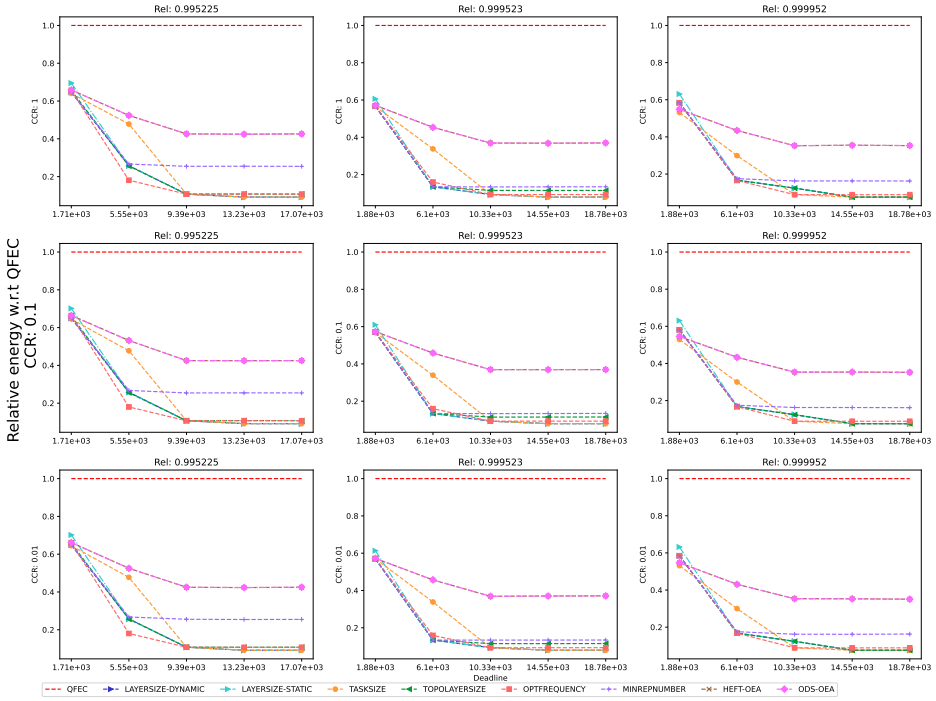

Figure 203: Performance of the different heuristics on the BWA workflow.

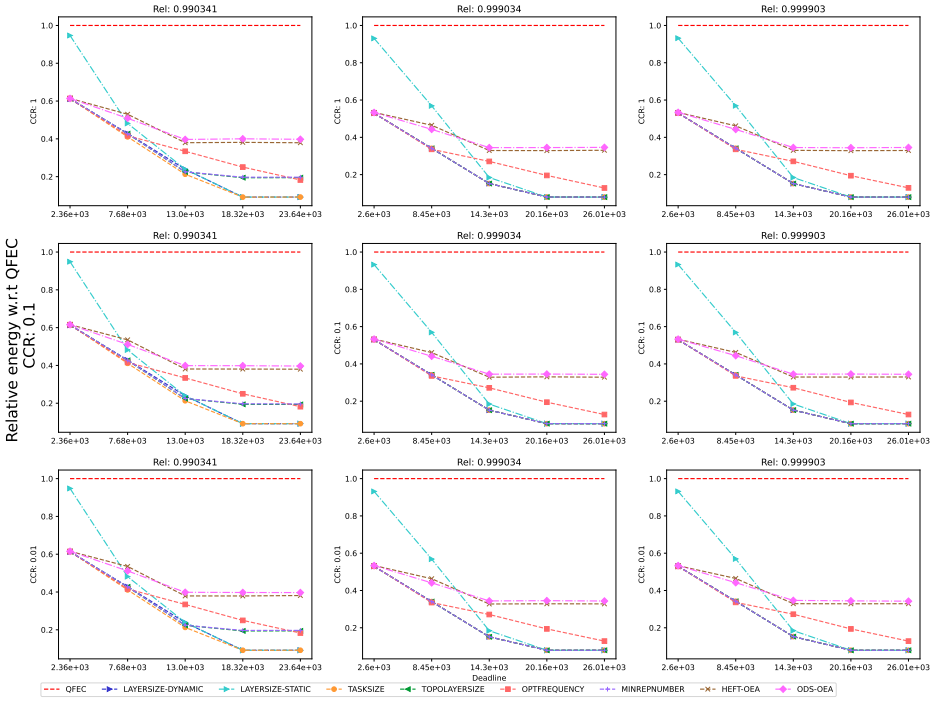

Figure 204: Performance of the different heuristics on the Cholesky workflow.

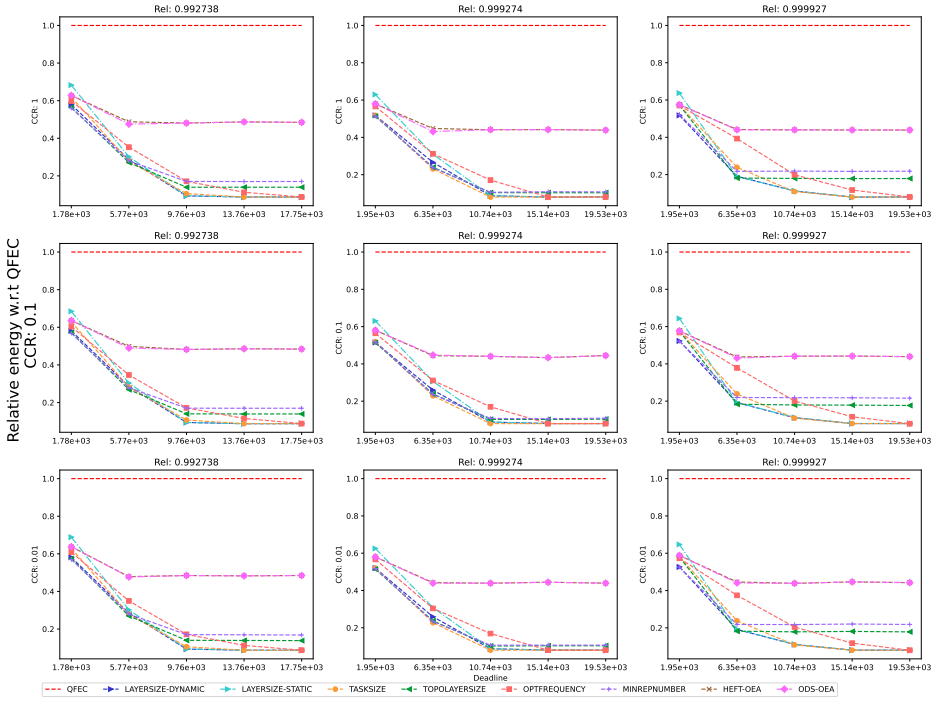

Figure 205: Performance of the different heuristics on the Cycles workflow.

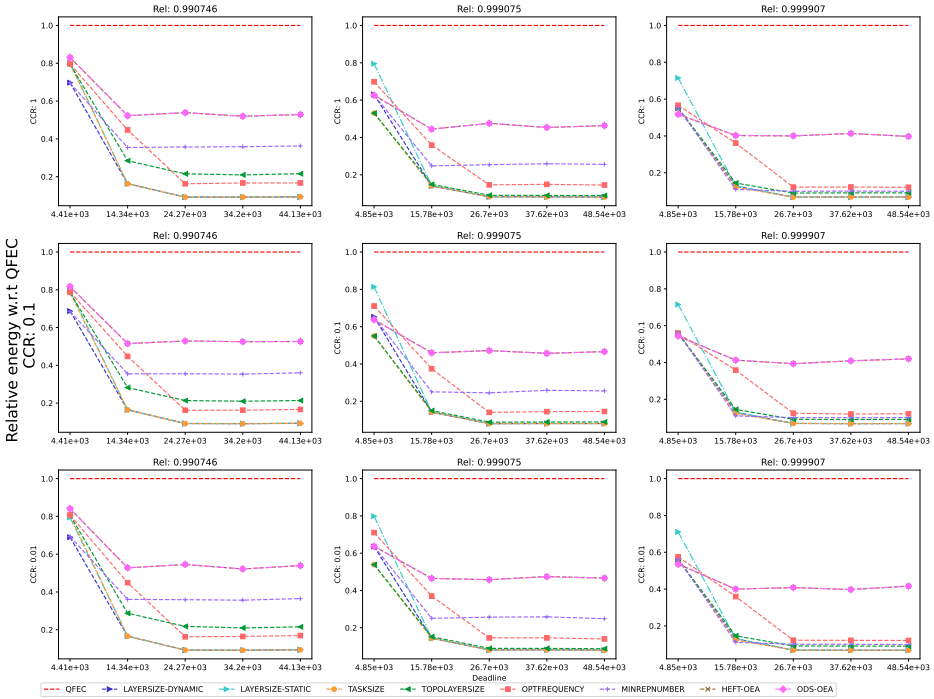

Figure 206: Performance of the different heuristics on the Epigenomics workflow.

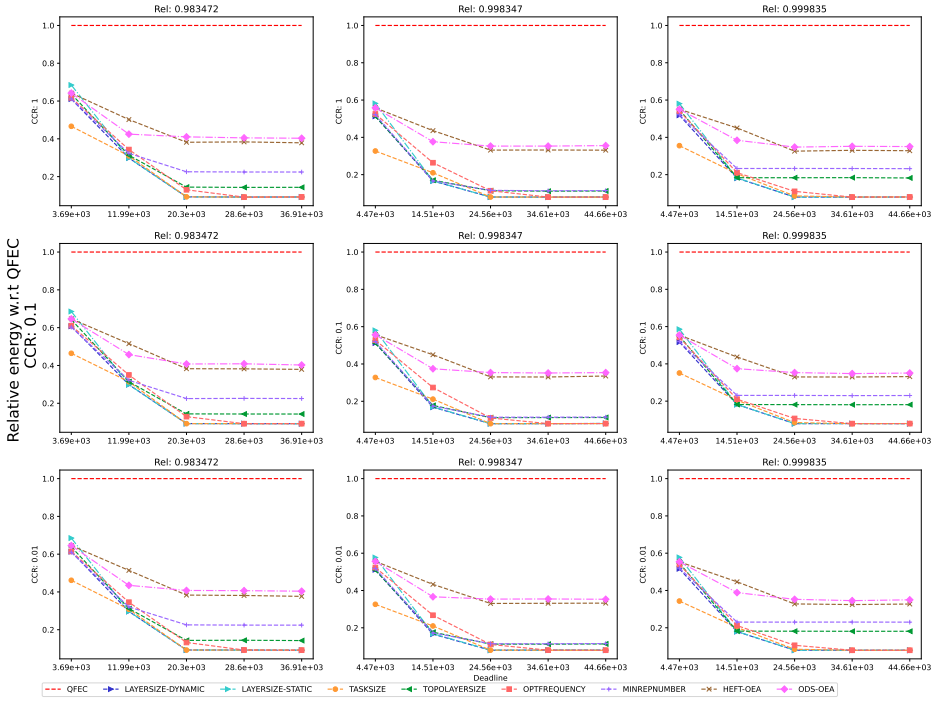

Figure 207: Performance of the different heuristics on the Genome workflow.

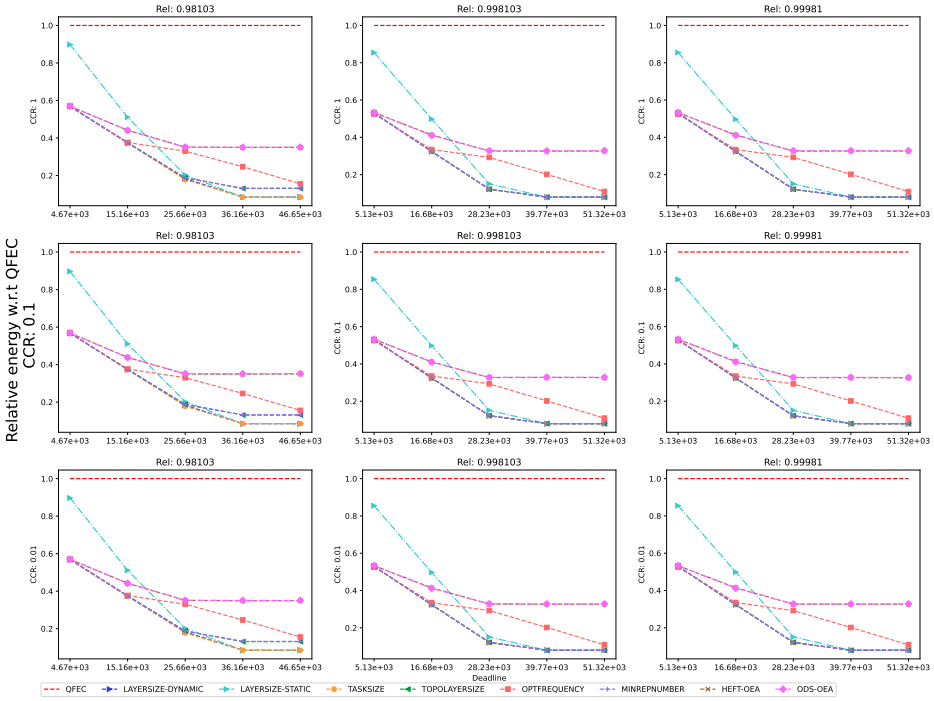

Figure 208: Performance of the different heuristics on the LU workflow.

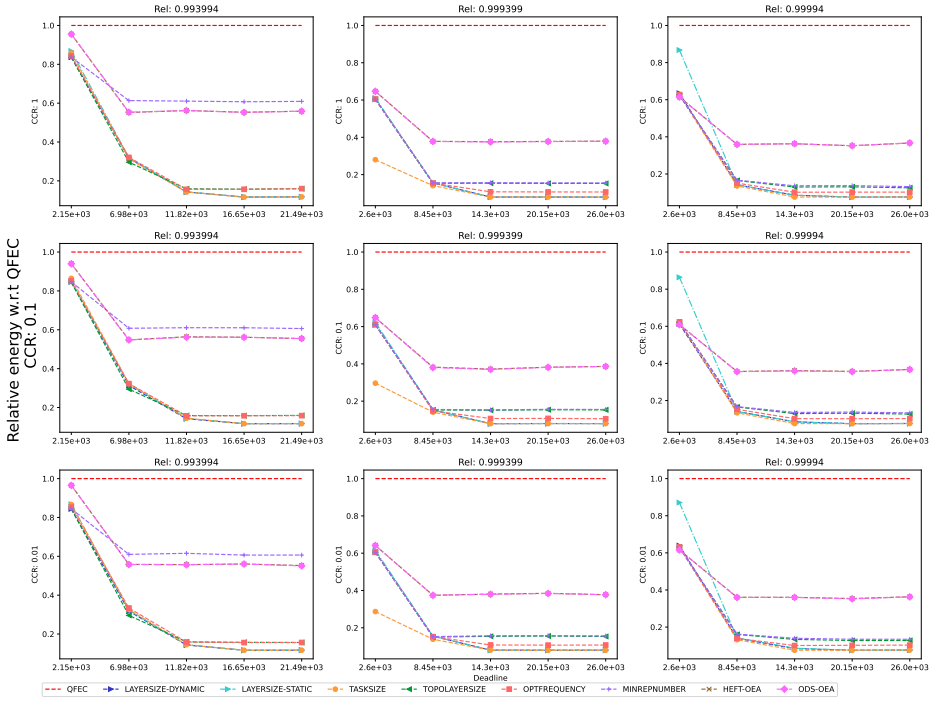

Figure 209: Performance of the different heuristics on the Montage workflow.

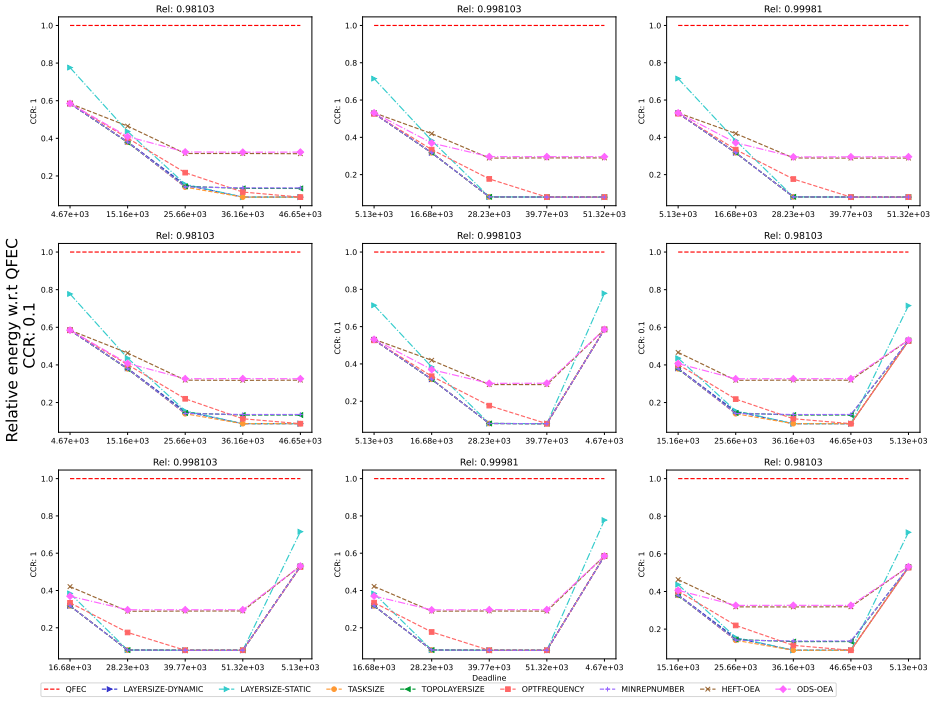

Figure 210: Performance of the different heuristics on the QR workflow.

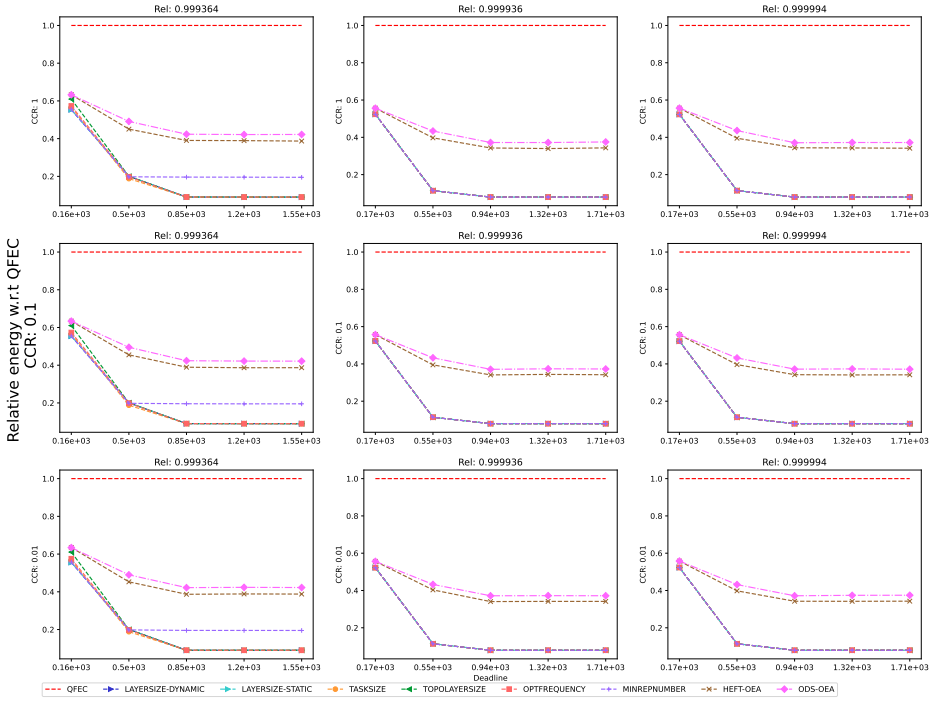

Figure 211: Performance of the different heuristics on the Seismology workflow.

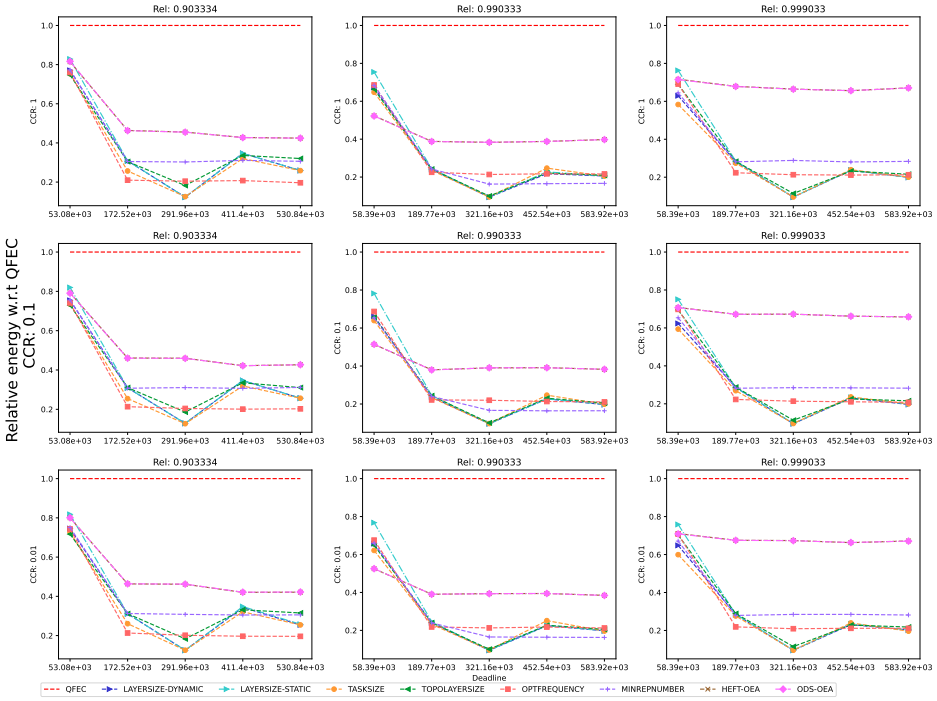

Figure 212: Performance of the different heuristics on the SoyKB workflow.

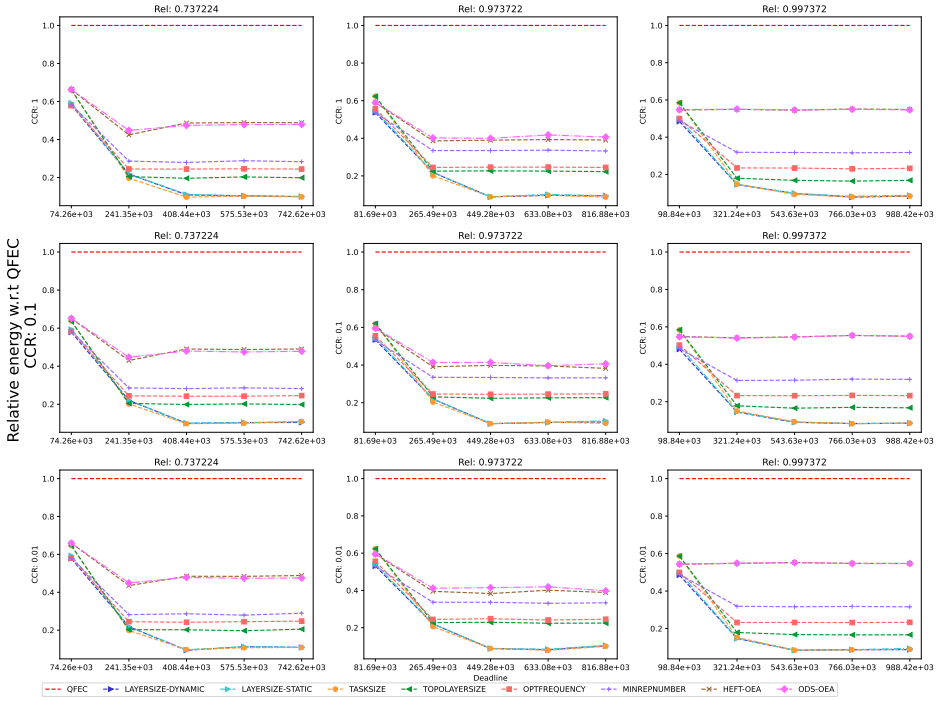

Figure 213: Performance of the different heuristics on the SRASearch.

D.5.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

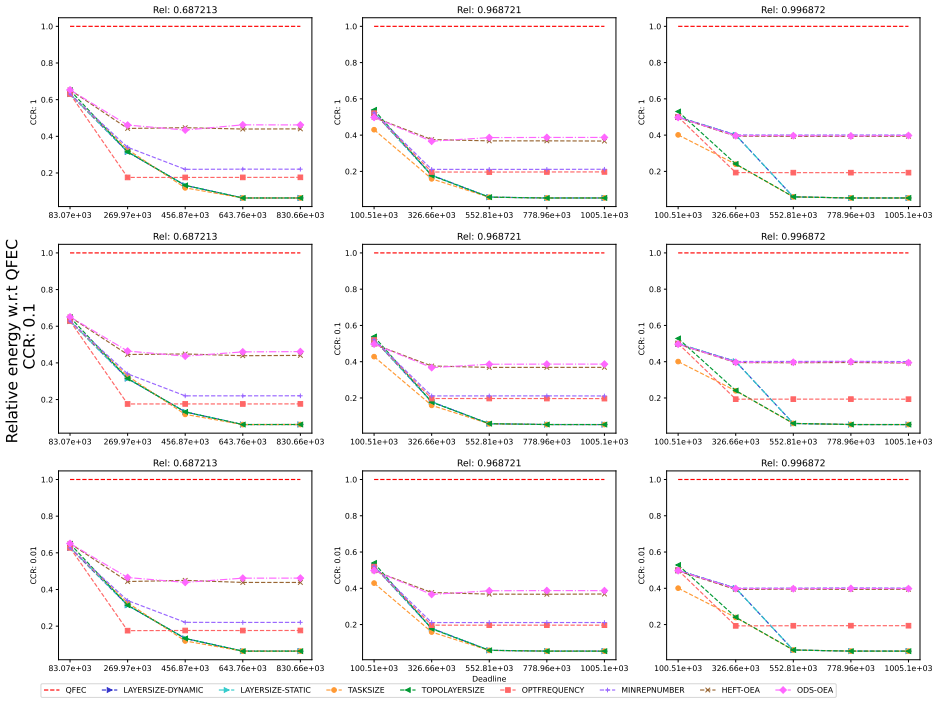

Figure 214: Performance of the different heuristics on the BLAST workflow.

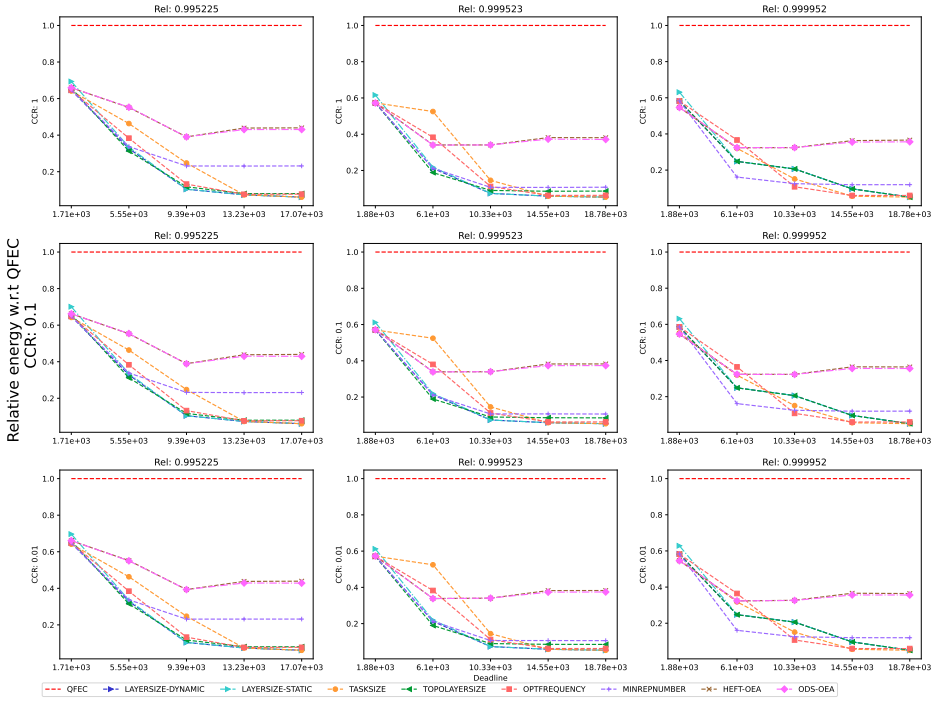

Figure 215: Performance of the different heuristics on the BWA workflow.

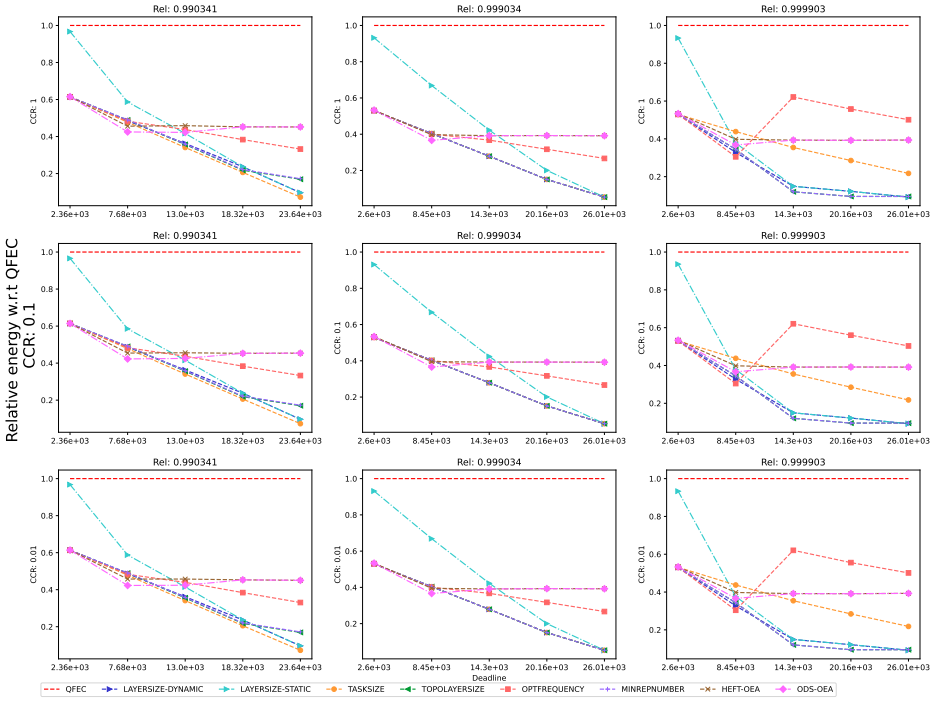

Figure 216: Performance of the different heuristics on the Cholesky workflow.

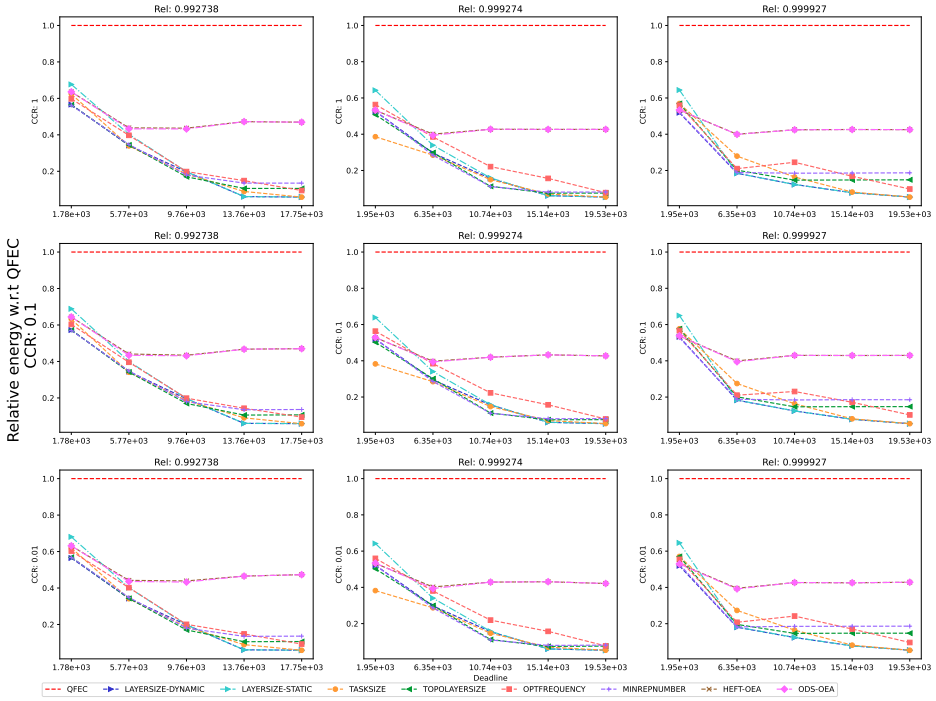

Figure 217: Performance of the different heuristics on the Cycles workflow.

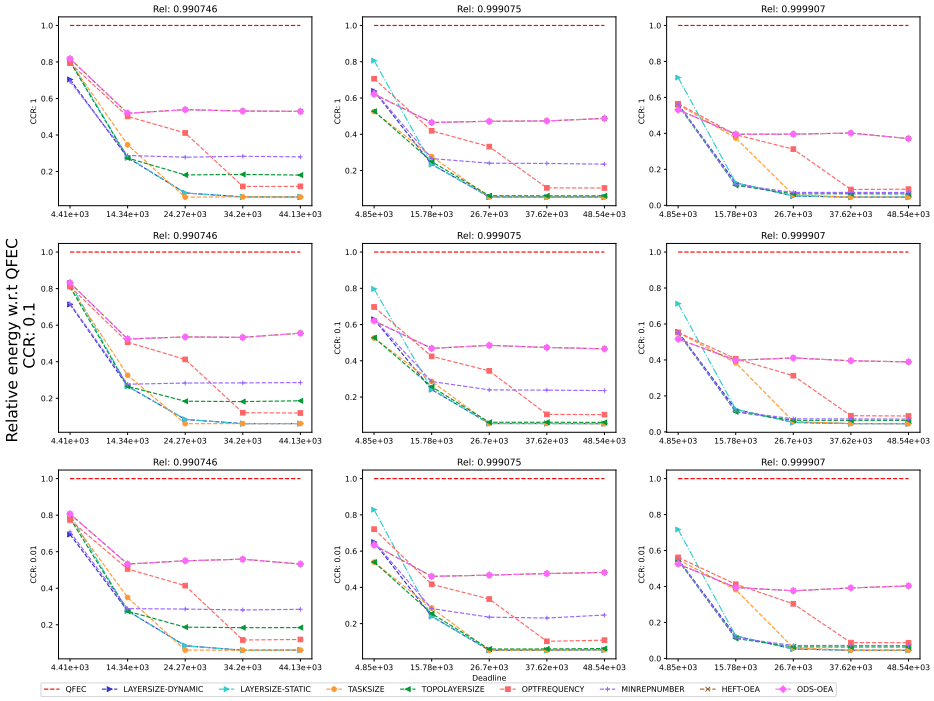

Figure 218: Performance of the different heuristics on the Epigenomics workflow.

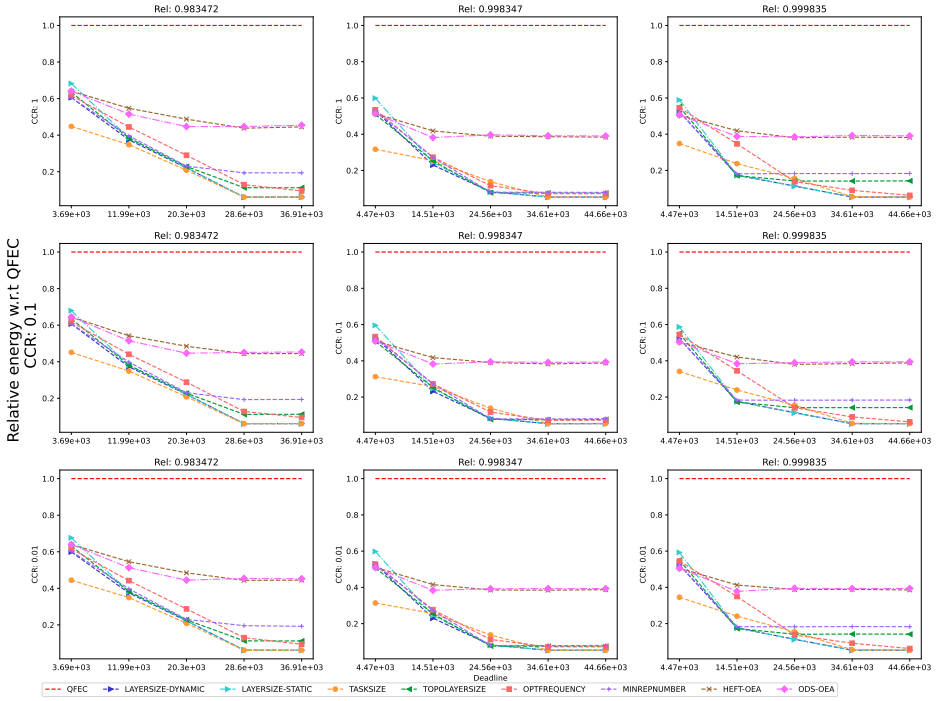

Figure 219: Performance of the different heuristics on the Genome workflow.

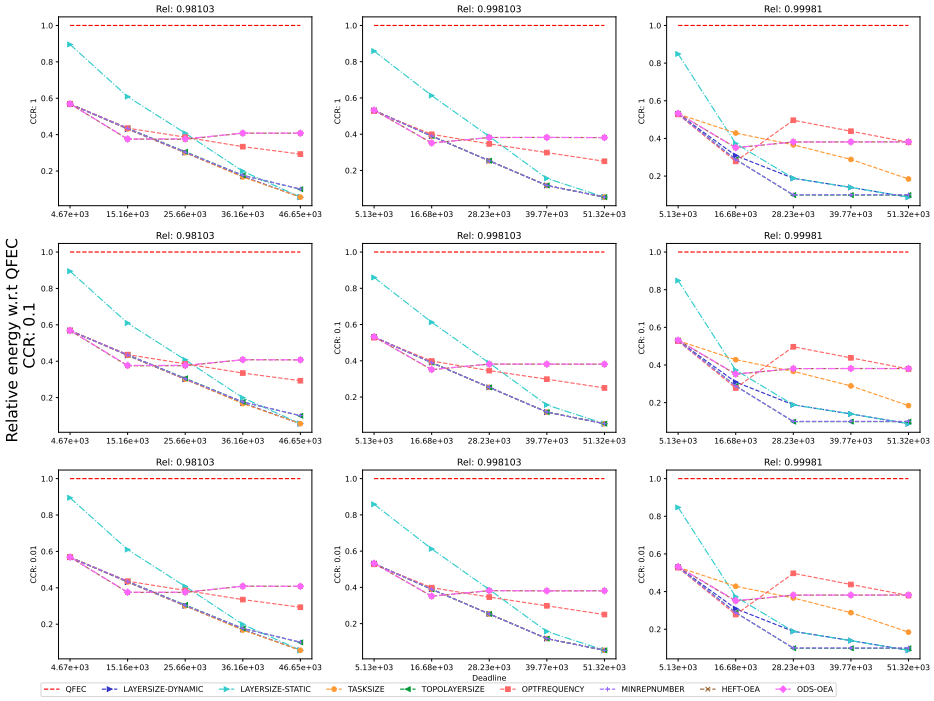

Figure 220: Performance of the different heuristics on the LU workflow.

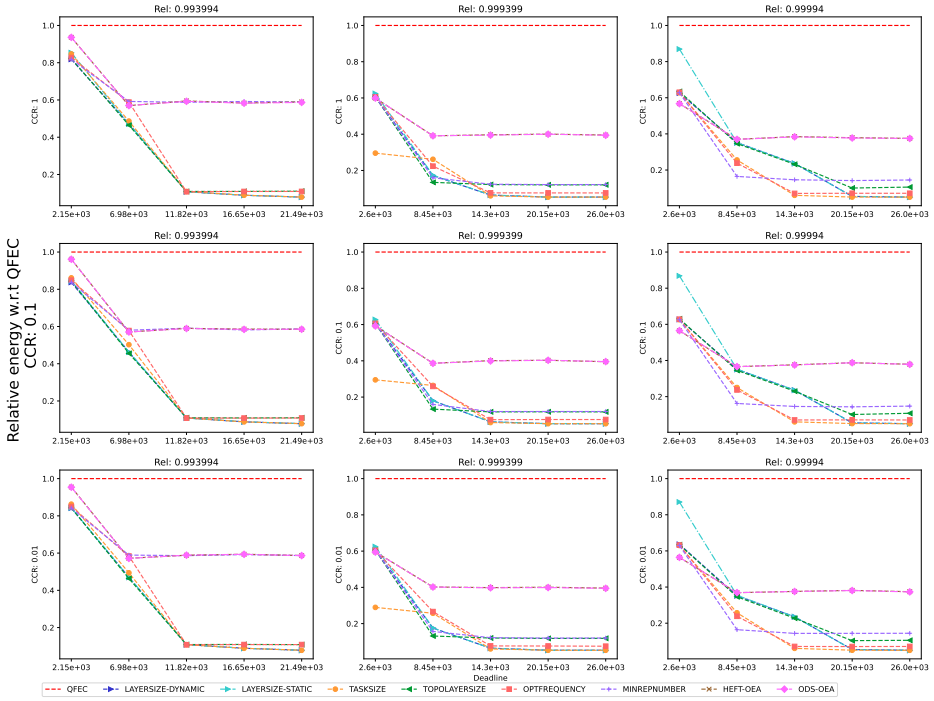

Figure 221: Performance of the different heuristics on the Montage workflow.

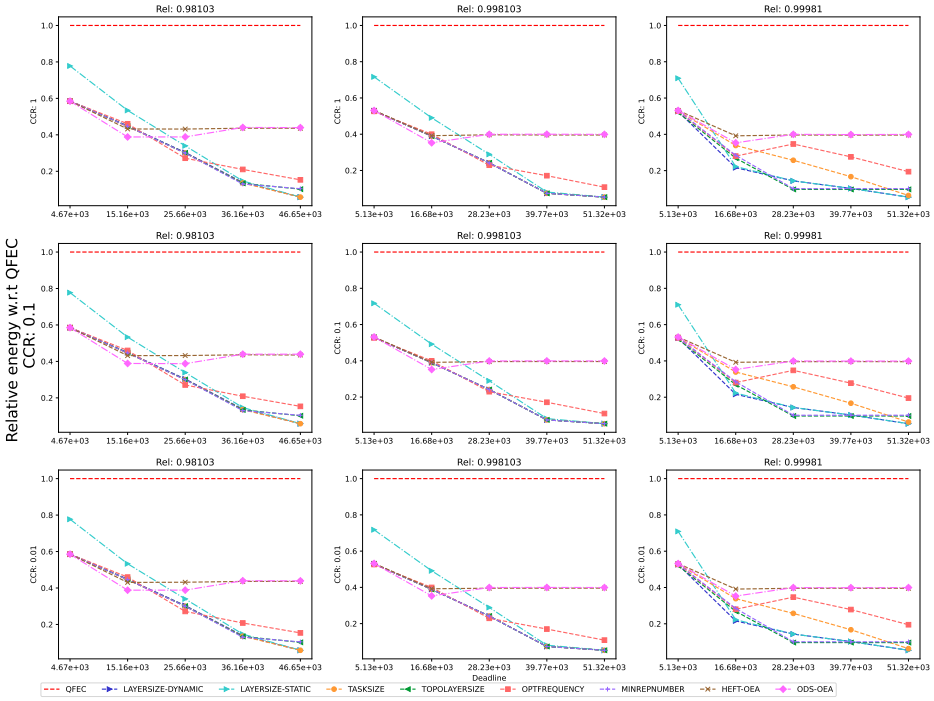

Figure 222: Performance of the different heuristics on the QR workflow.

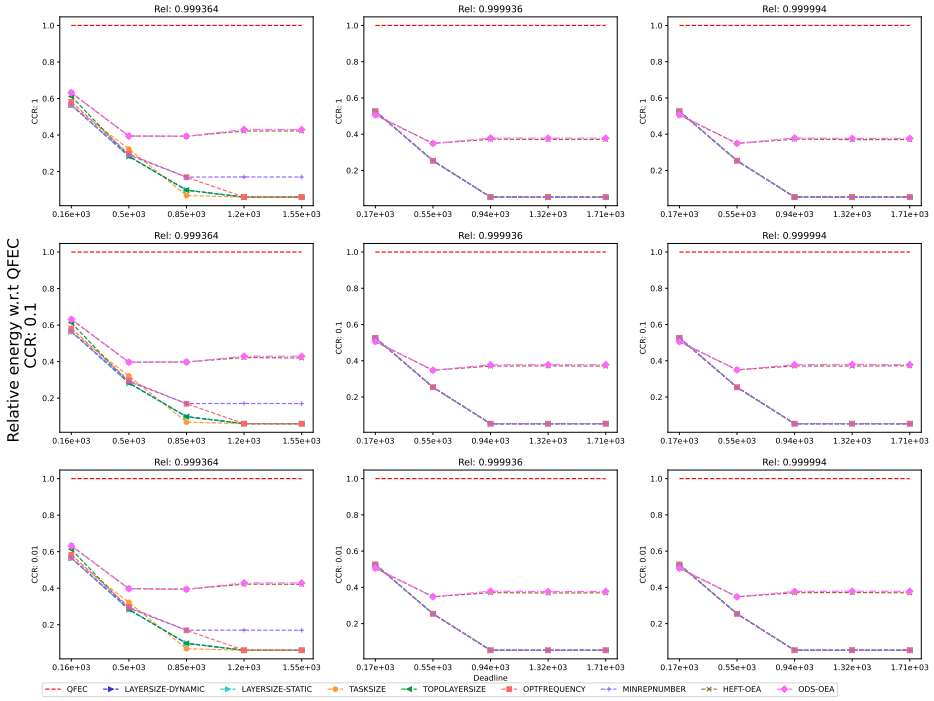

Figure 223: Performance of the different heuristics on the Seismology workflow.

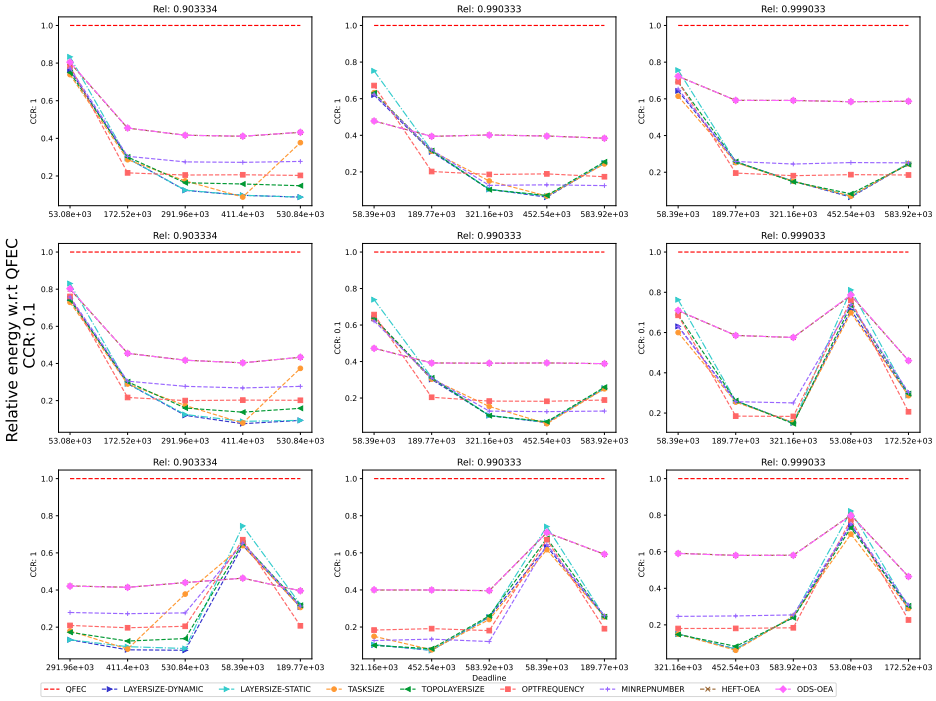

Figure 224: Performance of the different heuristics on the SoyKB workflow.

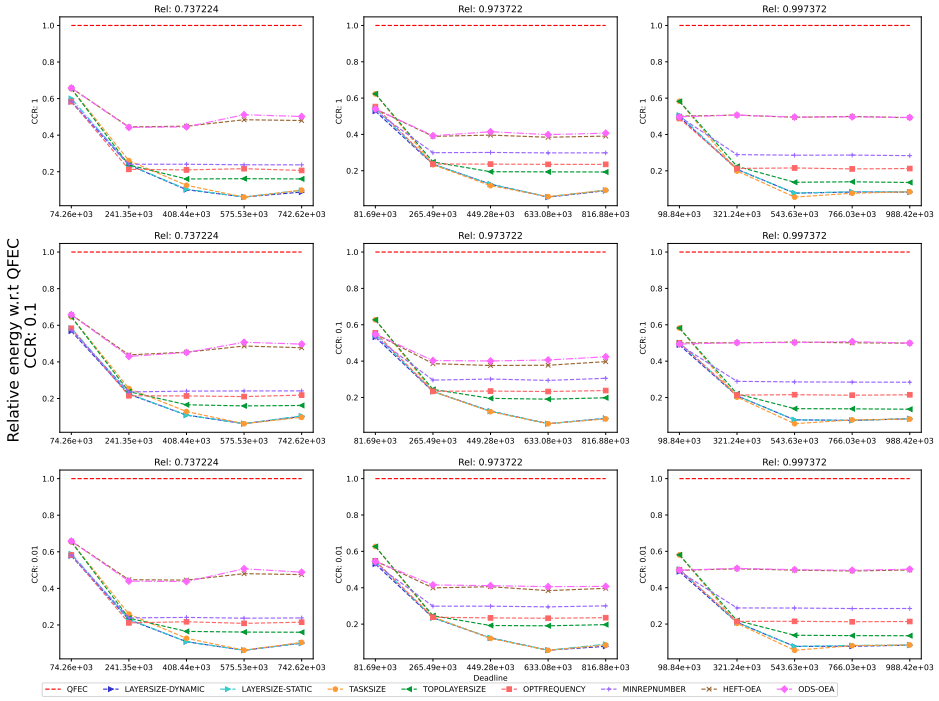

Figure 225: Performance of the different heuristics on the SRASearch.

D.5.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

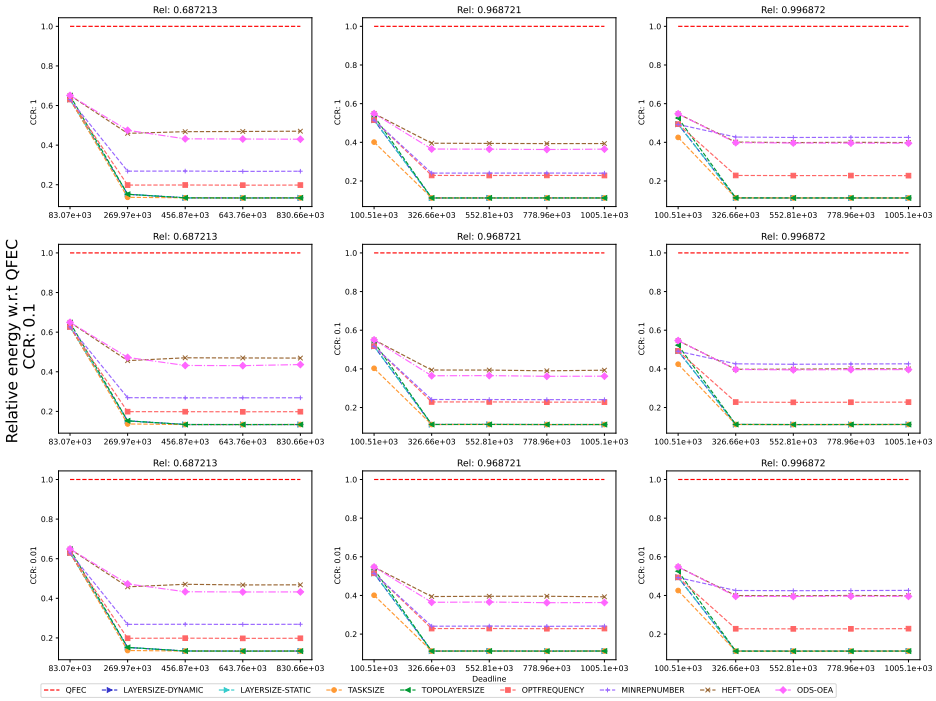

Figure 226: Performance of the different heuristics on the BLAST workflow.

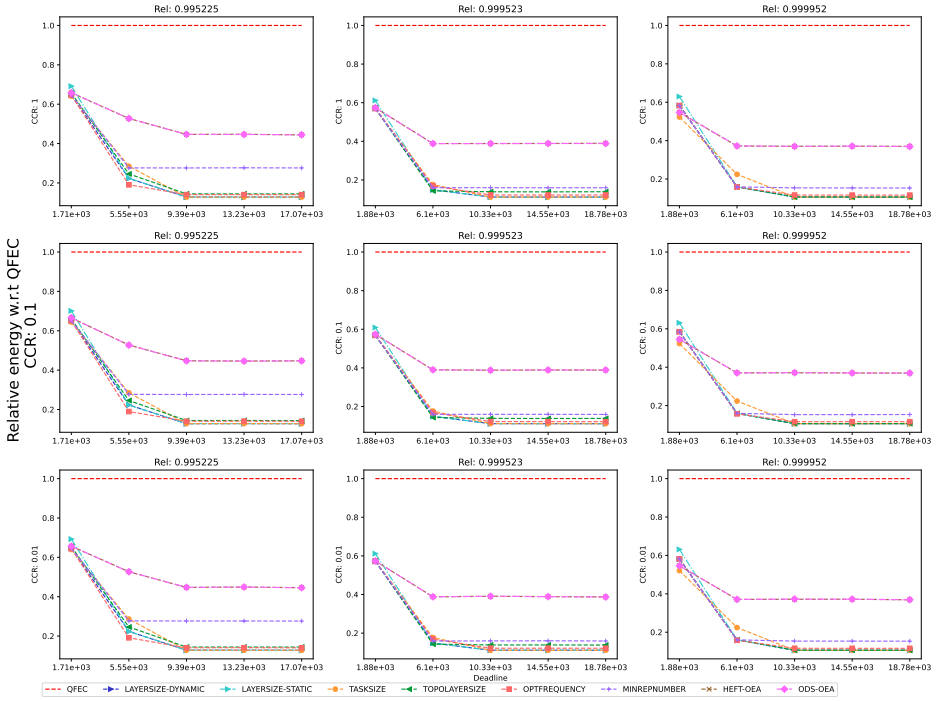

Figure 227: Performance of the different heuristics on the BWA workflow.

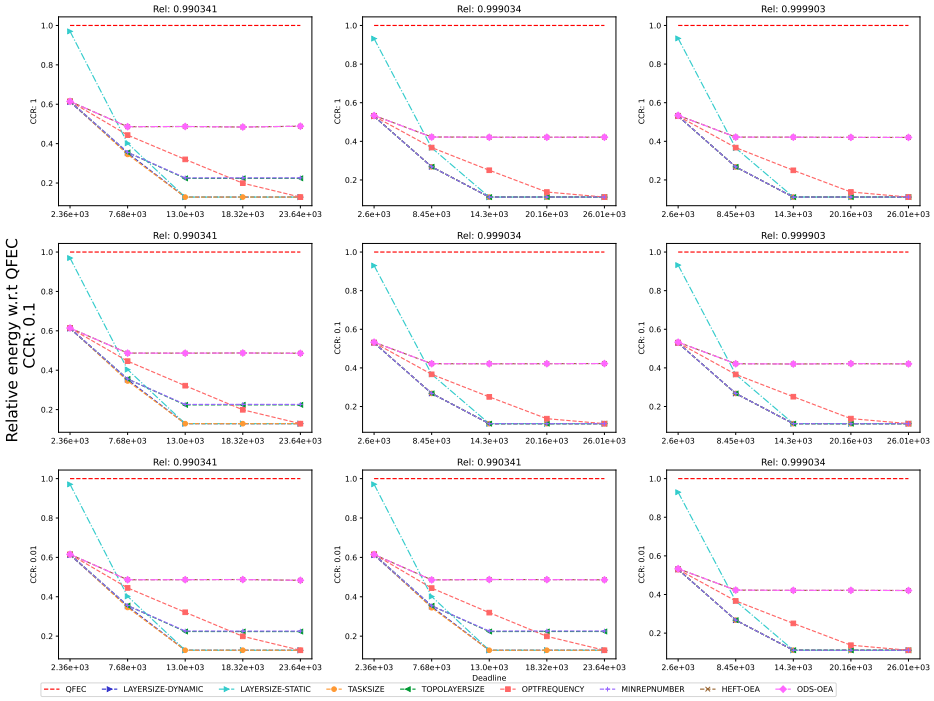

Figure 228: Performance of the different heuristics on the Cholesky workflow.

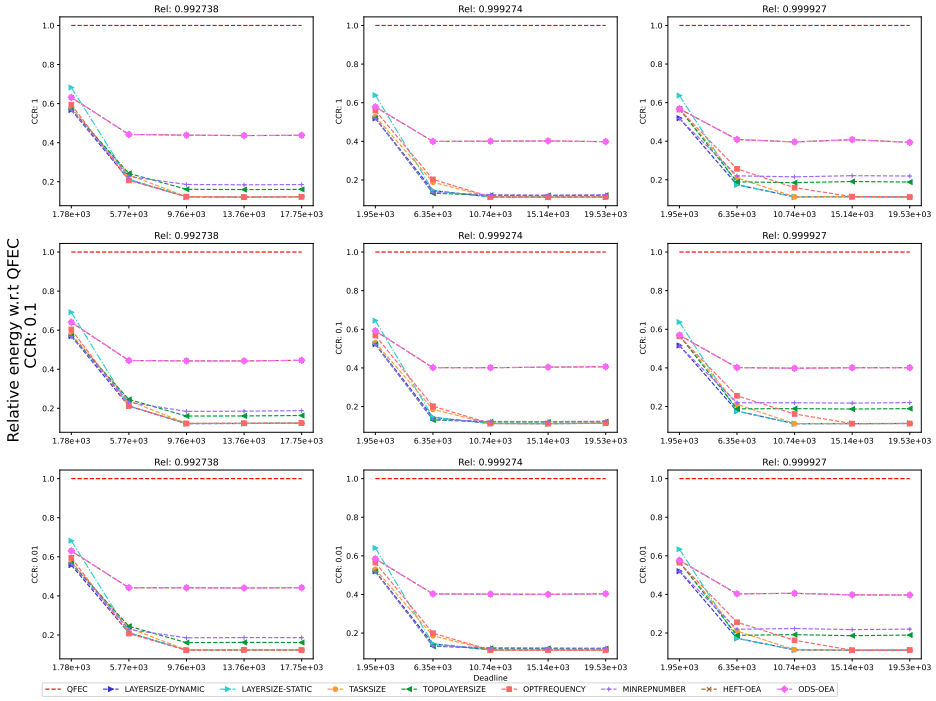

Figure 229: Performance of the different heuristics on the Cycles workflow.

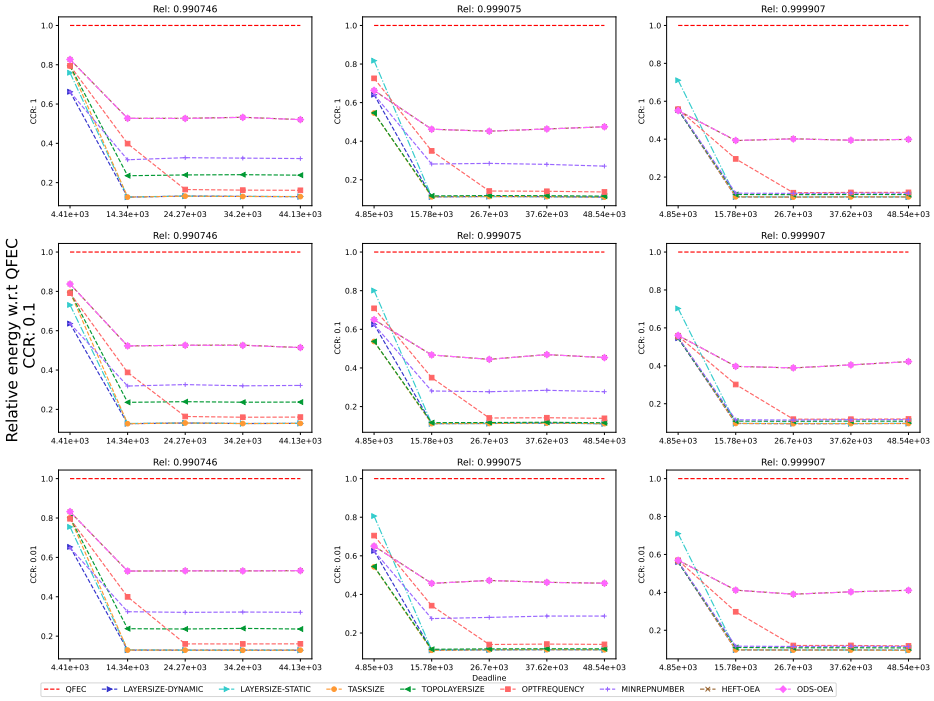

Figure 230: Performance of the different heuristics on the Epigenomics workflow.

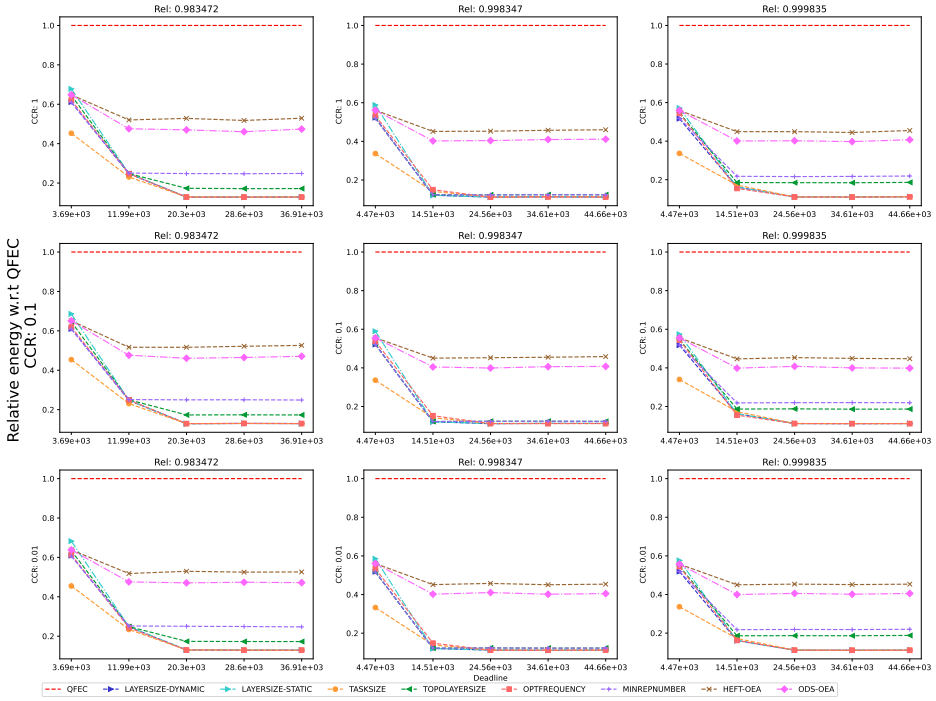

Figure 231: Performance of the different heuristics on the Genome workflow.

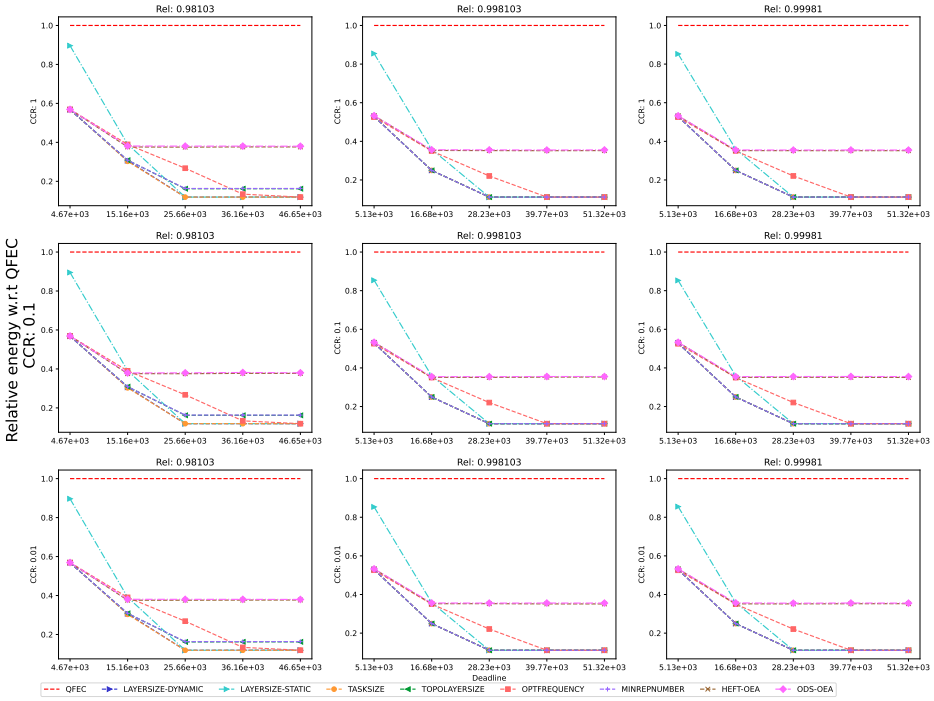

Figure 232: Performance of the different heuristics on the LU workflow.

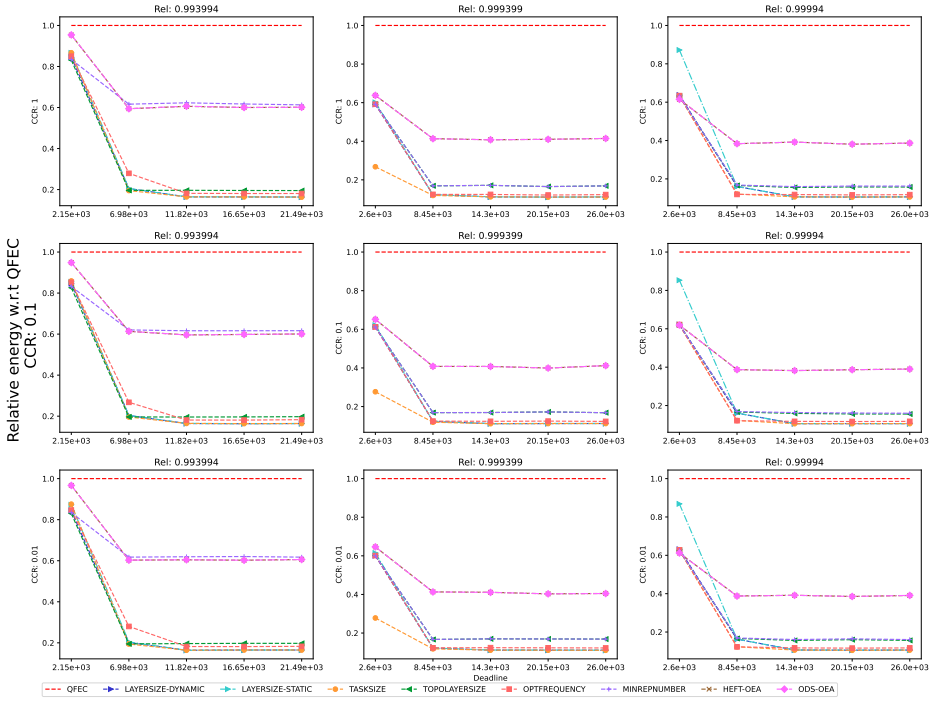

Figure 233: Performance of the different heuristics on the Montage workflow.

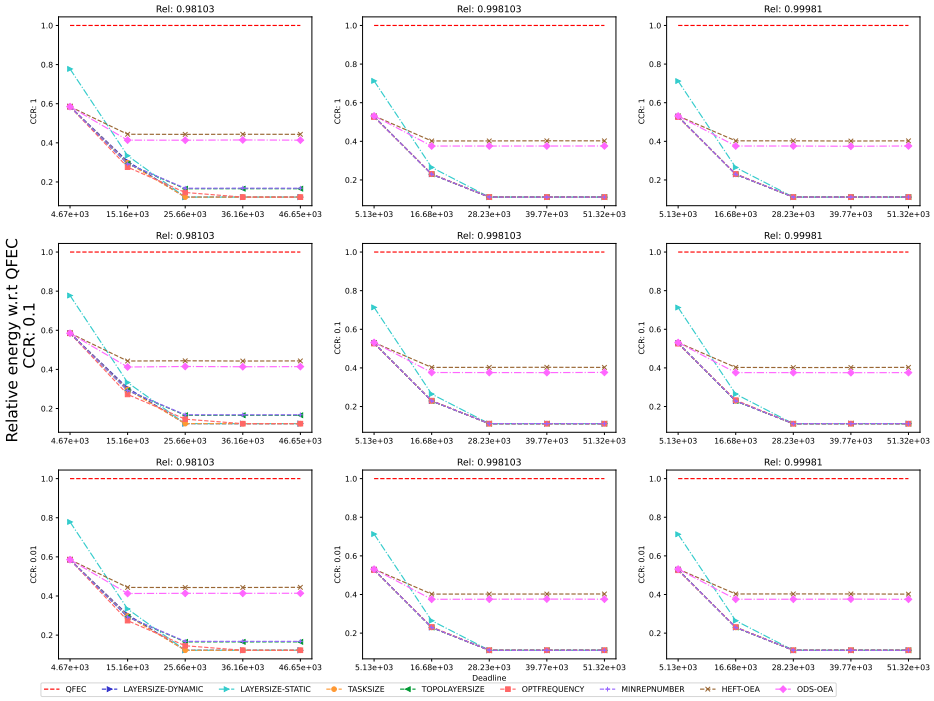

Figure 234: Performance of the different heuristics on the QR workflow.

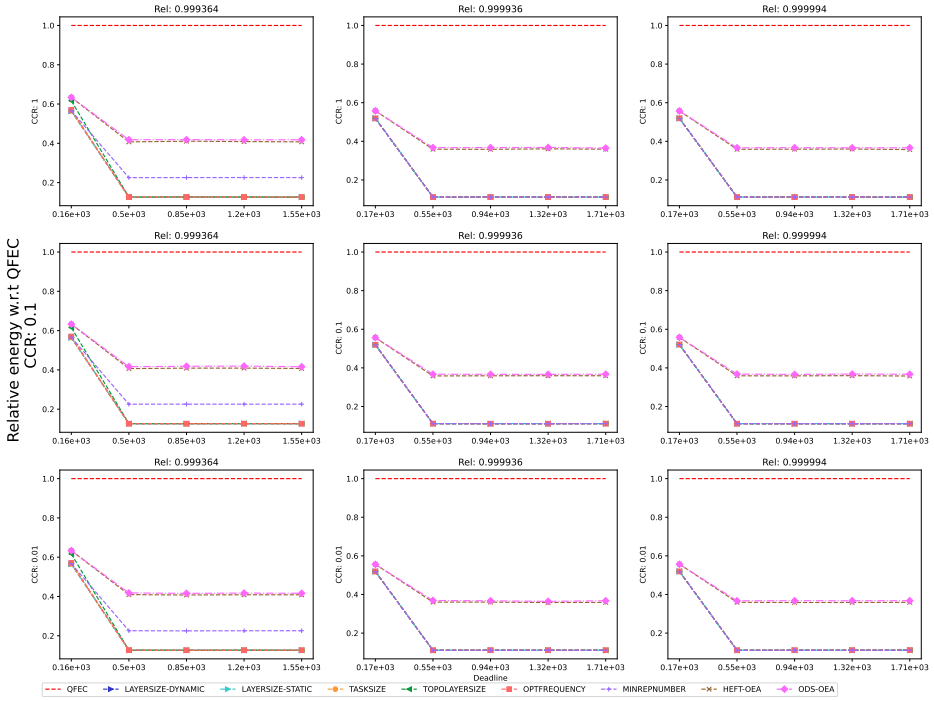

Figure 235: Performance of the different heuristics on the Seismology workflow.

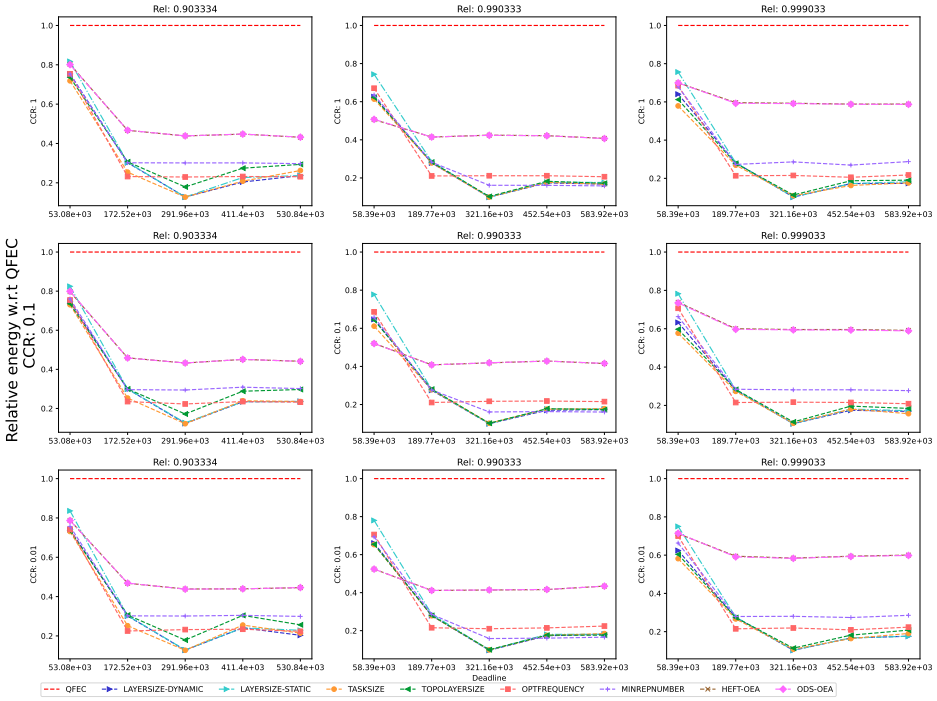

Figure 236: Performance of the different heuristics on the SoyKB workflow.

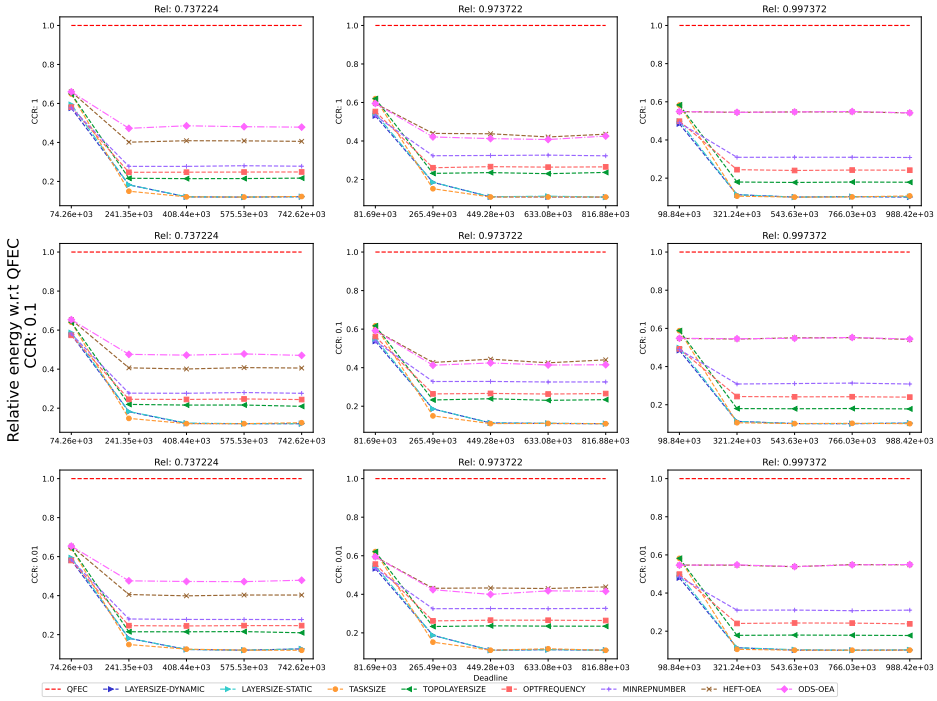

Figure 237: Performance of the different heuristics on the SRASearch.

D.5.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

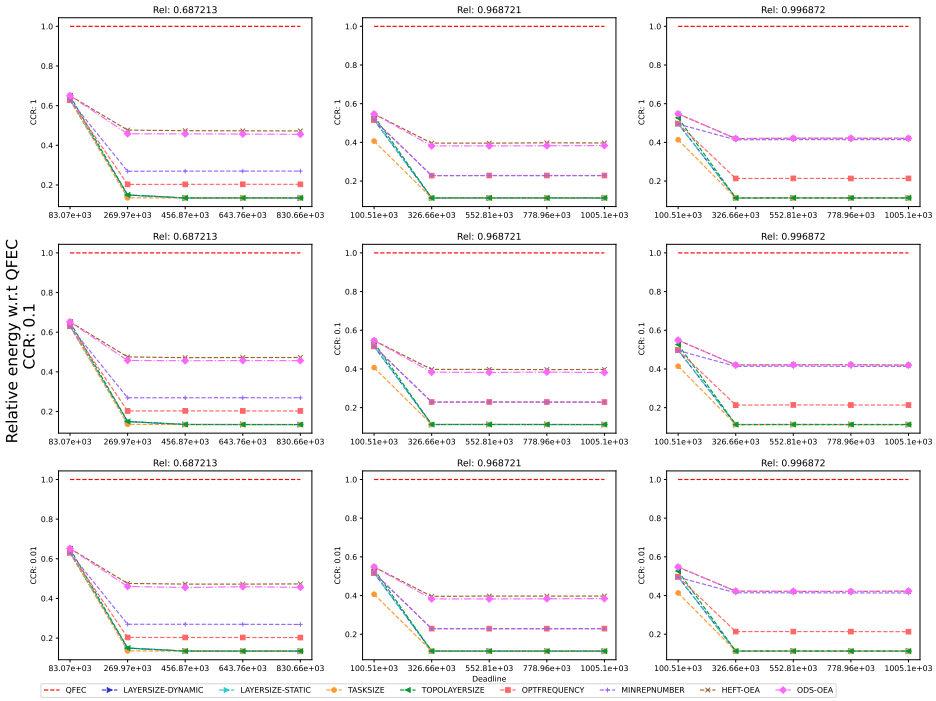

Figure 238: Performance of the different heuristics on the BLAST workflow.

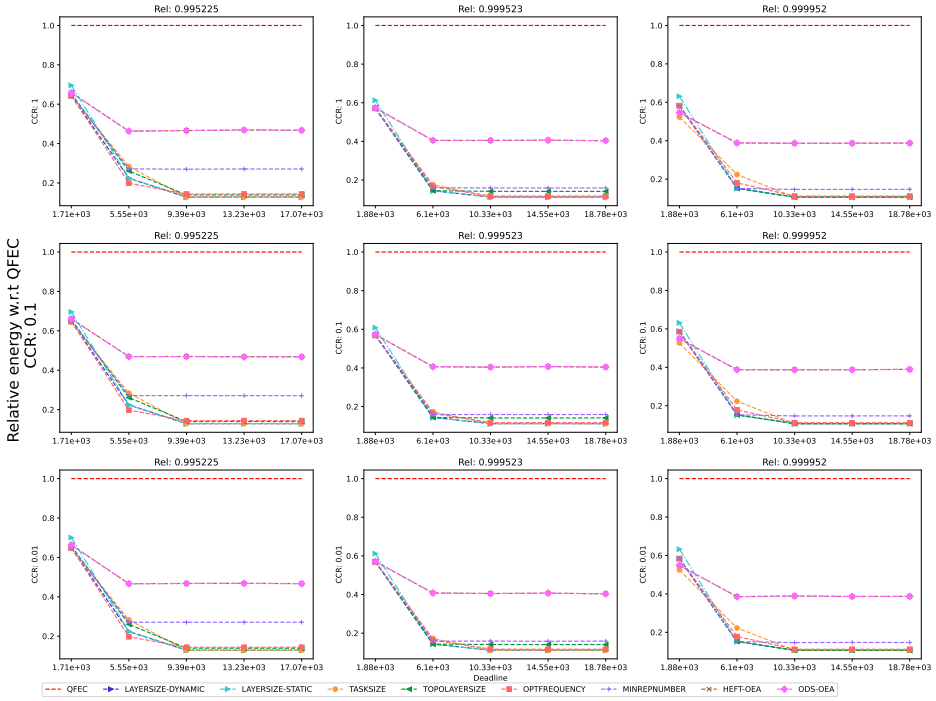

Figure 239: Performance of the different heuristics on the BWA workflow.

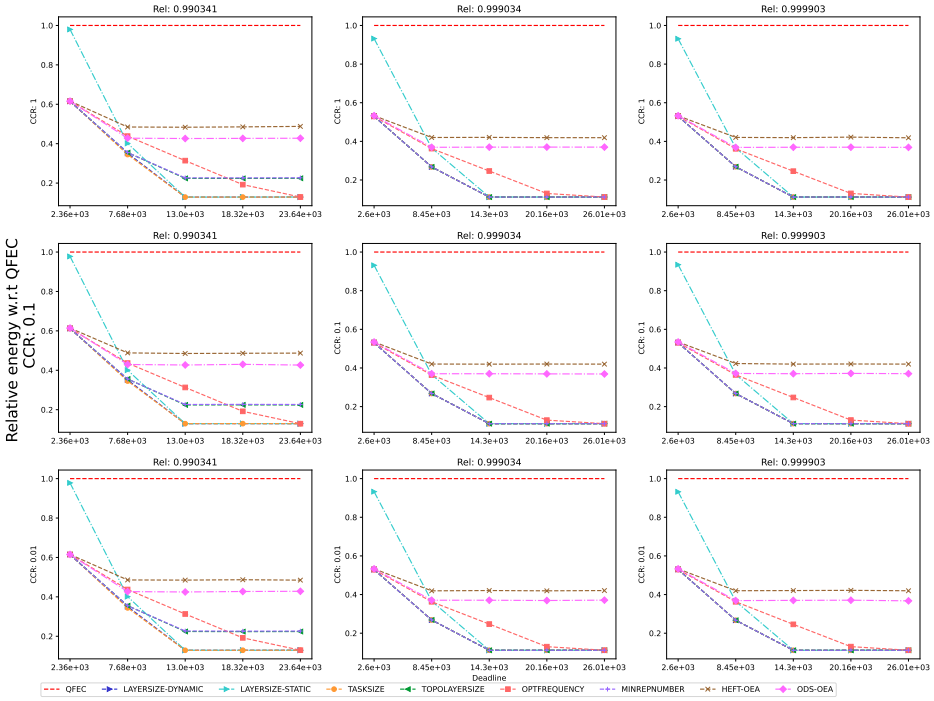

Figure 240: Performance of the different heuristics on the Cholesky workflow.

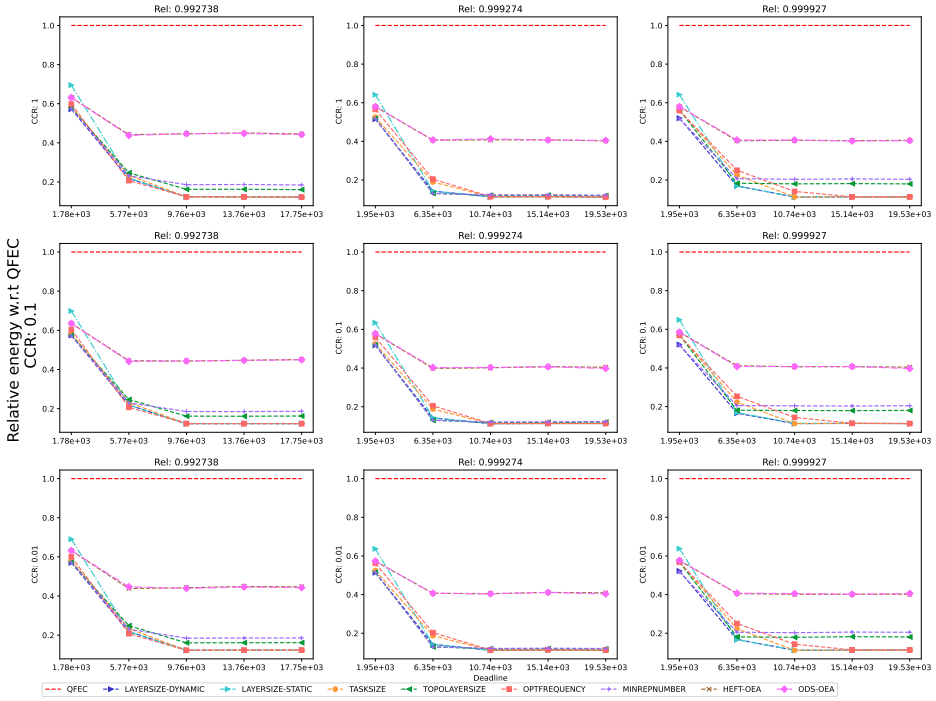

Figure 241: Performance of the different heuristics on the Cycles workflow.

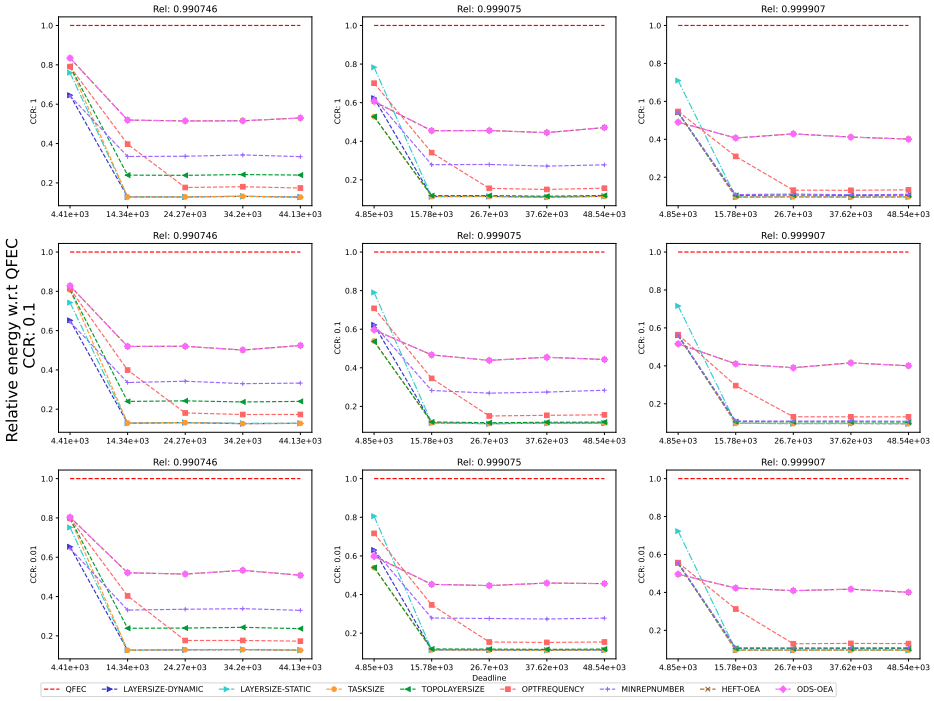

Figure 242: Performance of the different heuristics on the Epigenomics workflow.

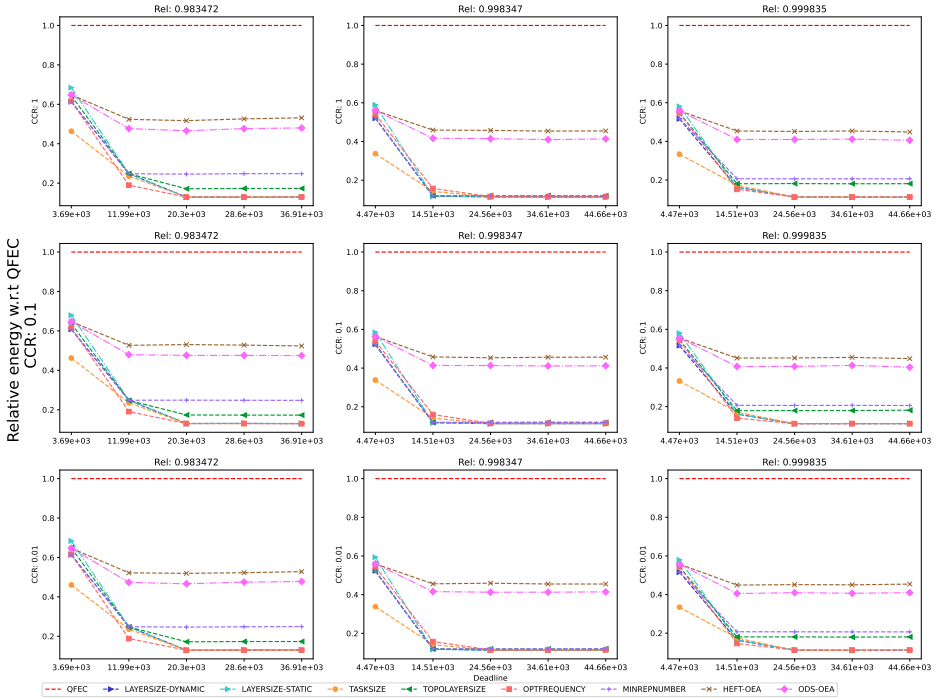

Figure 243: Performance of the different heuristics on the Genome workflow.

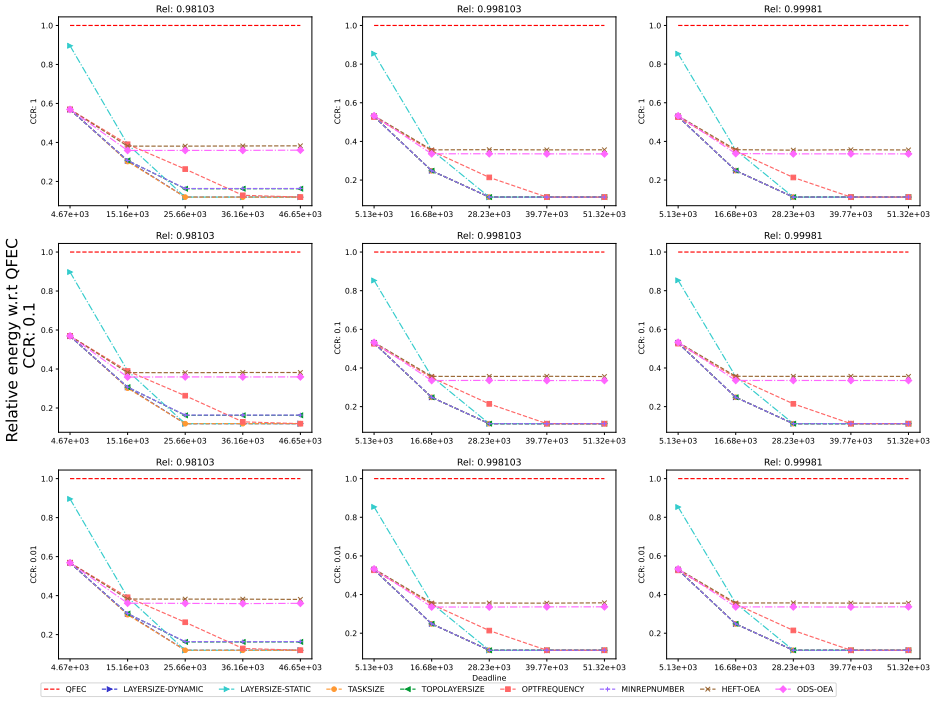

Figure 244: Performance of the different heuristics on the LU workflow.

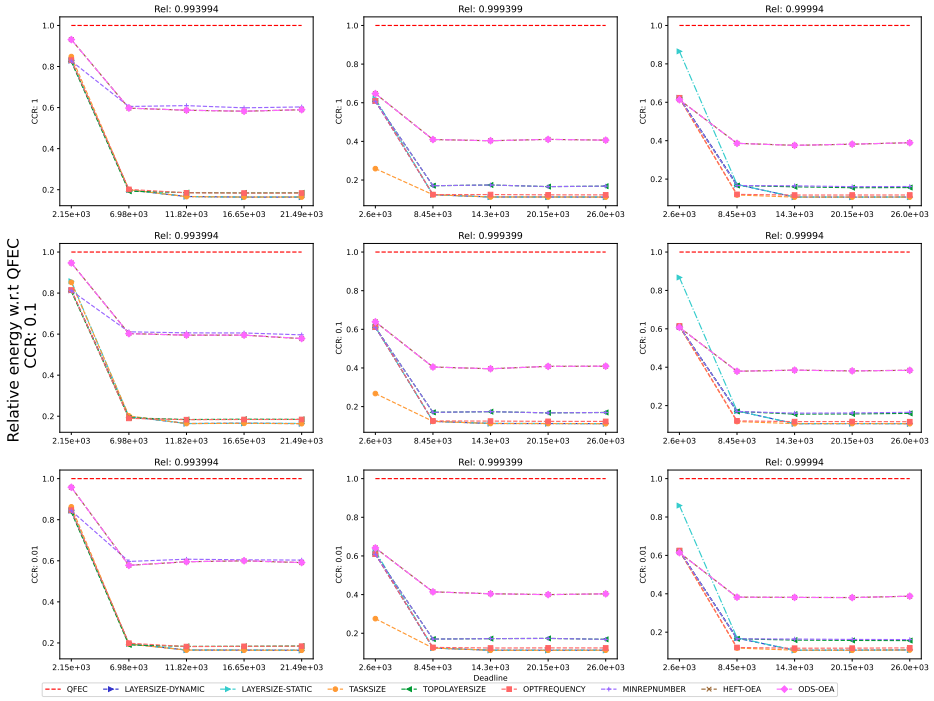

Figure 245: Performance of the different heuristics on the Montage workflow.

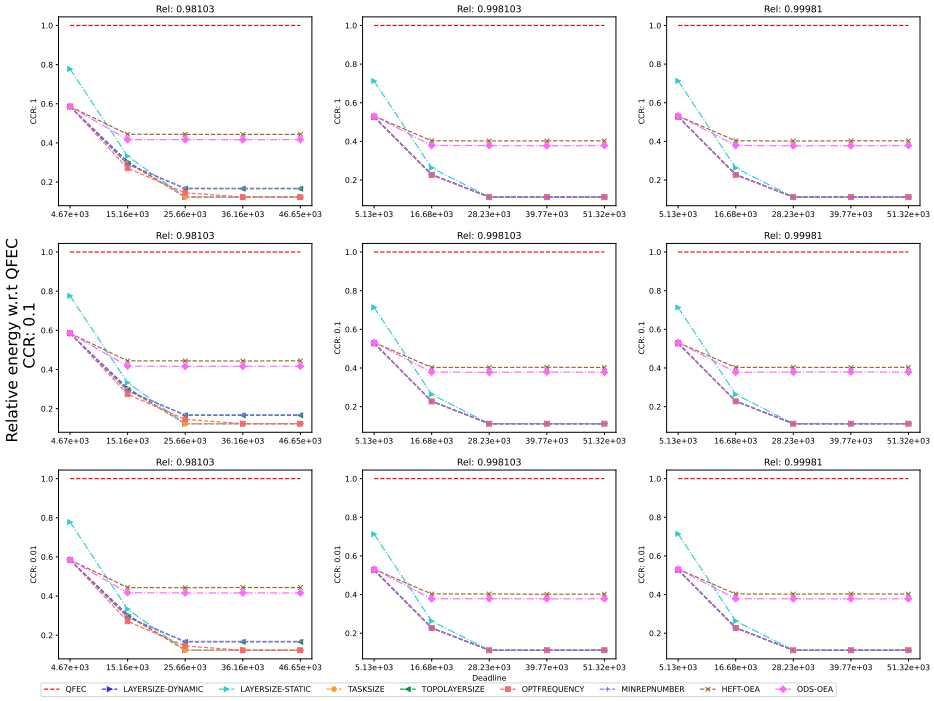

Figure 246: Performance of the different heuristics on the QR workflow.

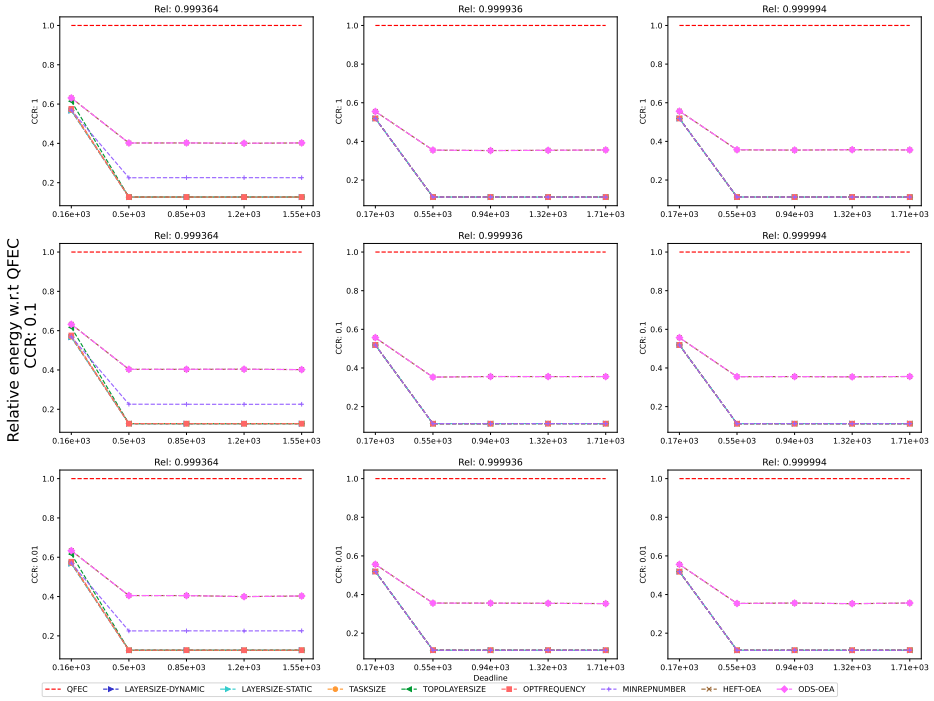

Figure 247: Performance of the different heuristics on the Seismology workflow.

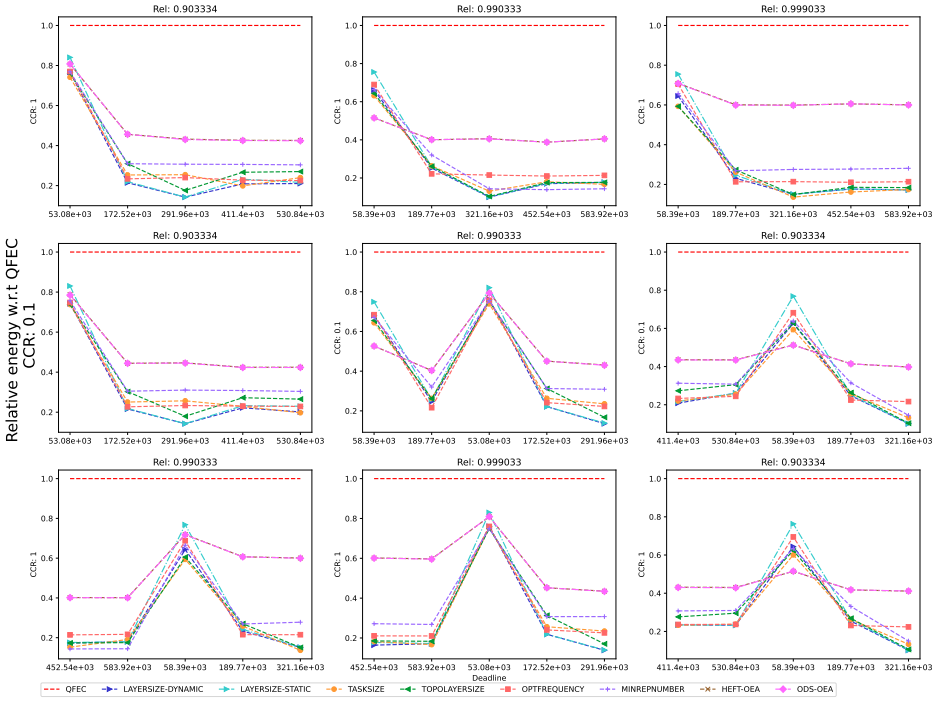

Figure 248: Performance of the different heuristics on the SoyKB workflow.

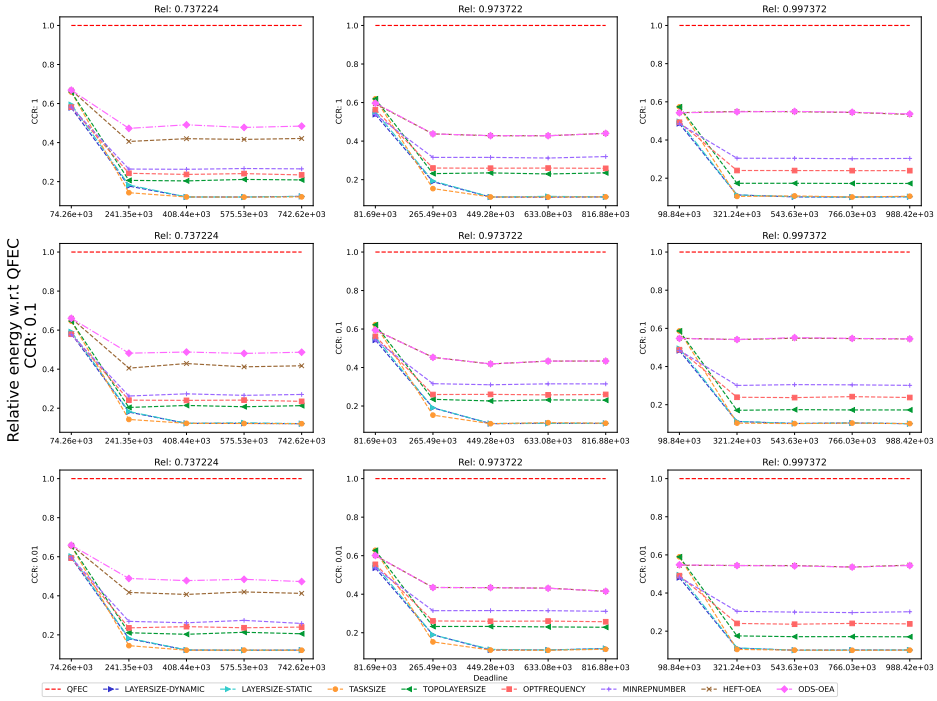

Figure 249: Performance of the different heuristics on the SRASearch.

D.6 $BC/WC = 0.6$

D.6.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

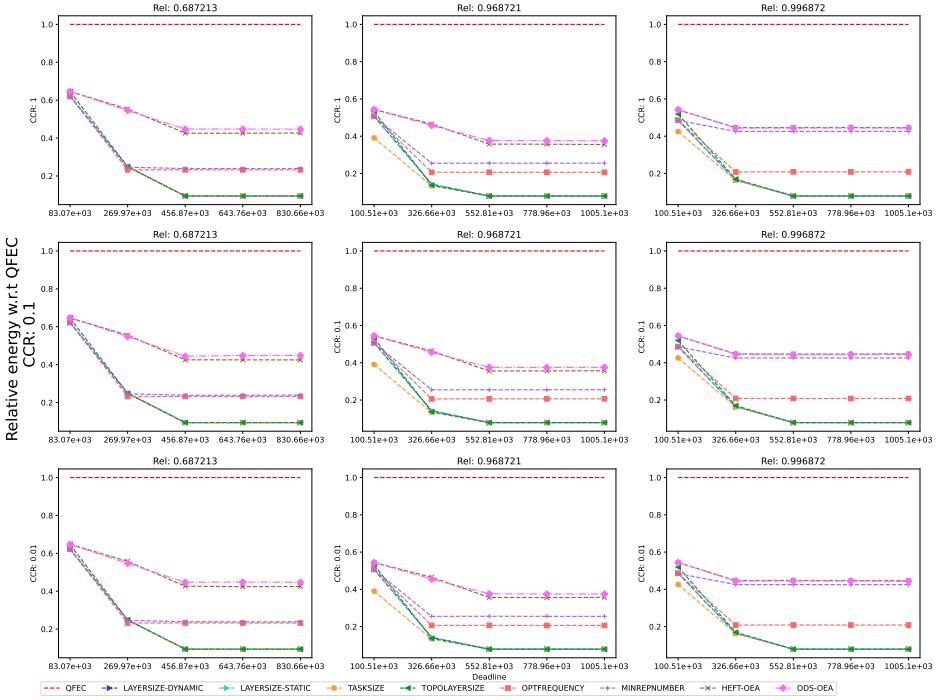

Figure 250: Performance of the different heuristics on the BLAST workflow.

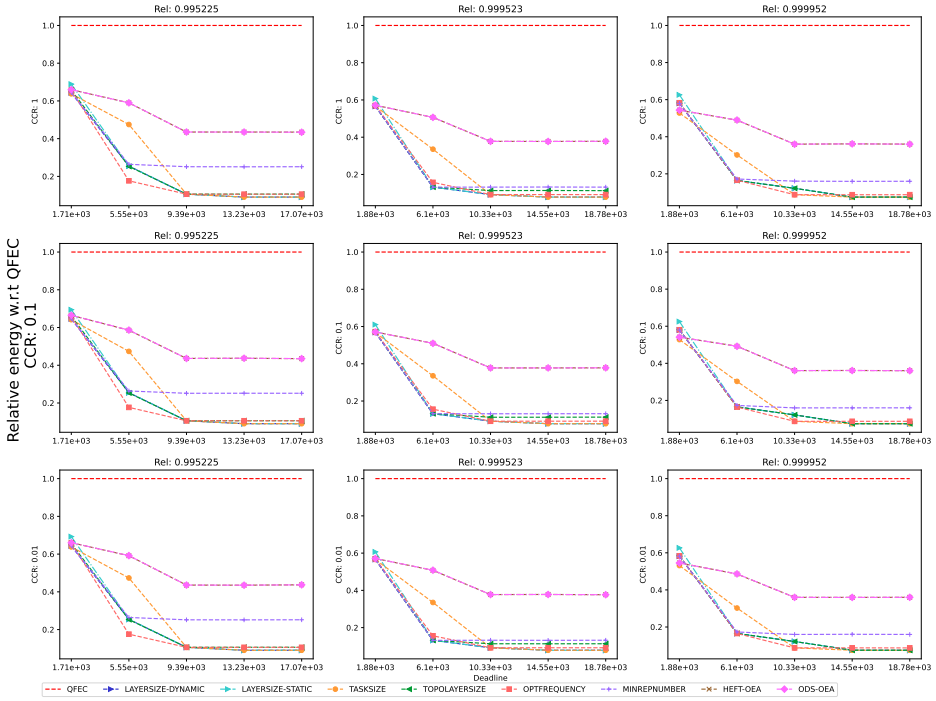

Figure 251: Performance of the different heuristics on the BWA workflow.

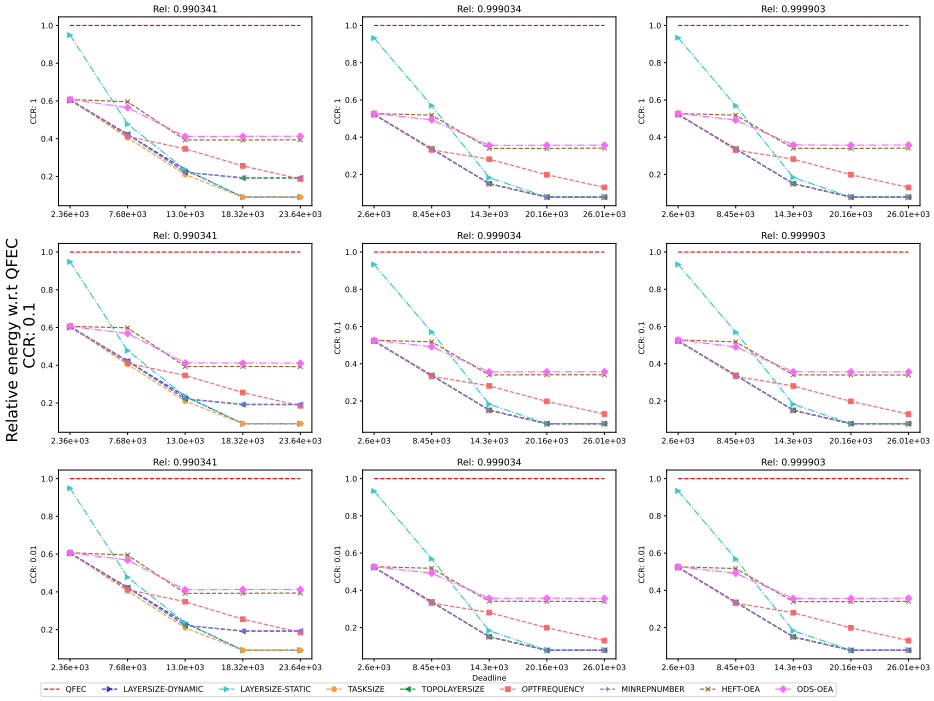

Figure 252: Performance of the different heuristics on the Cholesky workflow.

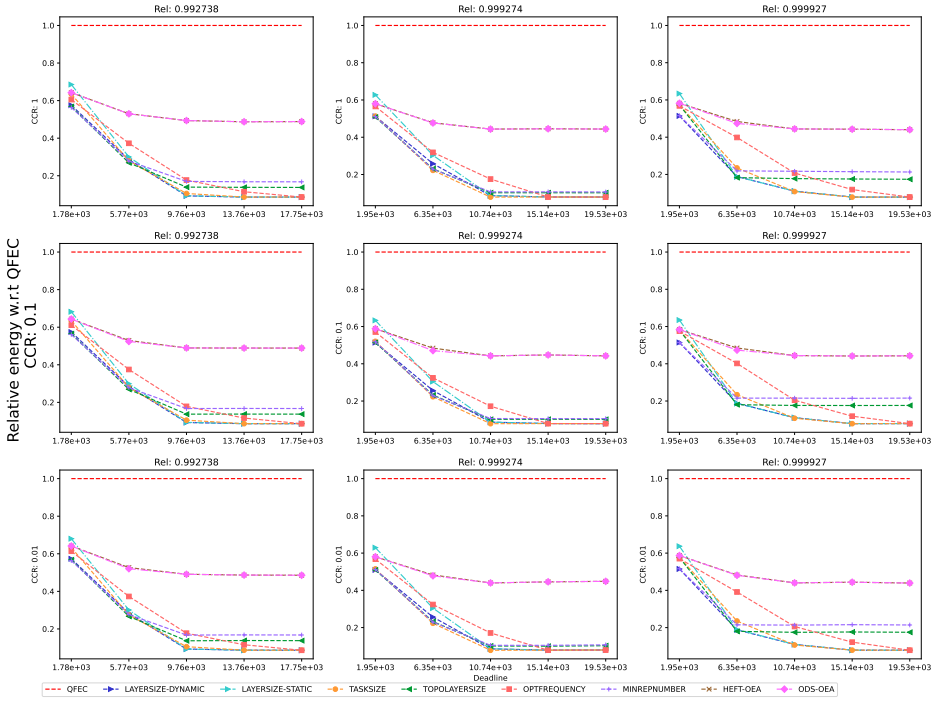

Figure 253: Performance of the different heuristics on the Cycles workflow.

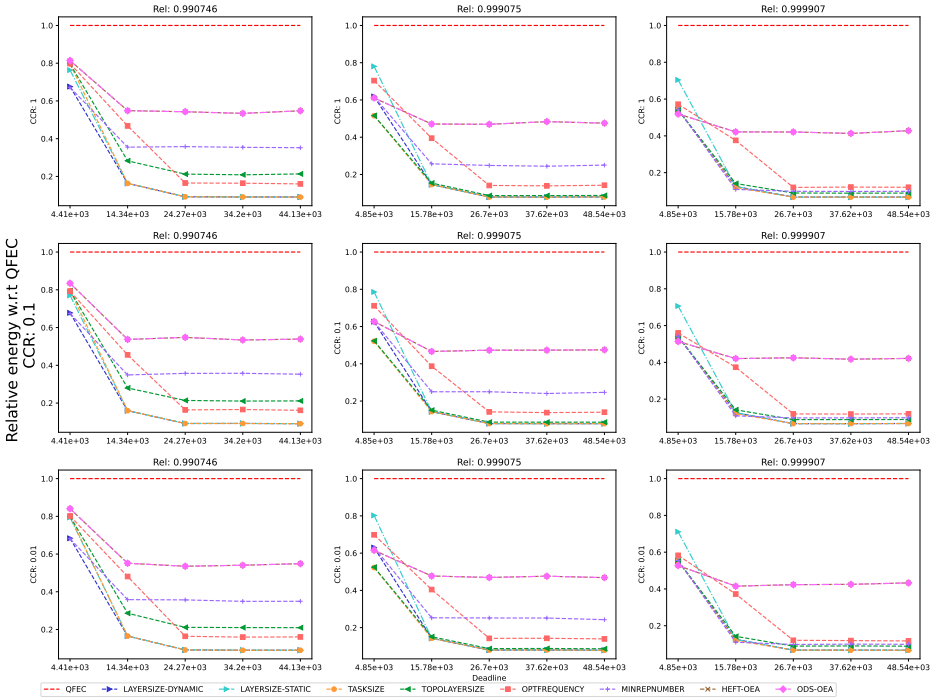

Figure 254: Performance of the different heuristics on the Epigenomics workflow.

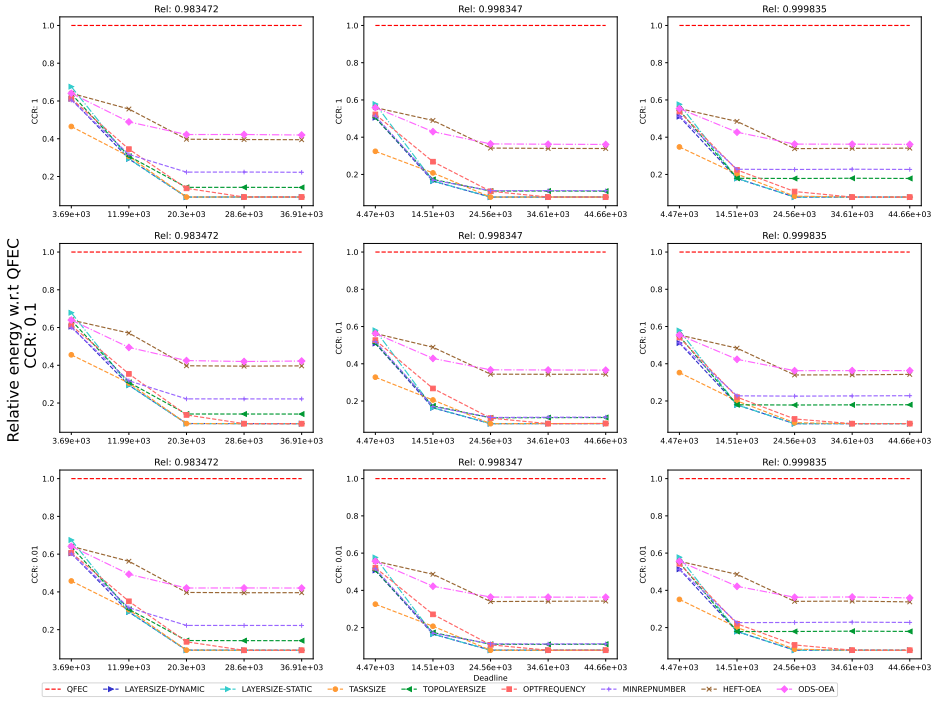

Figure 255: Performance of the different heuristics on the Genome workflow.

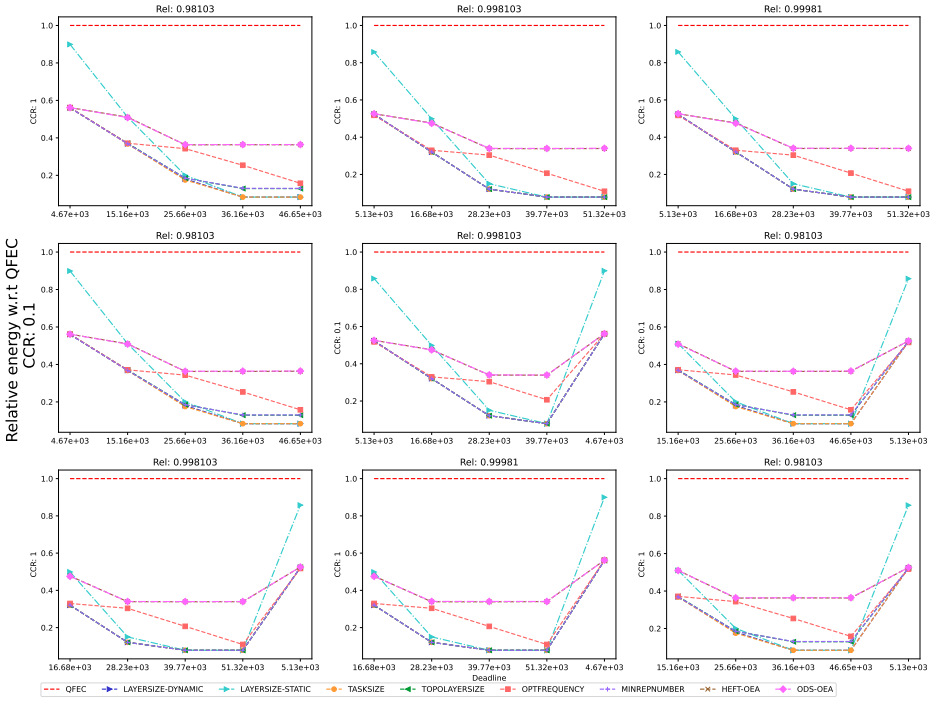

Figure 256: Performance of the different heuristics on the LU workflow.

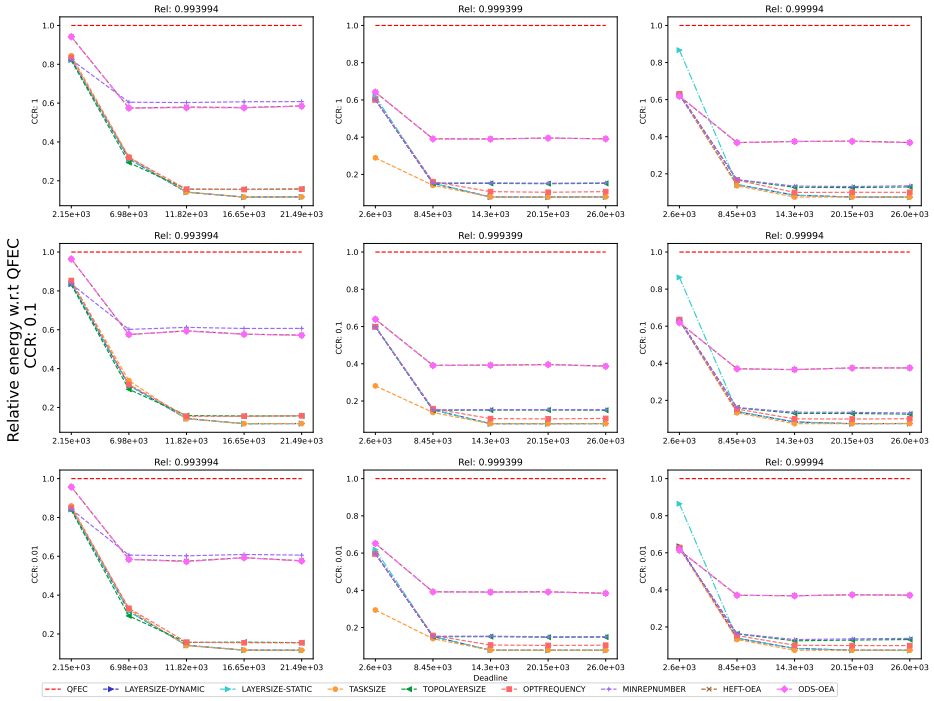

Figure 257: Performance of the different heuristics on the Montage workflow.

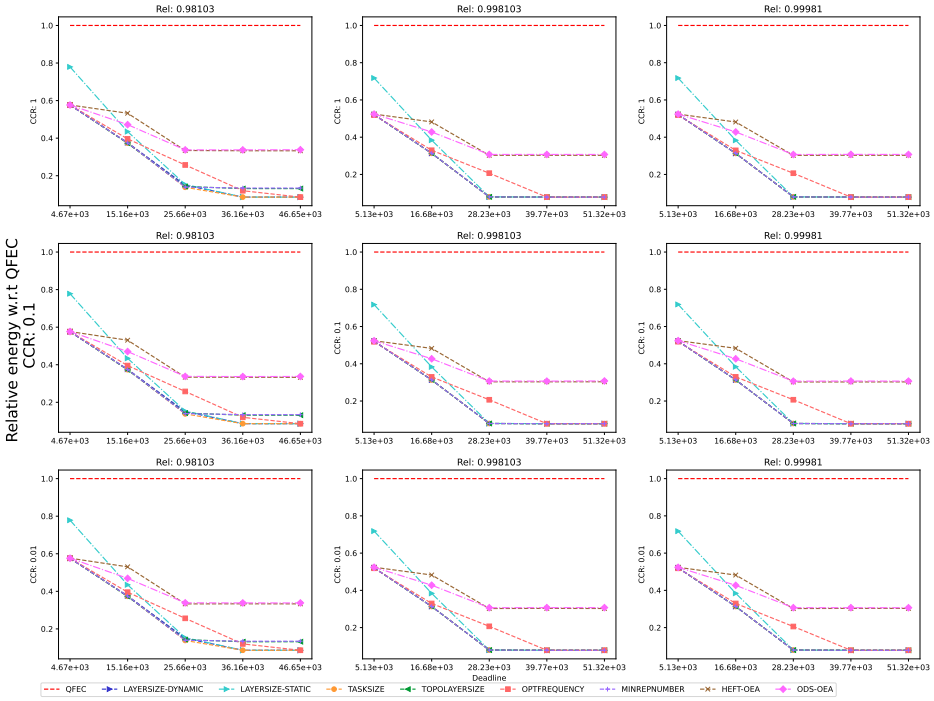

Figure 258: Performance of the different heuristics on the QR workflow.

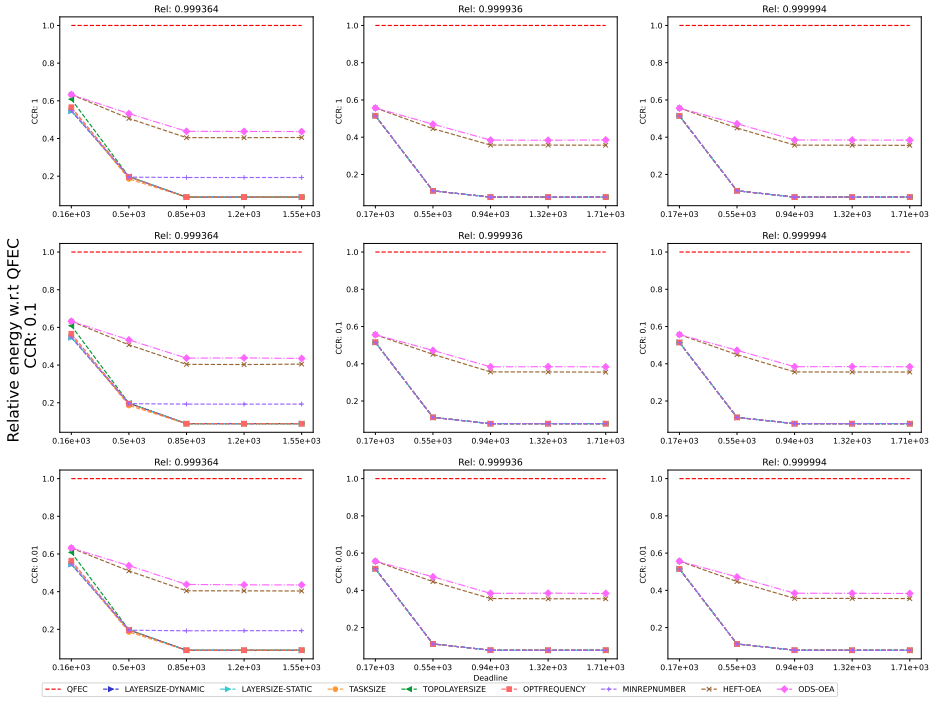

Figure 259: Performance of the different heuristics on the Seismology workflow.

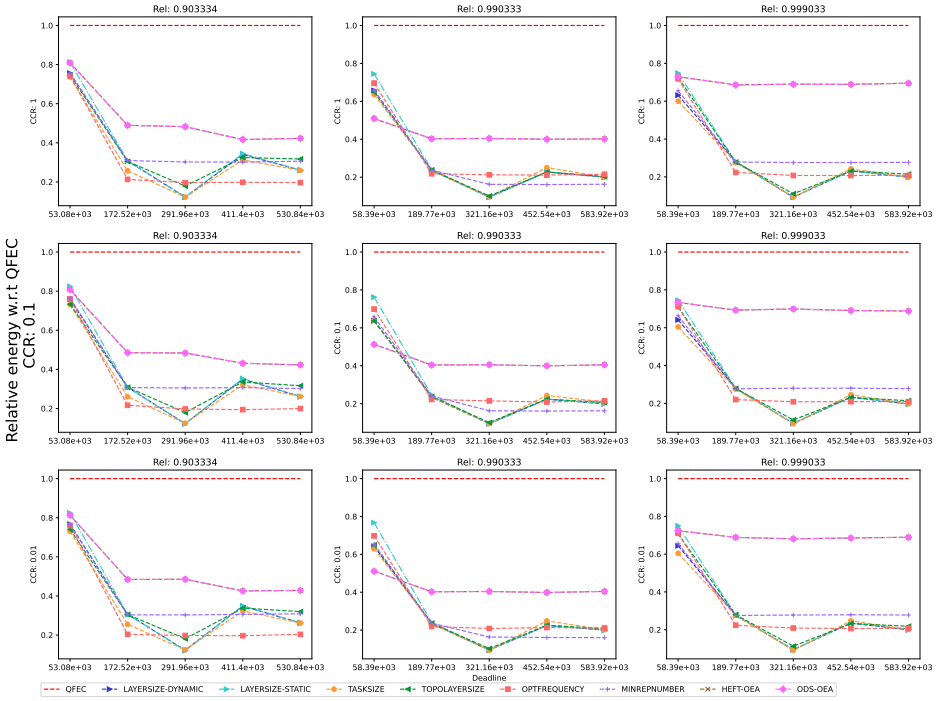

Figure 260: Performance of the different heuristics on the SoyKB workflow.

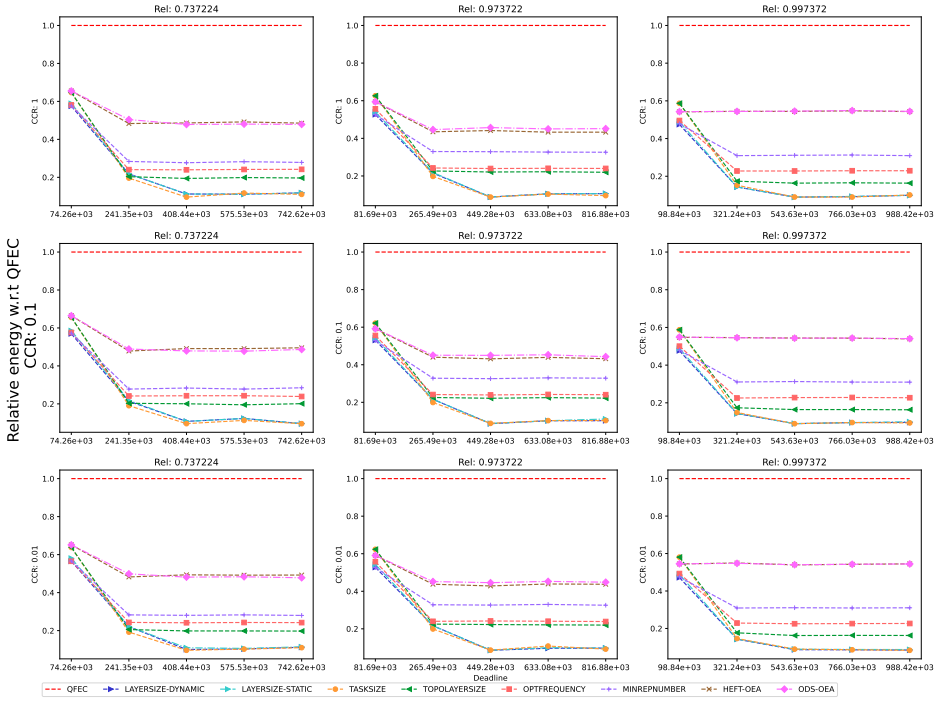

Figure 261: Performance of the different heuristics on the SRASearch.

D.6.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

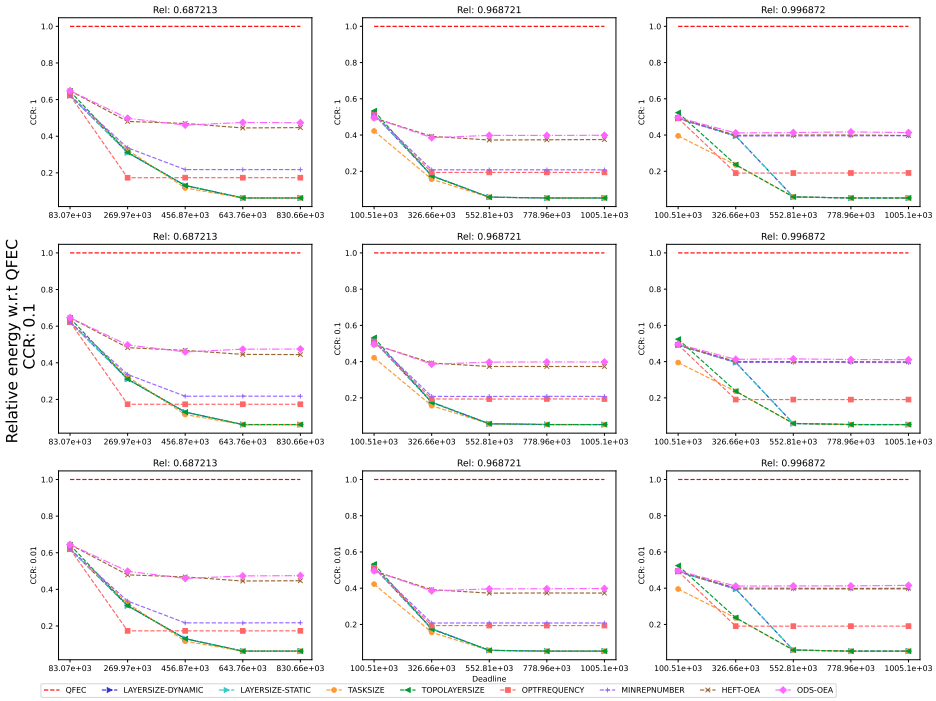

Figure 262: Performance of the different heuristics on the BLAST workflow.

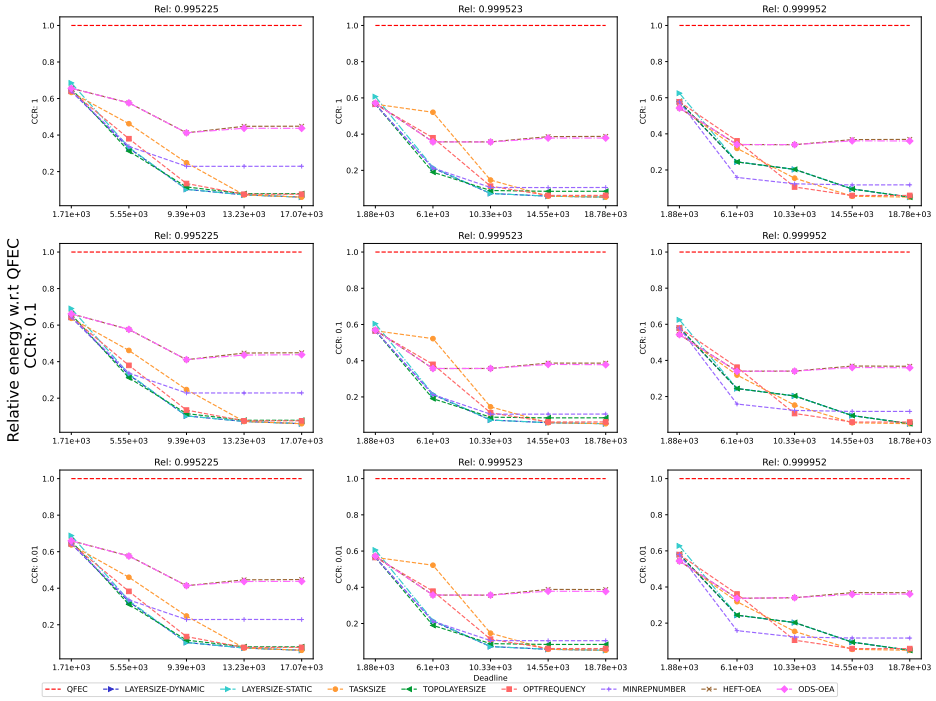

Figure 263: Performance of the different heuristics on the BWA workflow.

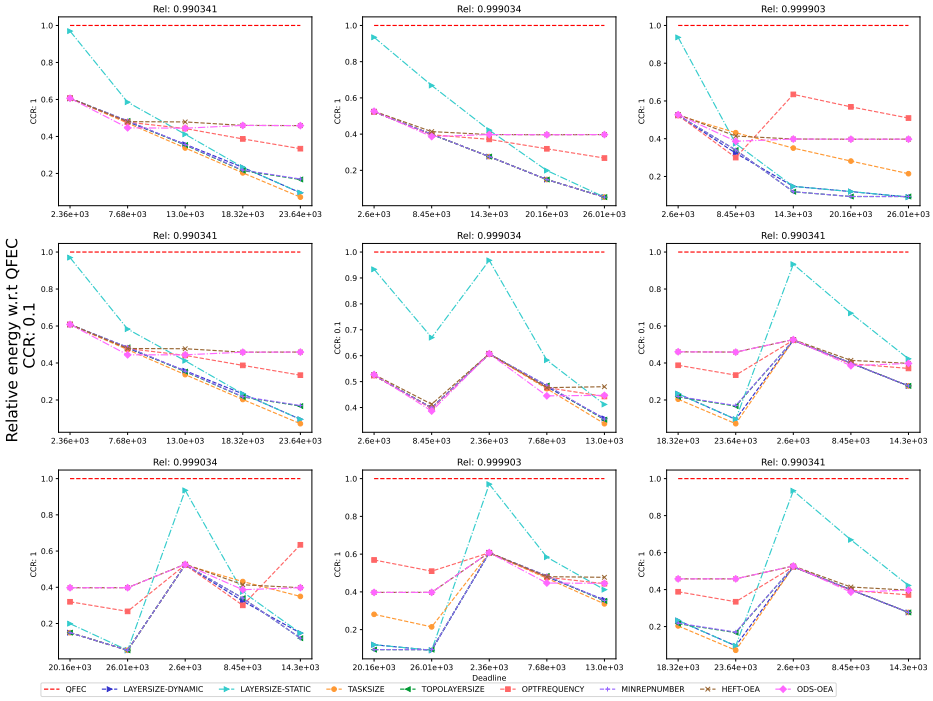

Figure 264: Performance of the different heuristics on the Cholesky workflow.

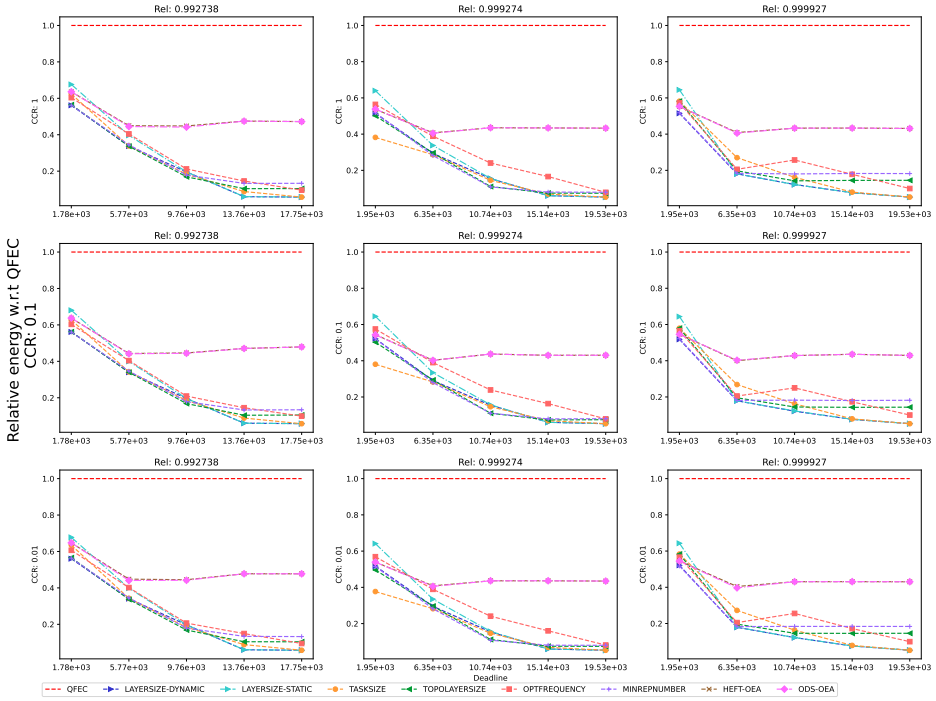

Figure 265: Performance of the different heuristics on the Cycles workflow.

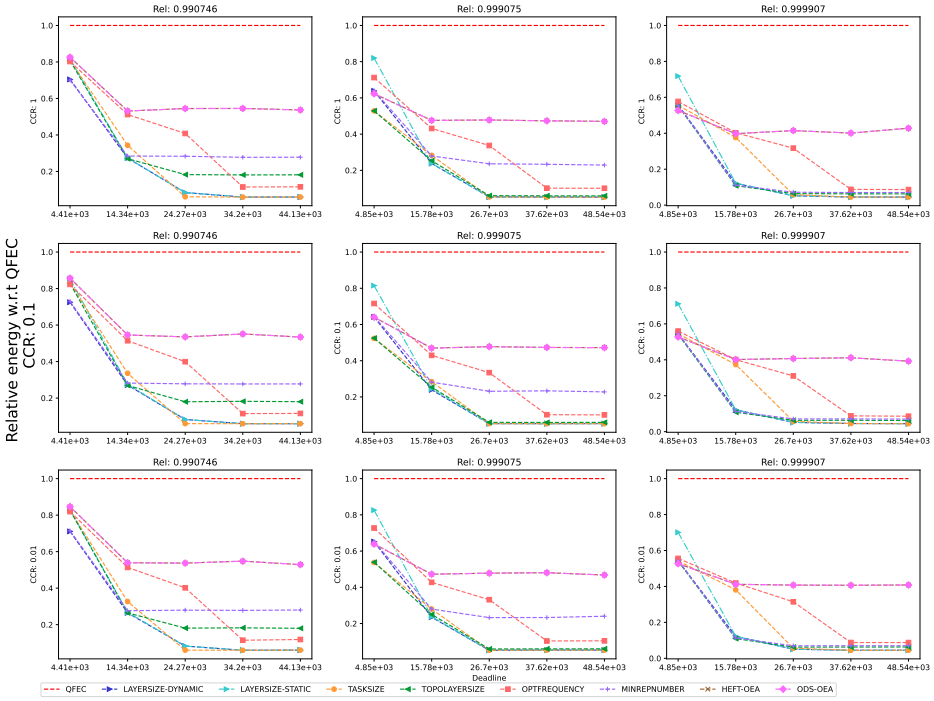

Figure 266: Performance of the different heuristics on the Epigenomics workflow.

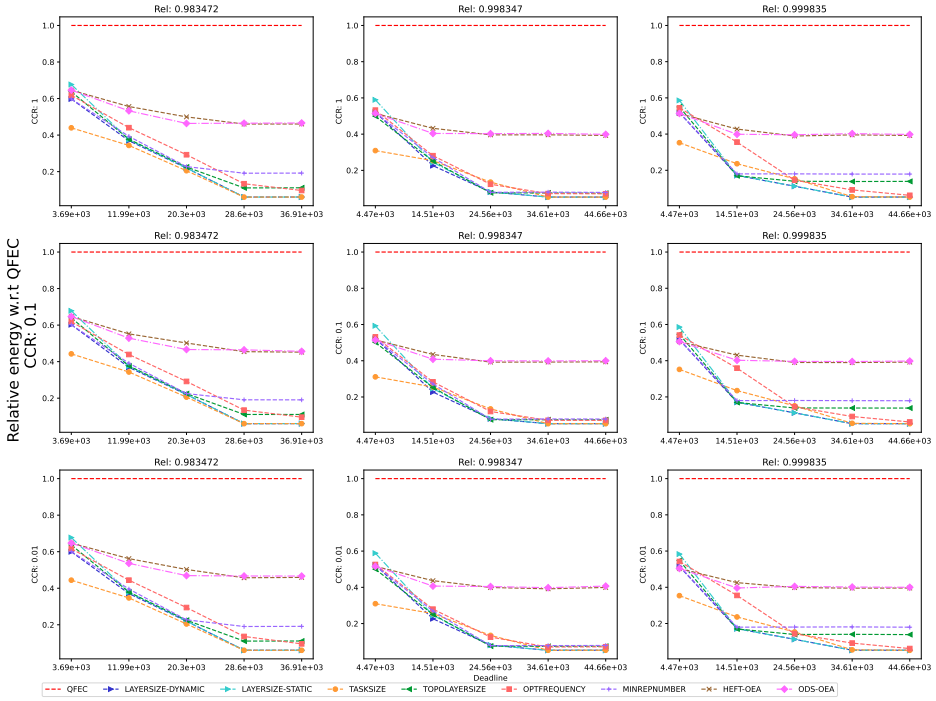

Figure 267: Performance of the different heuristics on the Genome workflow.

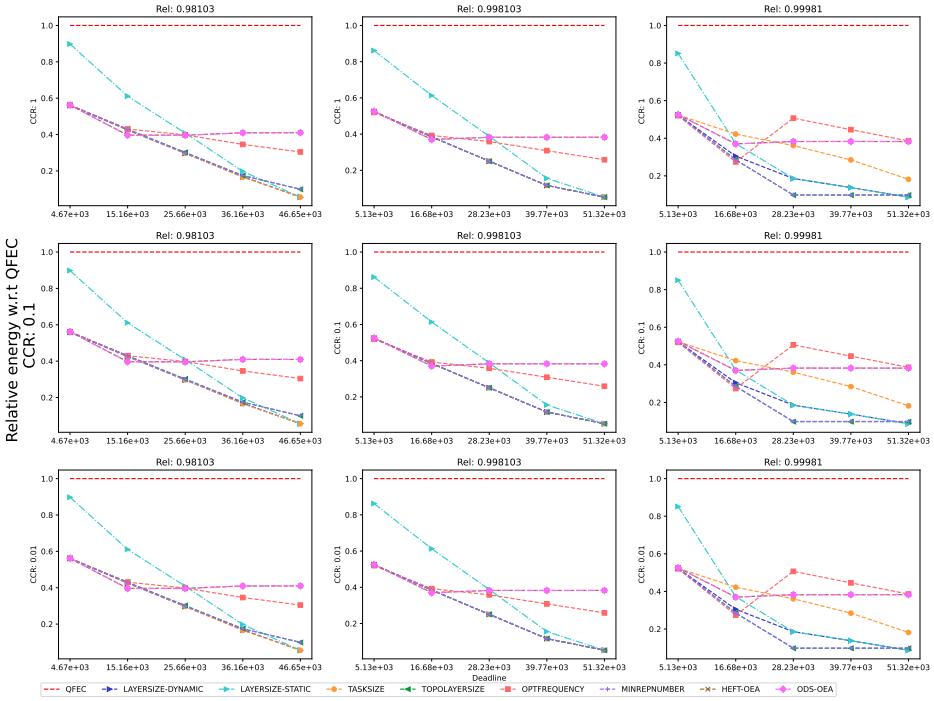

Figure 268: Performance of the different heuristics on the LU workflow.

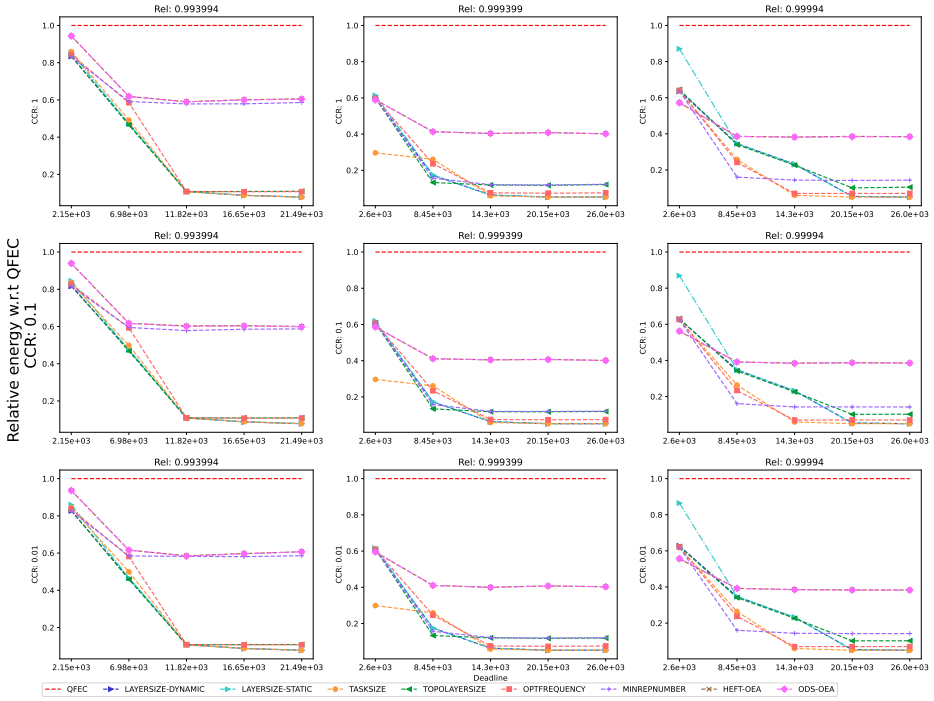

Figure 269: Performance of the different heuristics on the Montage workflow.

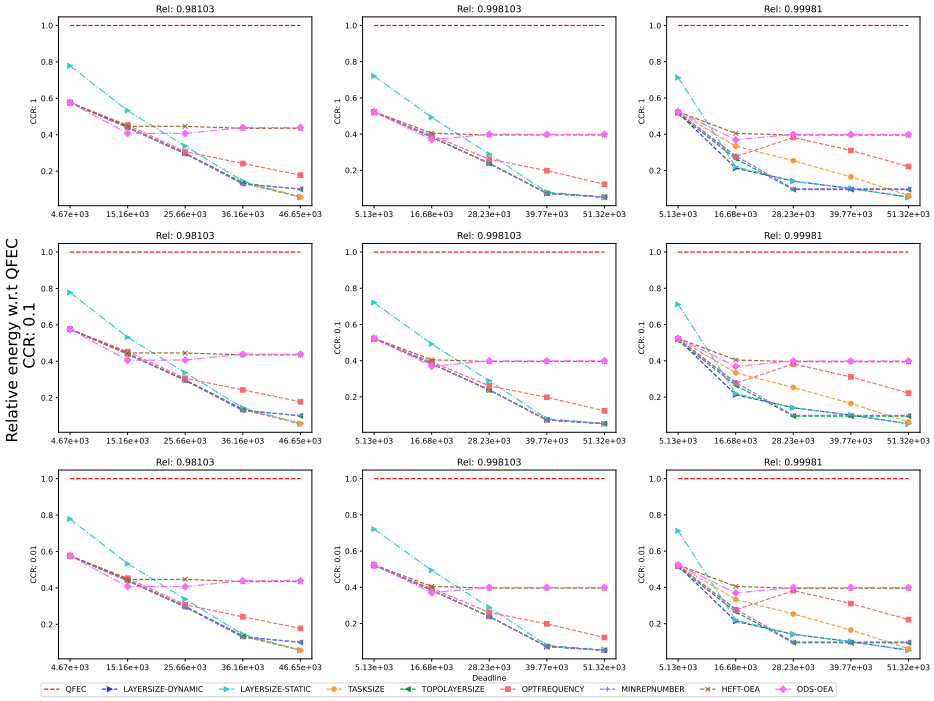

Figure 270: Performance of the different heuristics on the QR workflow.

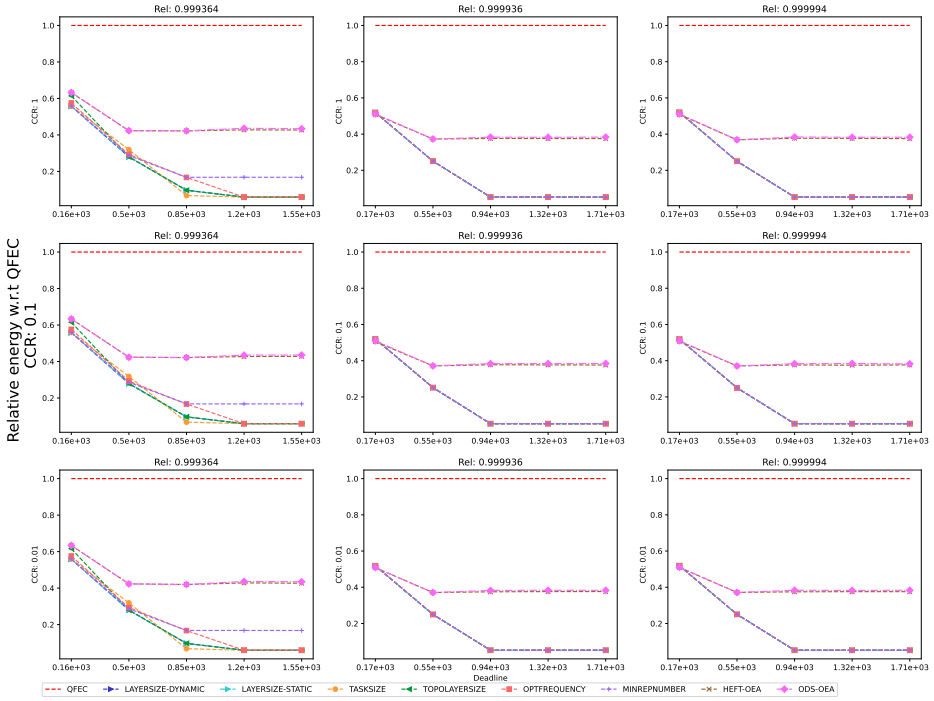

Figure 271: Performance of the different heuristics on the Seismology workflow.

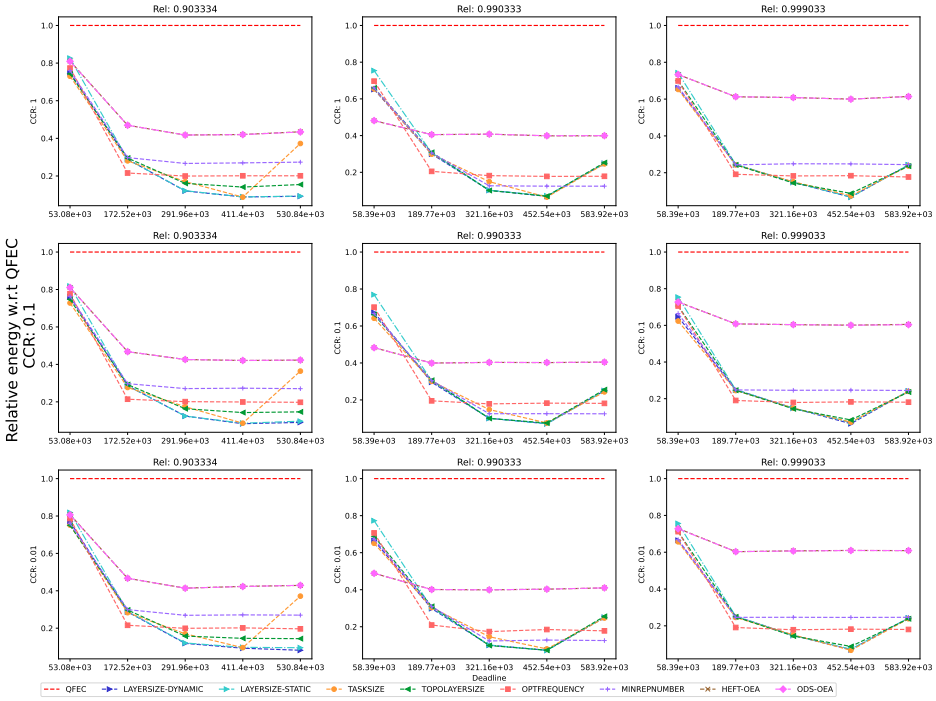

Figure 272: Performance of the different heuristics on the SoyKB workflow.

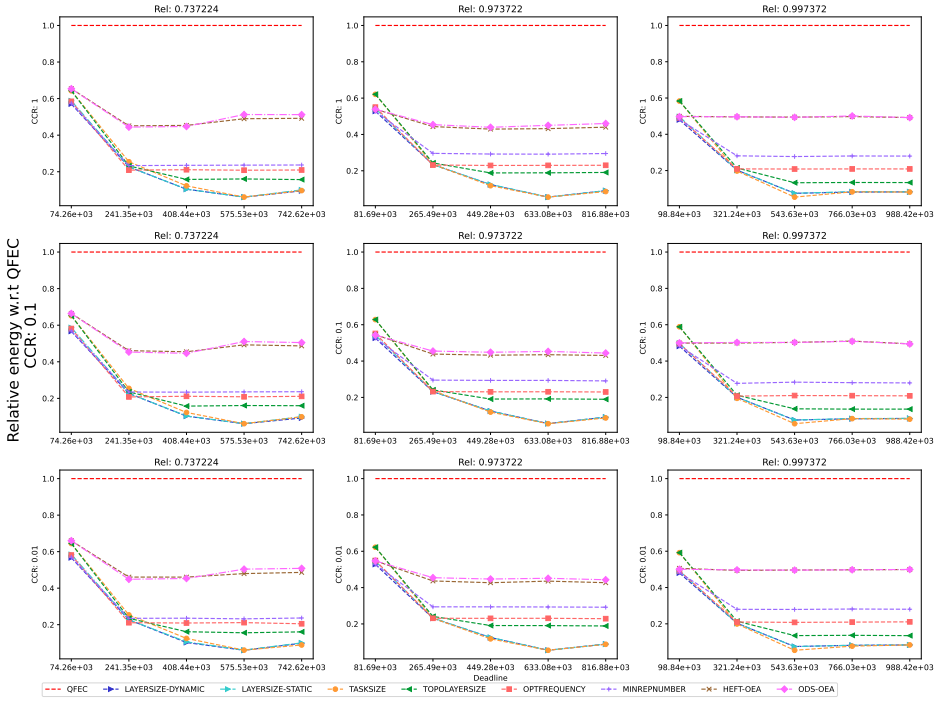

Figure 273: Performance of the different heuristics on the SRASearch.

D.6.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

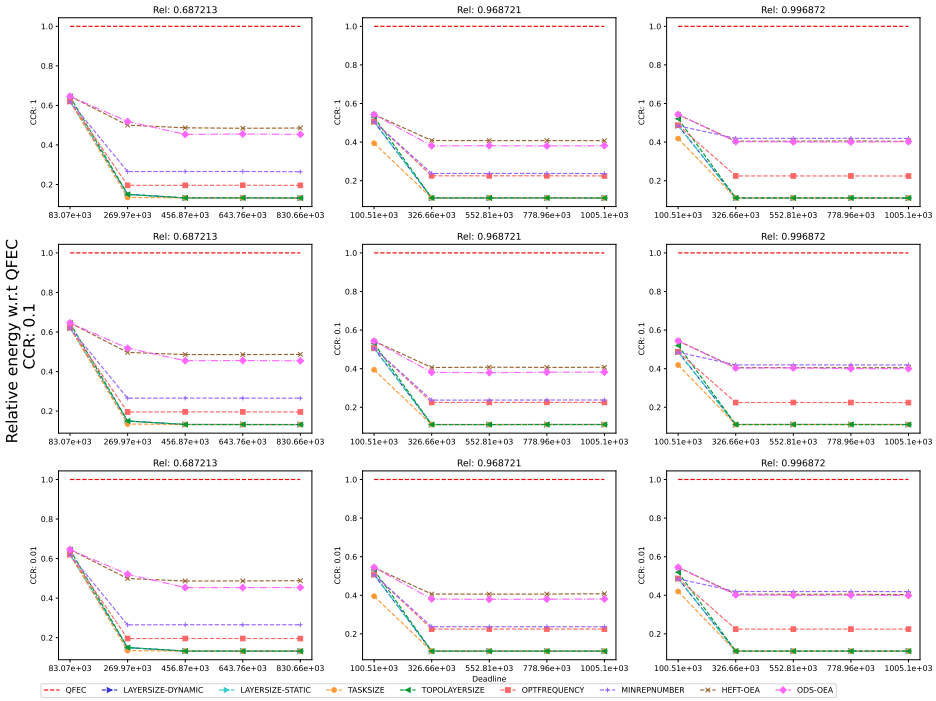

Figure 274: Performance of the different heuristics on the BLAST workflow.

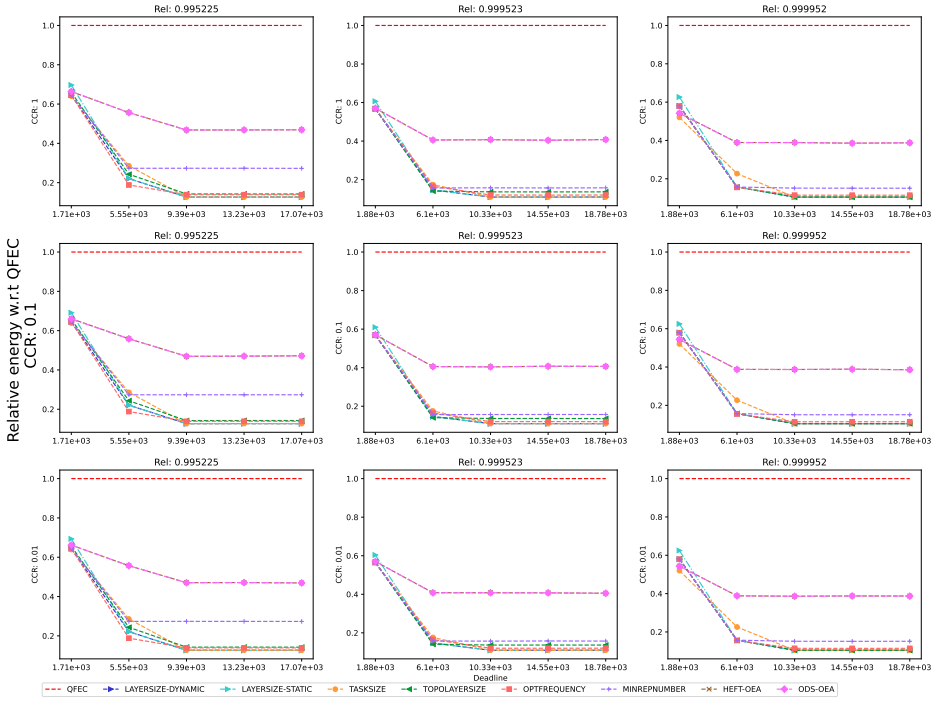

Figure 275: Performance of the different heuristics on the BWA workflow.

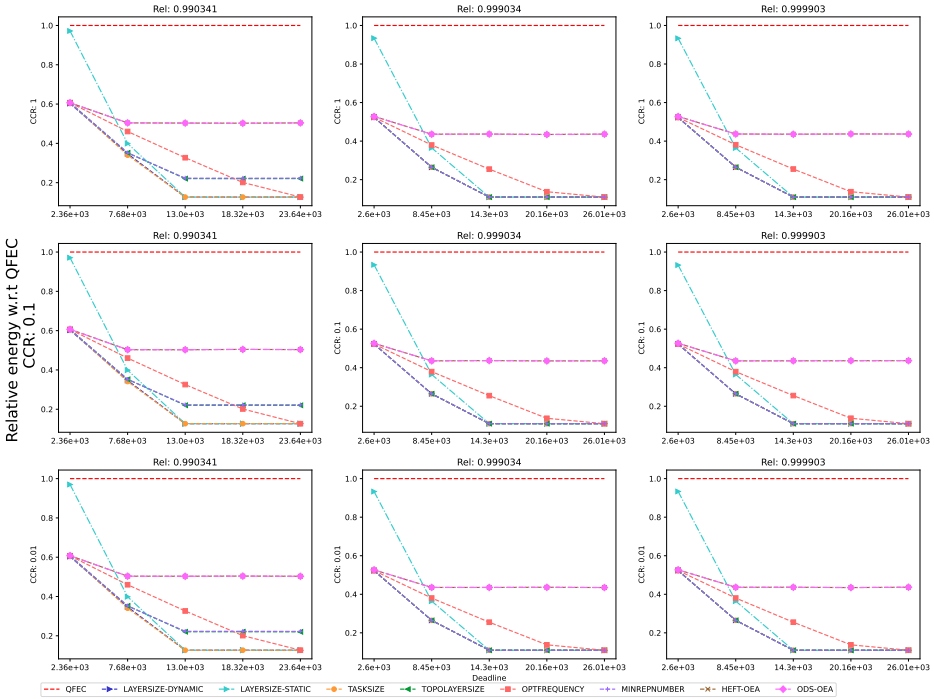

Figure 276: Performance of the different heuristics on the Cholesky workflow.

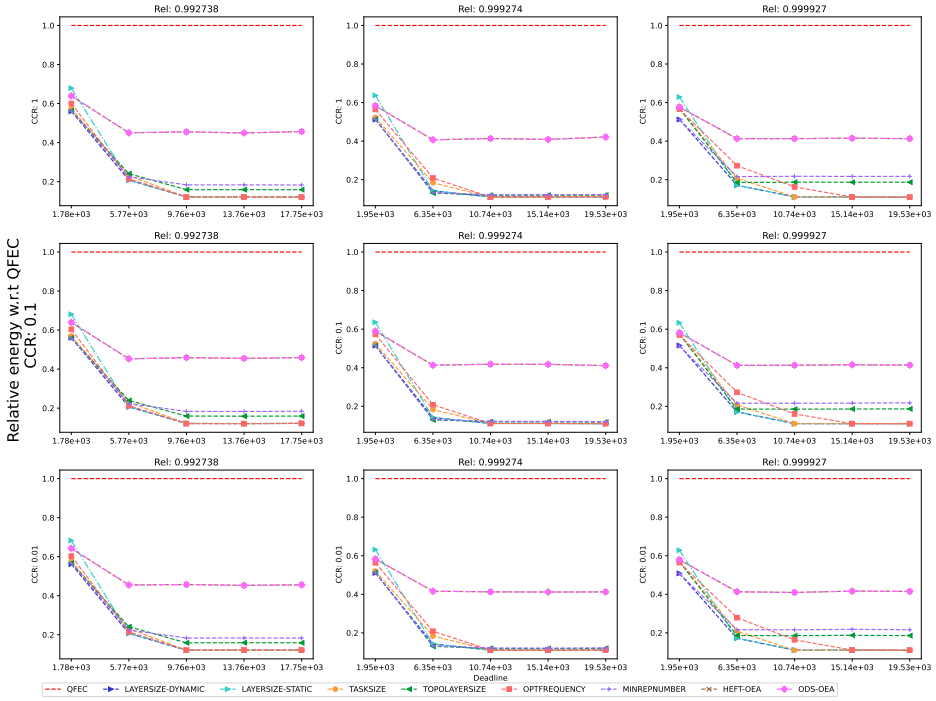

Figure 277: Performance of the different heuristics on the Cycles workflow.

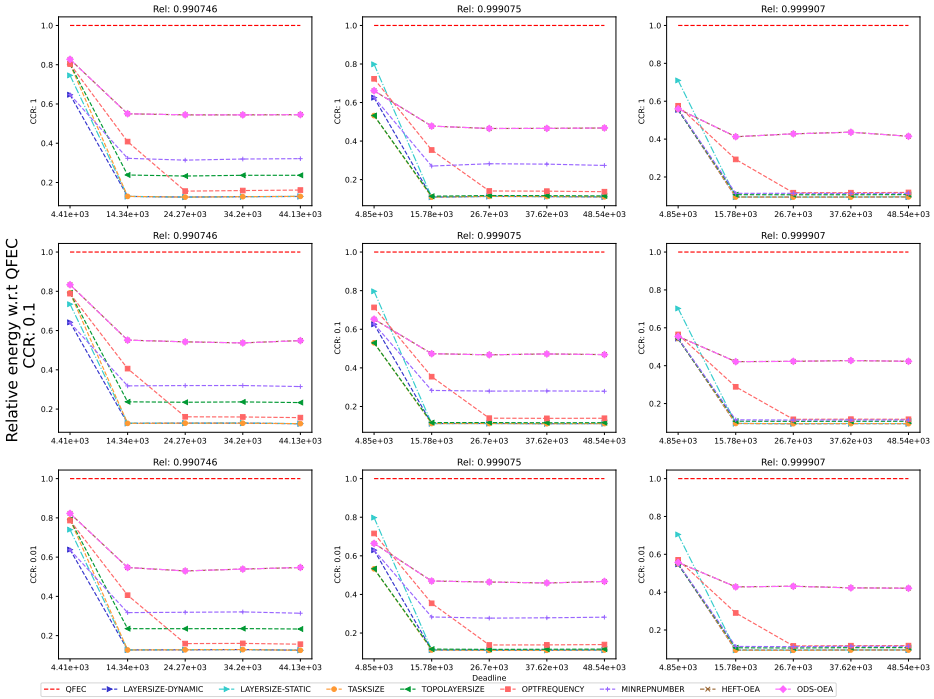

Figure 278: Performance of the different heuristics on the Epigenomics workflow.

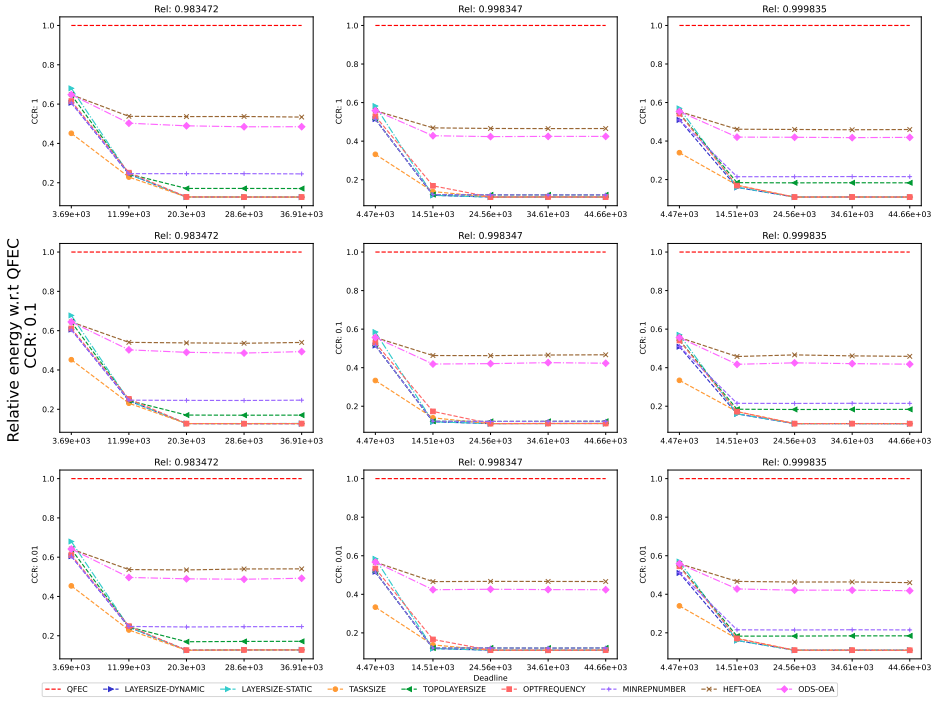

Figure 279: Performance of the different heuristics on the Genome workflow.

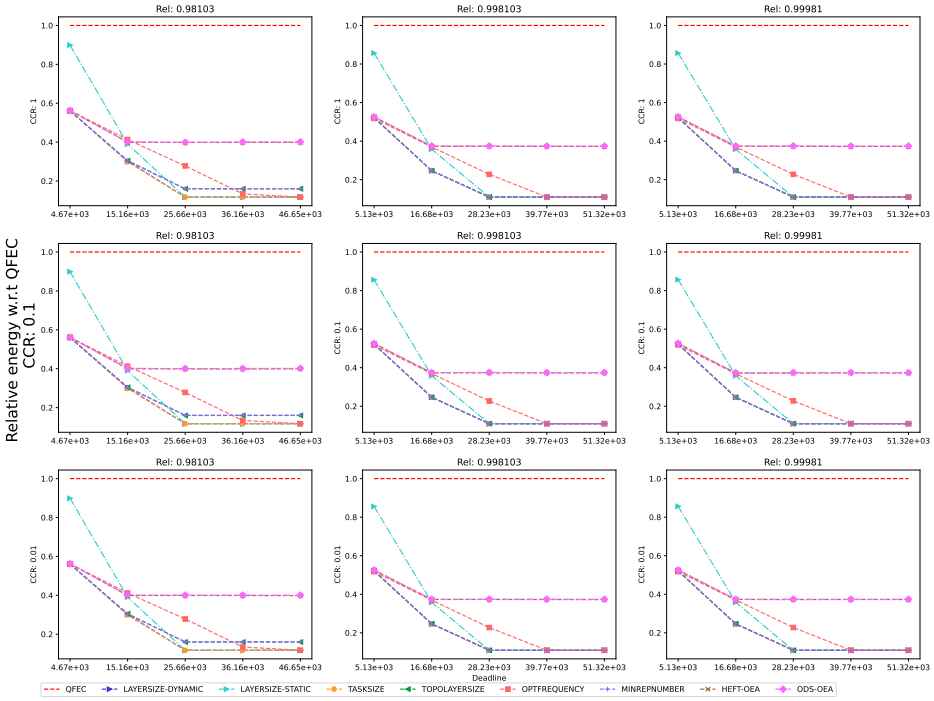

Figure 280: Performance of the different heuristics on the LU workflow.

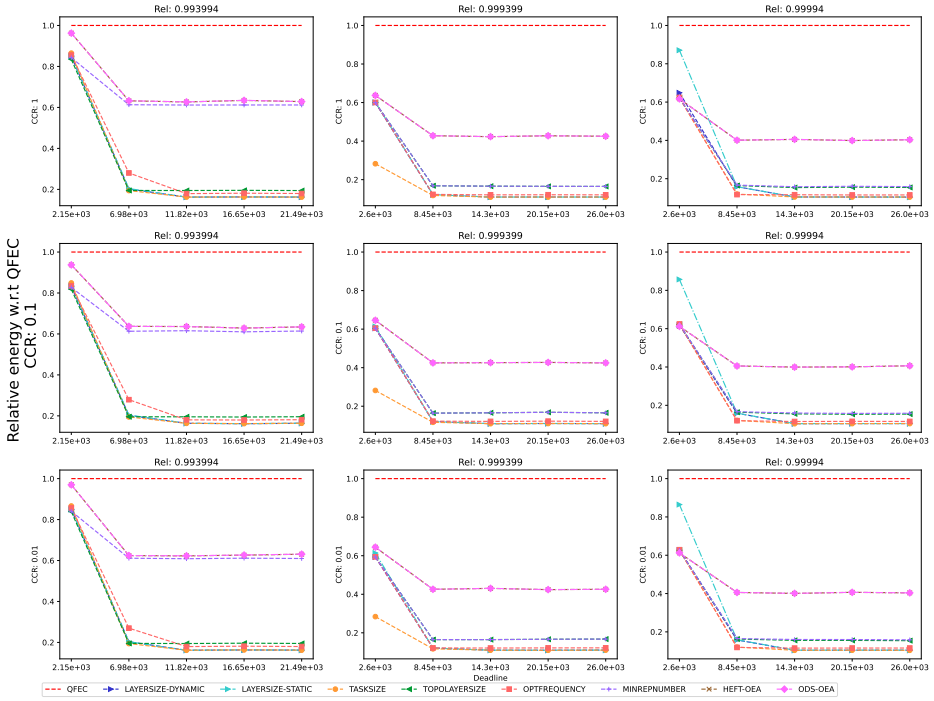

Figure 281: Performance of the different heuristics on the Montage workflow.

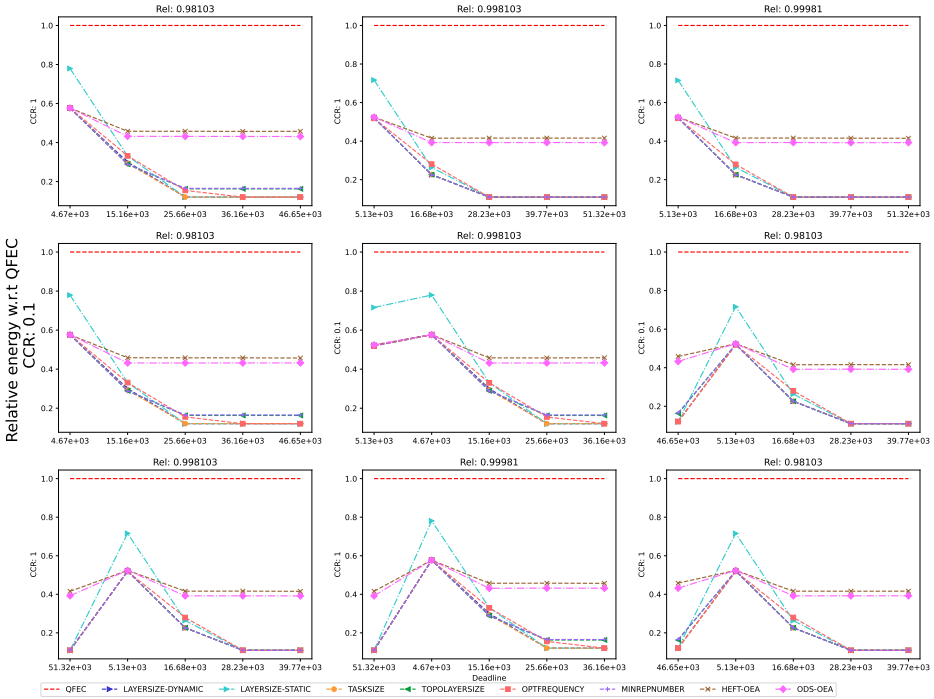

Figure 282: Performance of the different heuristics on the QR workflow.

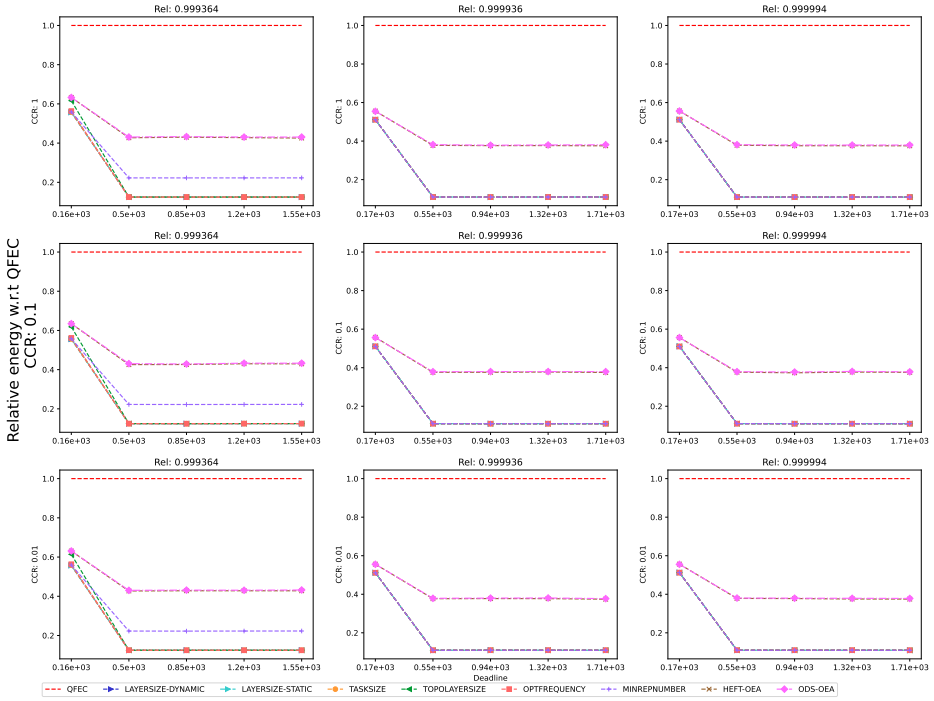

Figure 283: Performance of the different heuristics on the Seismology workflow.

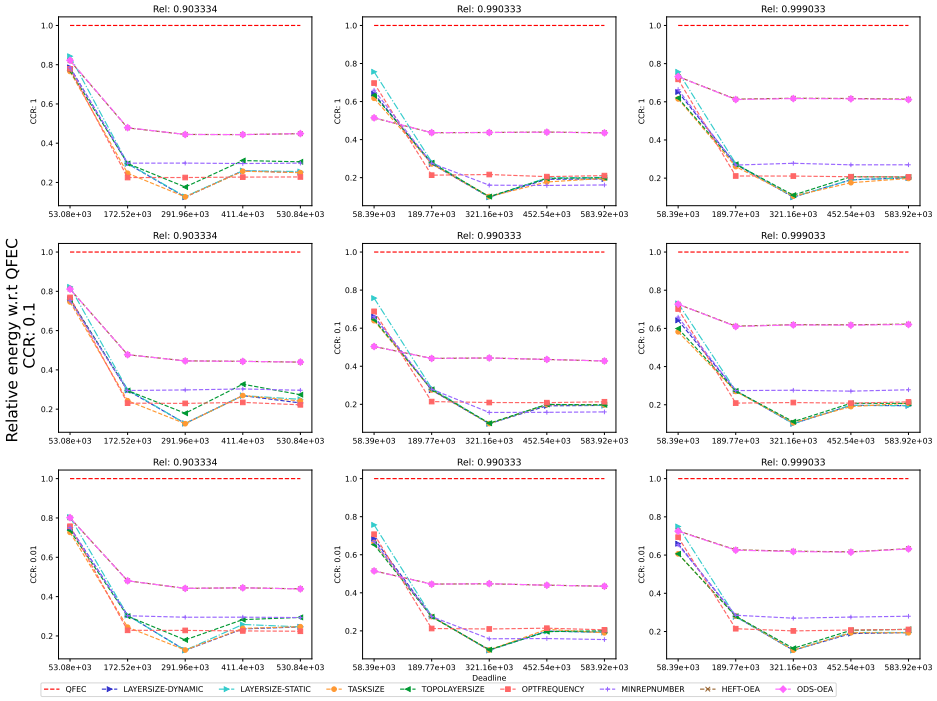

Figure 284: Performance of the different heuristics on the SoyKB workflow.

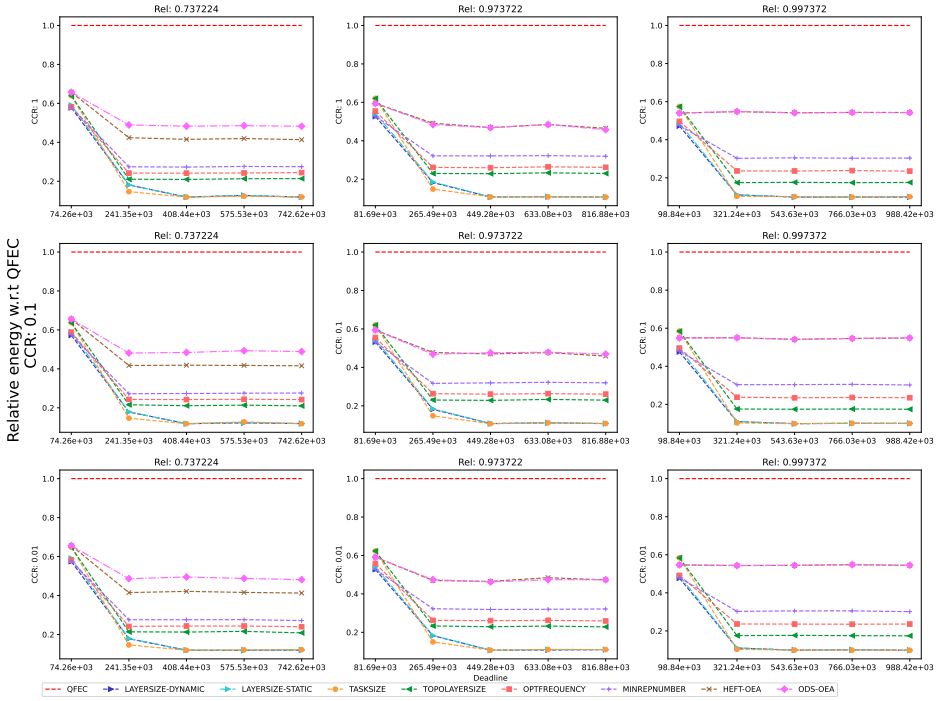

Figure 285: Performance of the different heuristics on the SRASearch.

D.6.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

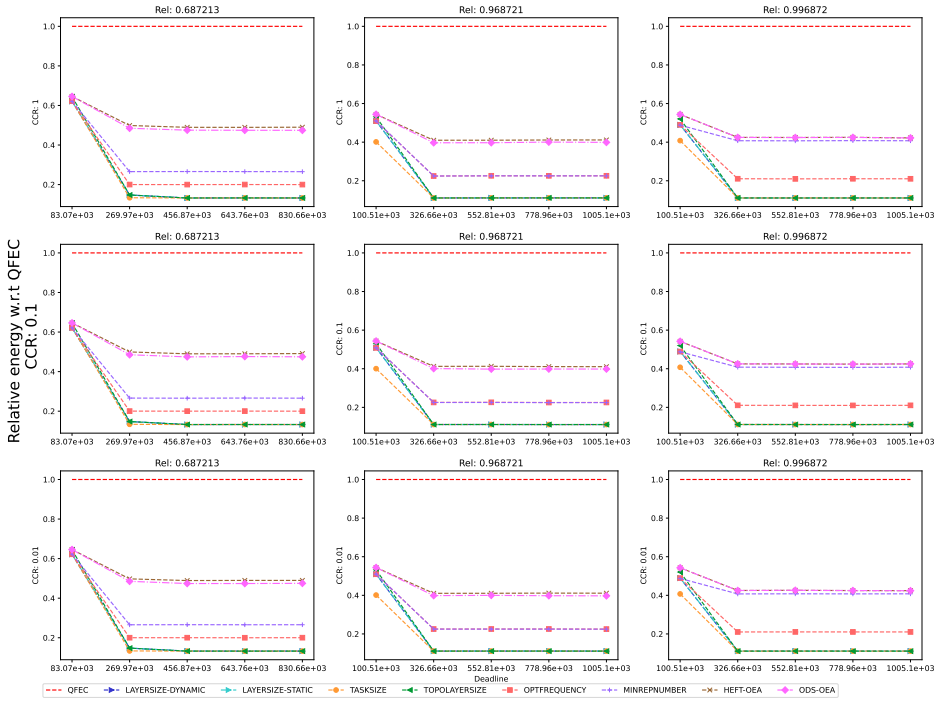

Figure 286: Performance of the different heuristics on the BLAST workflow.

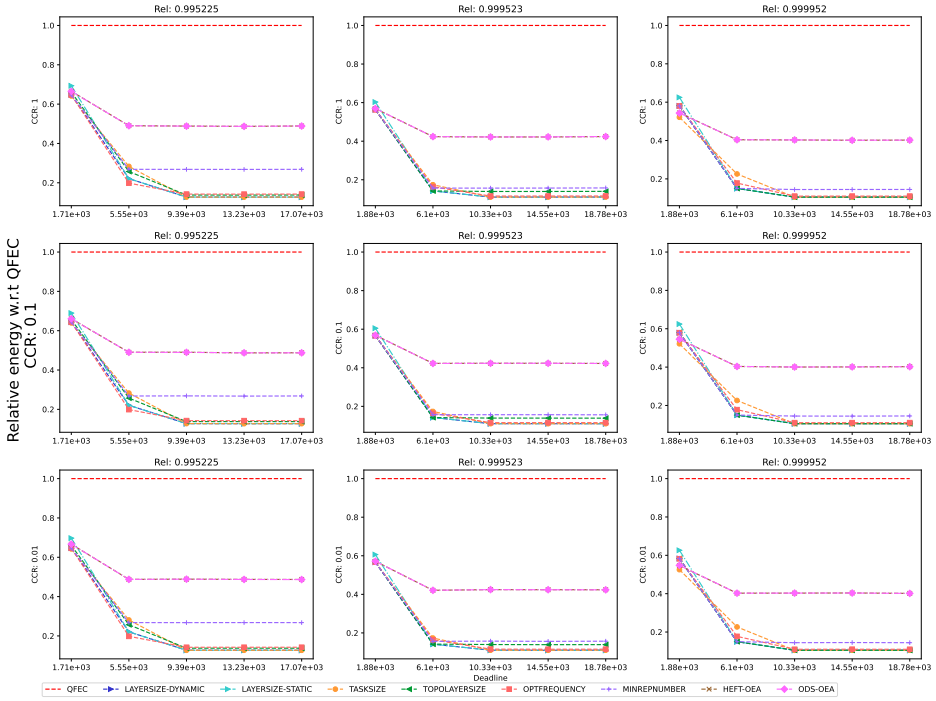

Figure 287: Performance of the different heuristics on the BWA workflow.

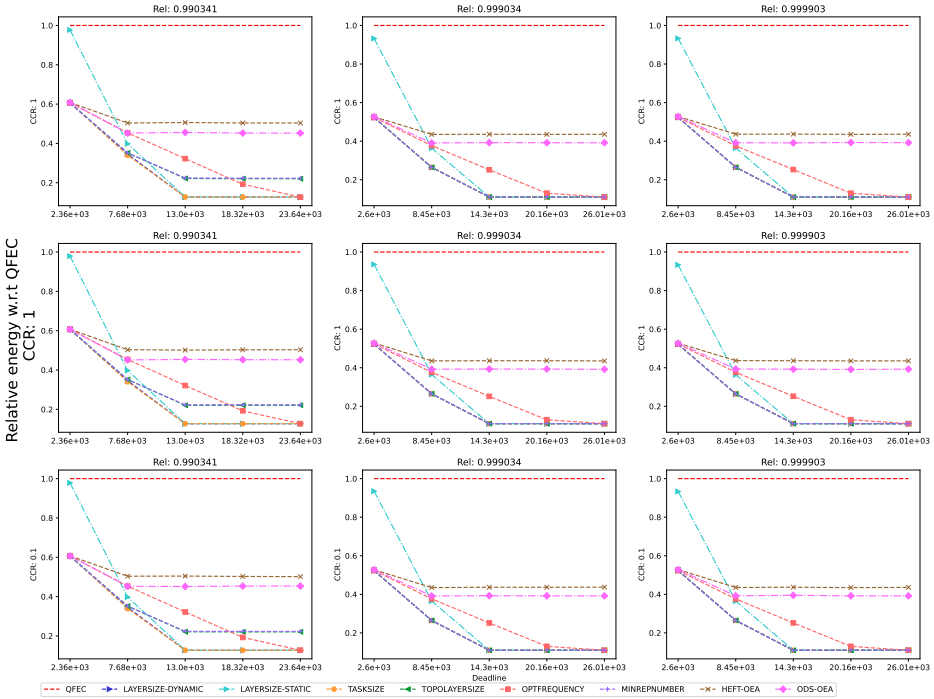

Figure 288: Performance of the different heuristics on the Cholesky workflow.

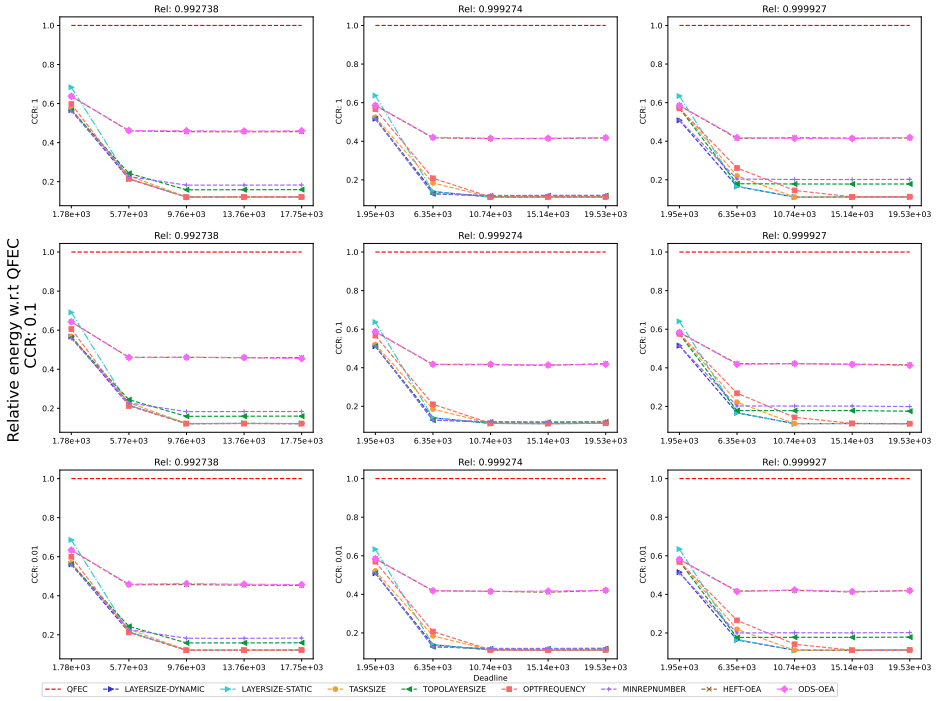

Figure 289: Performance of the different heuristics on the Cycles workflow.

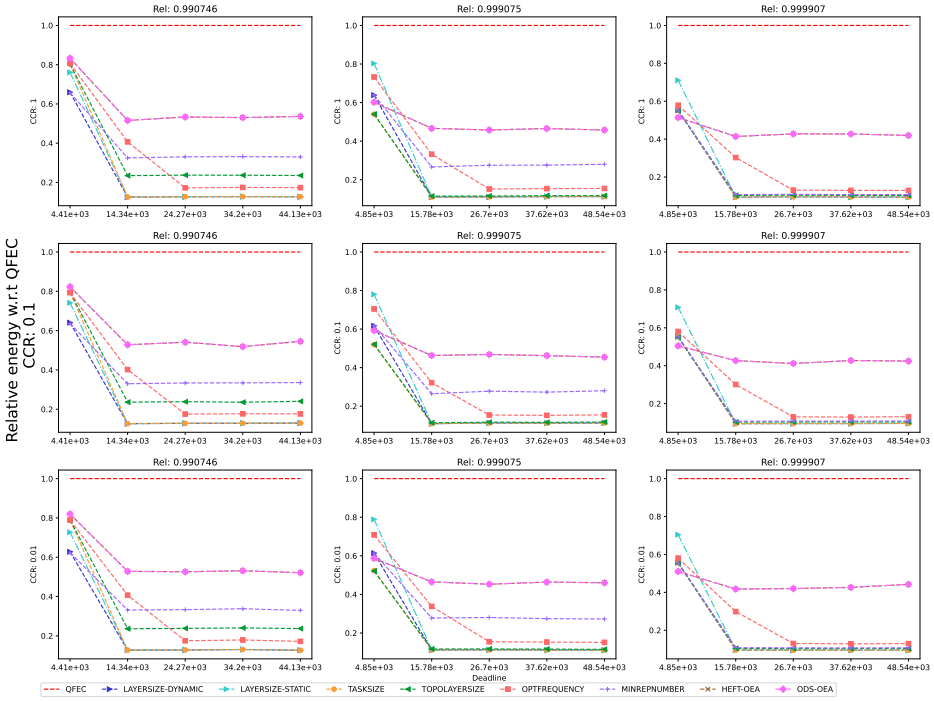

Figure 290: Performance of the different heuristics on the Epigenomics workflow.

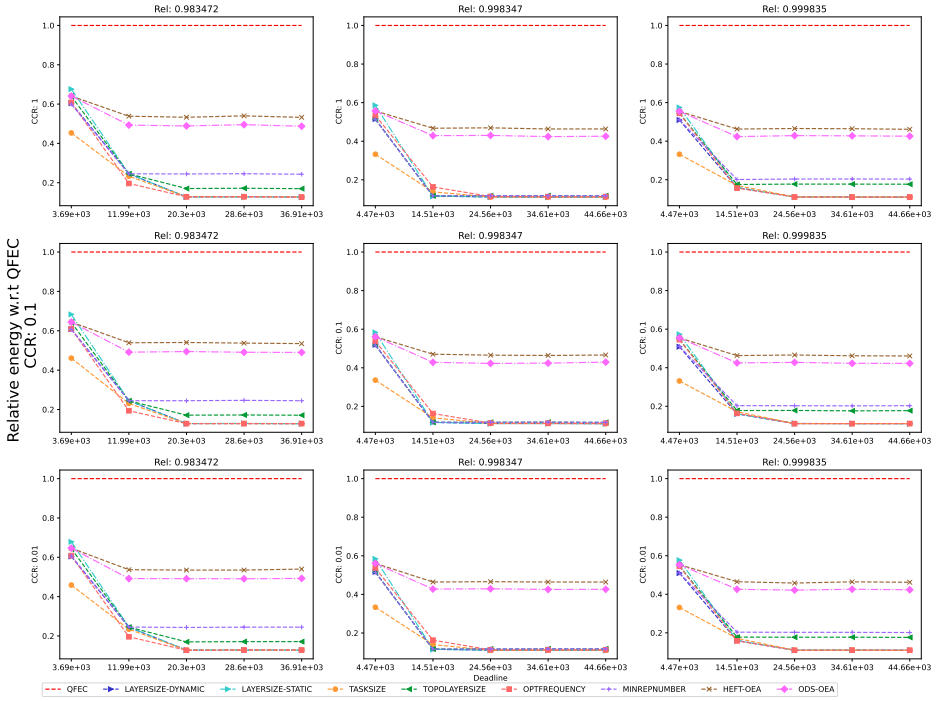

Figure 291: Performance of the different heuristics on the Genome workflow.

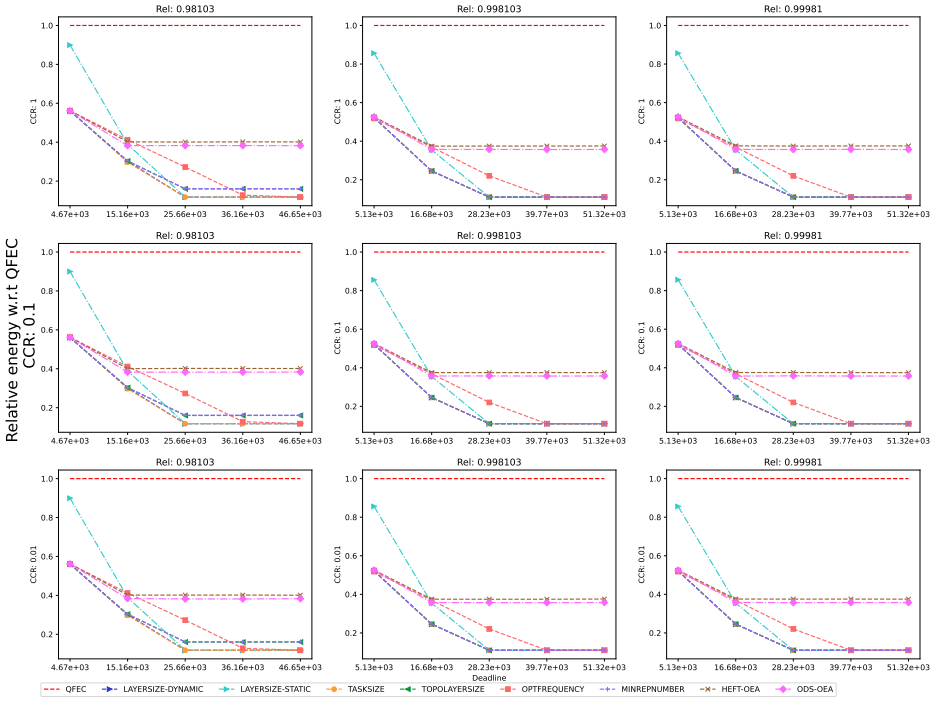

Figure 292: Performance of the different heuristics on the LU workflow.

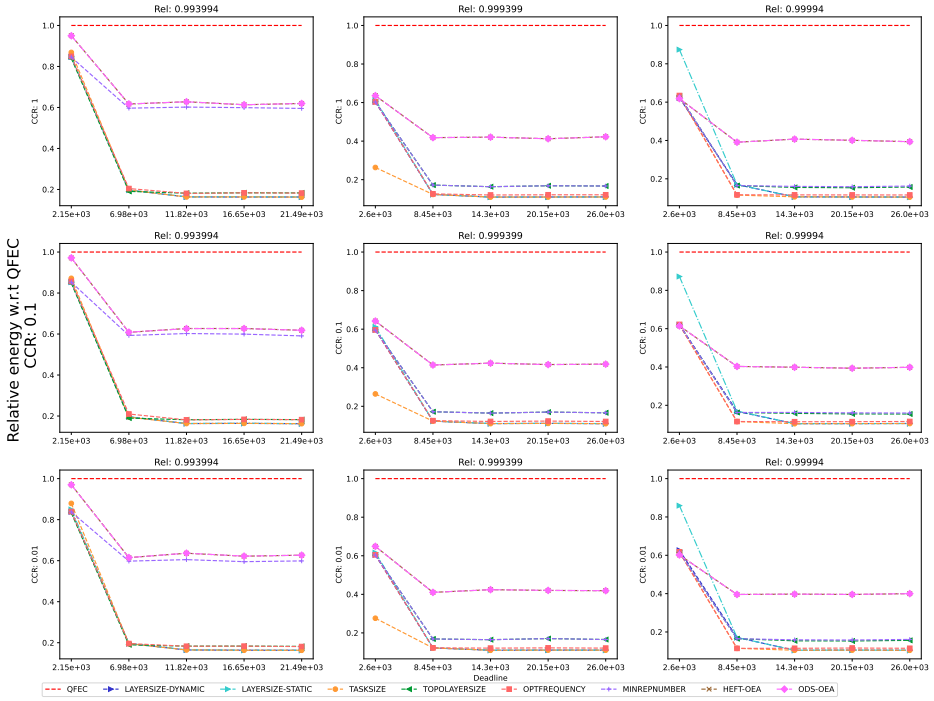

Figure 293: Performance of the different heuristics on the Montage workflow.

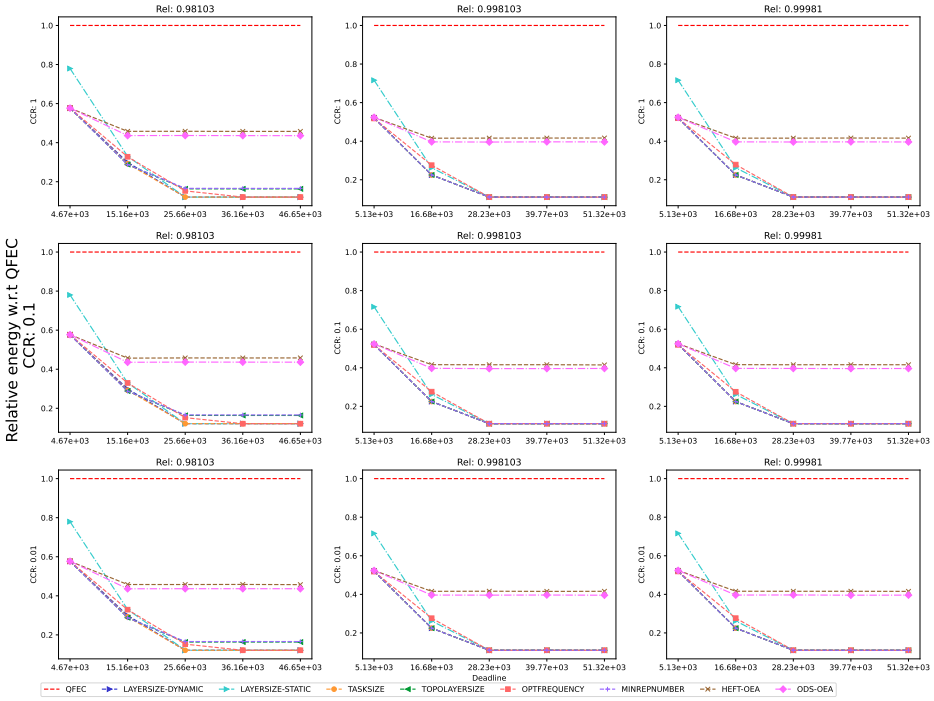

Figure 294: Performance of the different heuristics on the QR workflow.

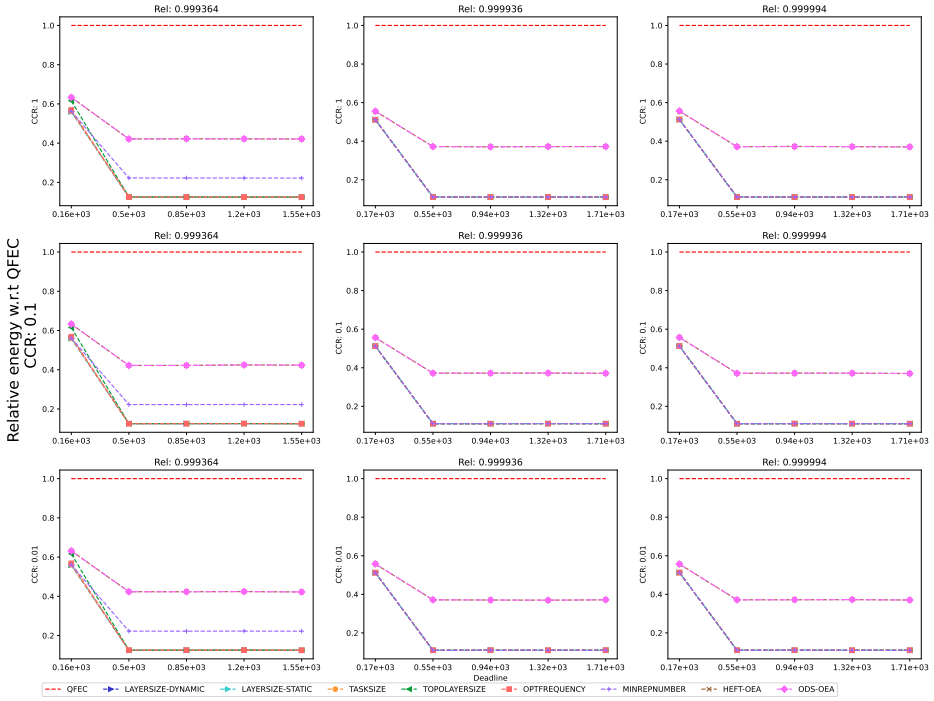

Figure 295: Performance of the different heuristics on the Seismology workflow.

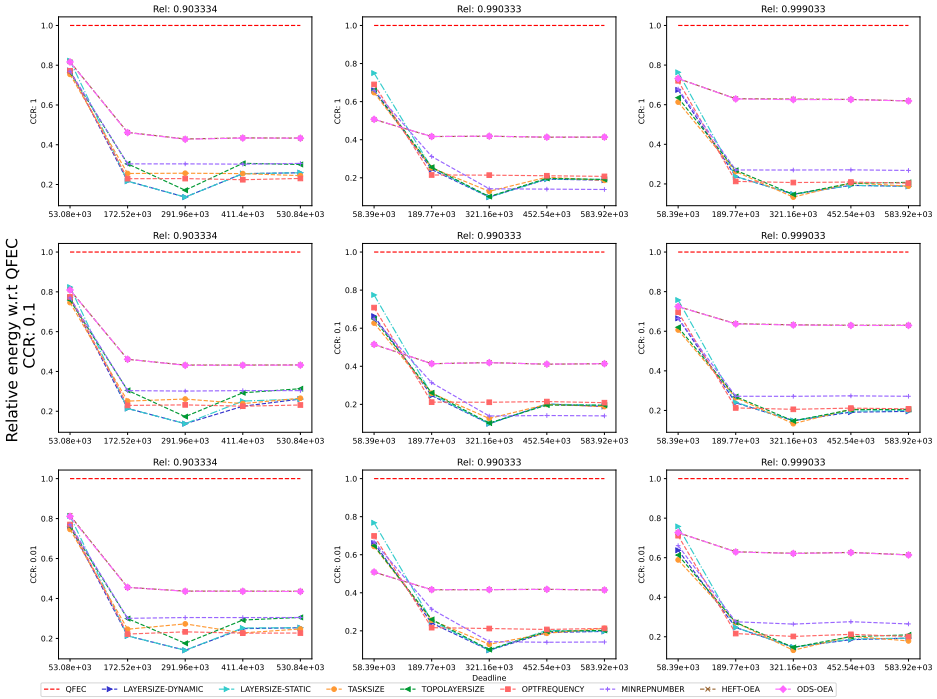

Figure 296: Performance of the different heuristics on the SoyKB workflow.

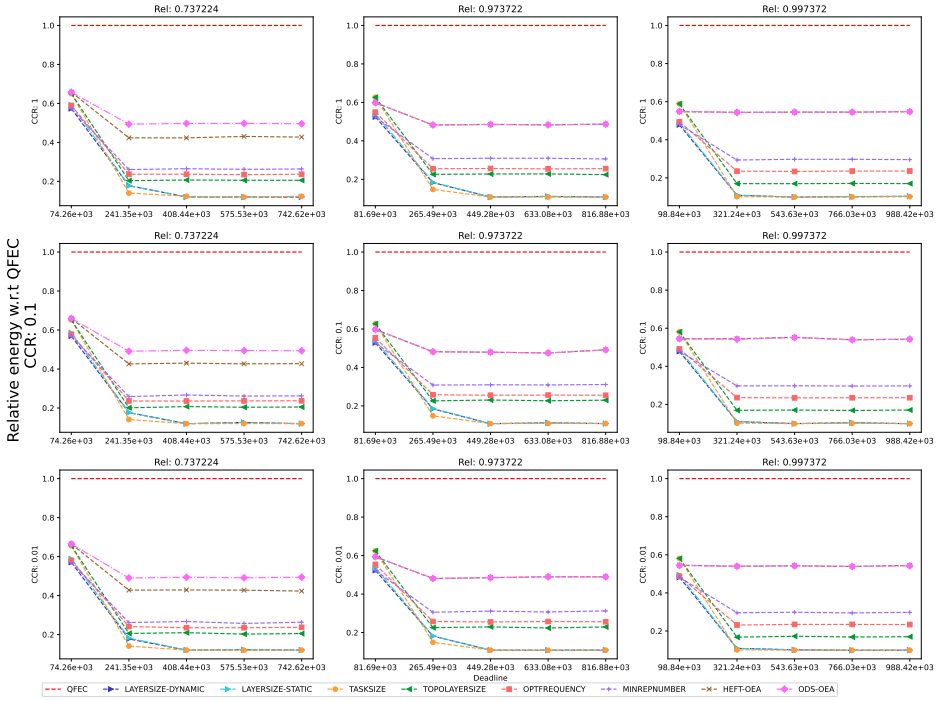

Figure 297: Performance of the different heuristics on the SRASearch.

D.7 $BC/WC = 0.7$

D.7.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

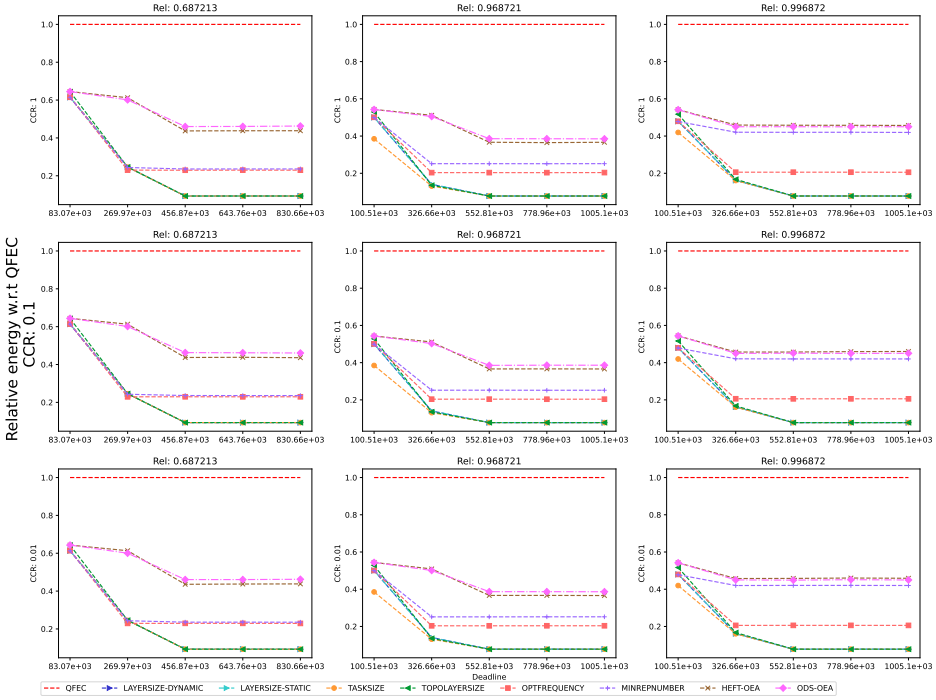

Figure 298: Performance of the different heuristics on the BLAST workflow.

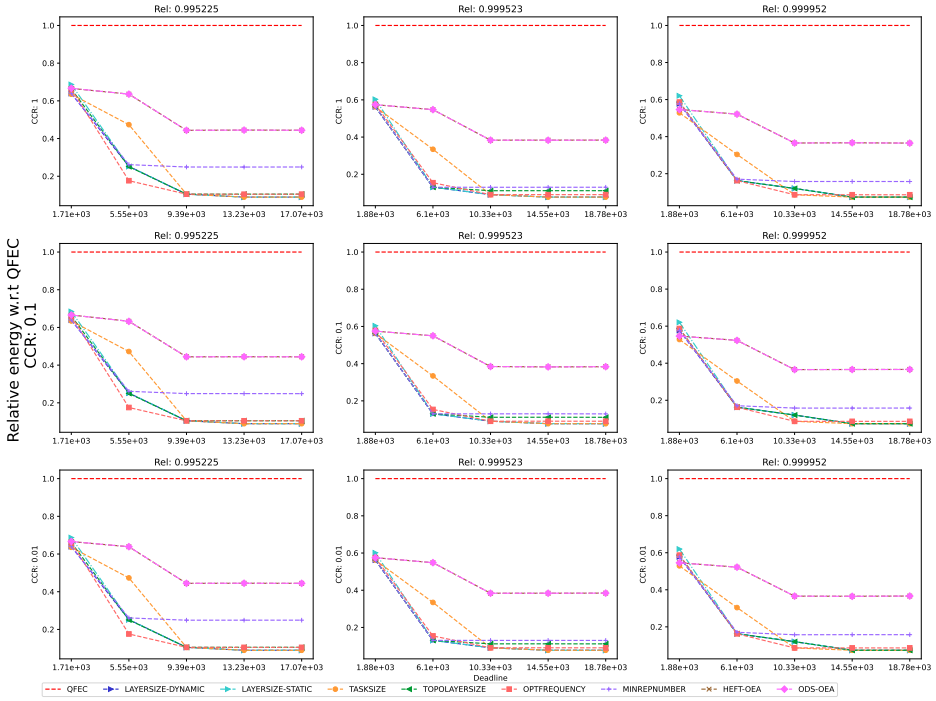

Figure 299: Performance of the different heuristics on the BWA workflow.

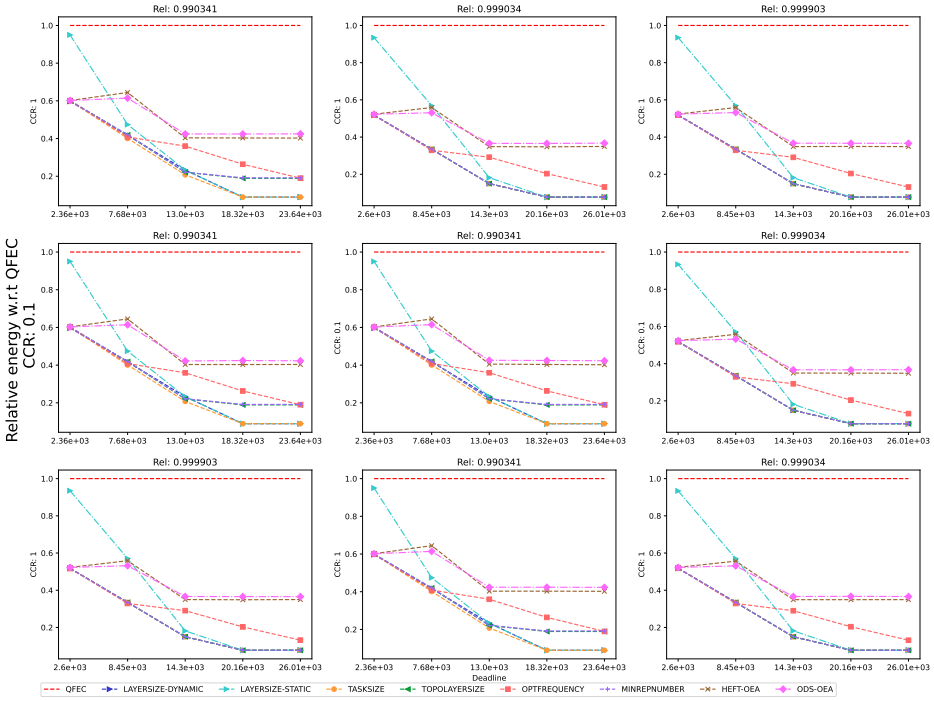

Figure 300: Performance of the different heuristics on the Cholesky workflow.

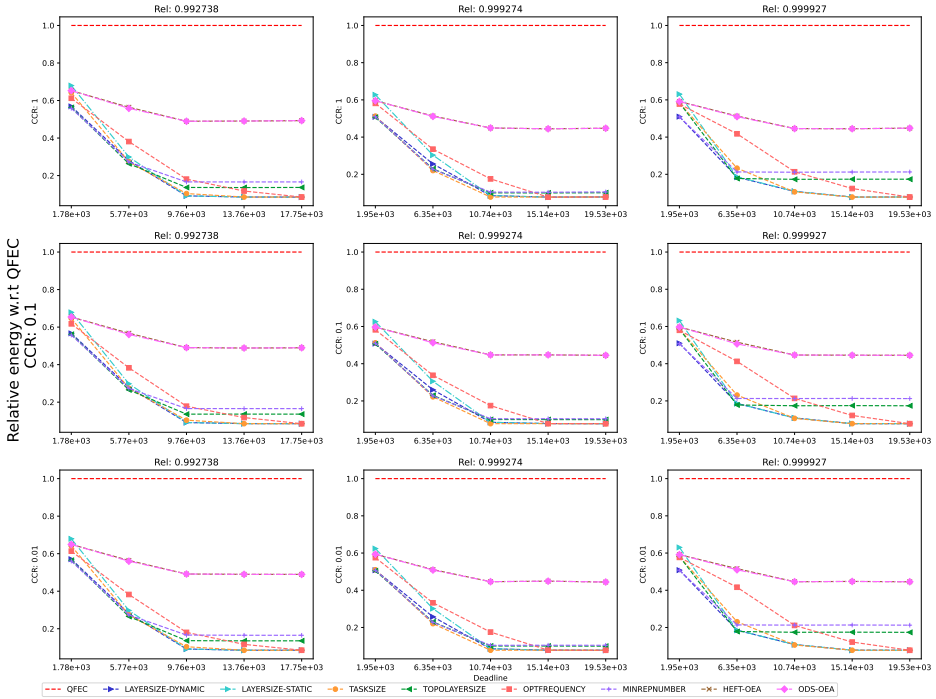

Figure 301: Performance of the different heuristics on the Cycles workflow.

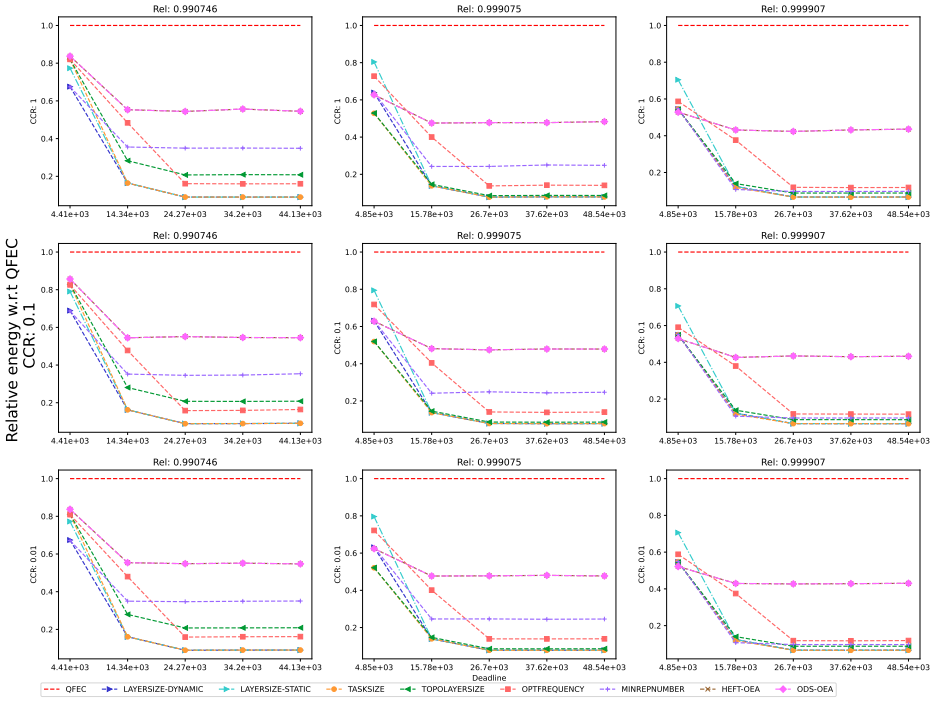

Figure 302: Performance of the different heuristics on the Epigenomics workflow.

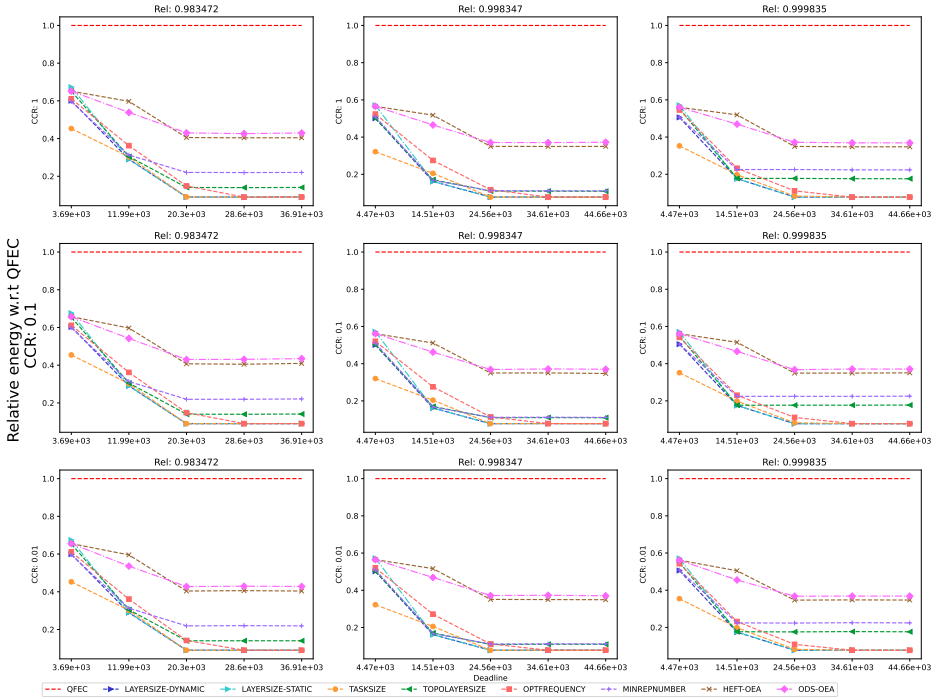

Figure 303: Performance of the different heuristics on the Genome workflow.

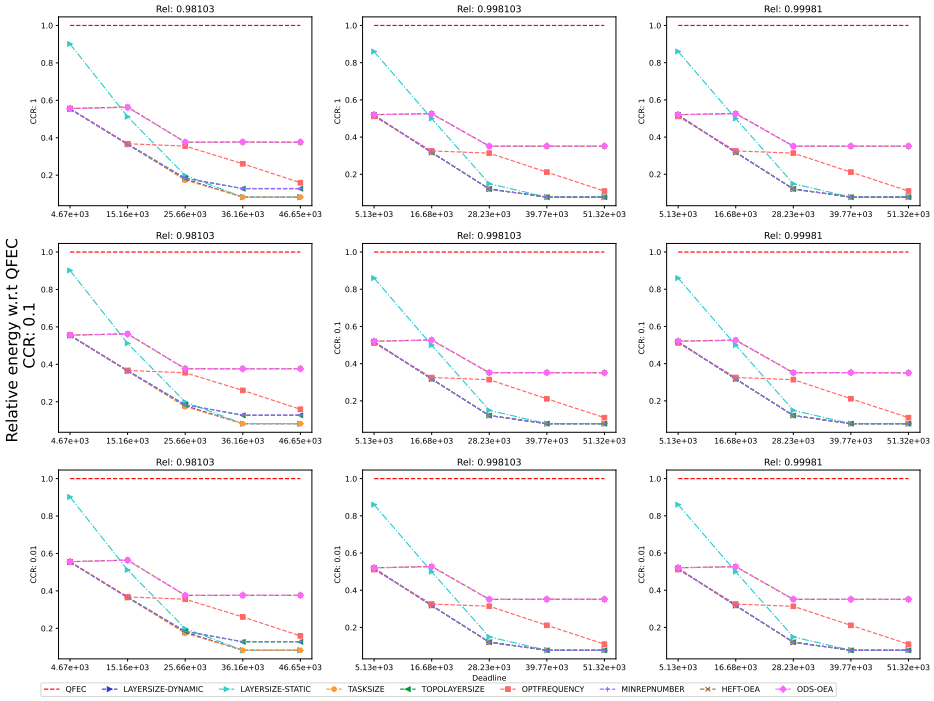

Figure 304: Performance of the different heuristics on the LU workflow.

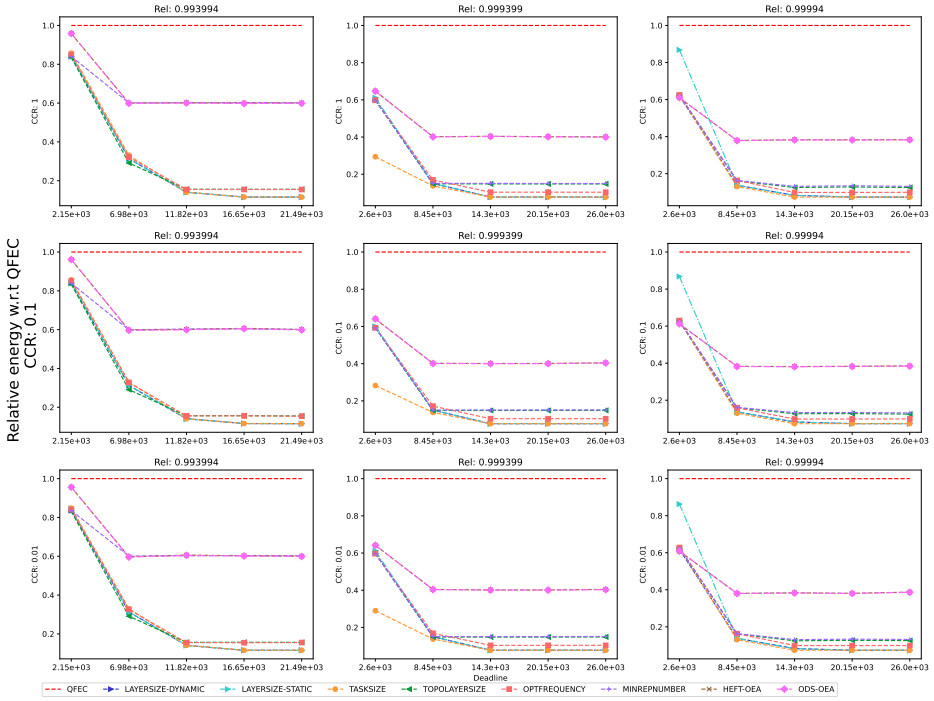

Figure 305: Performance of the different heuristics on the Montage workflow.

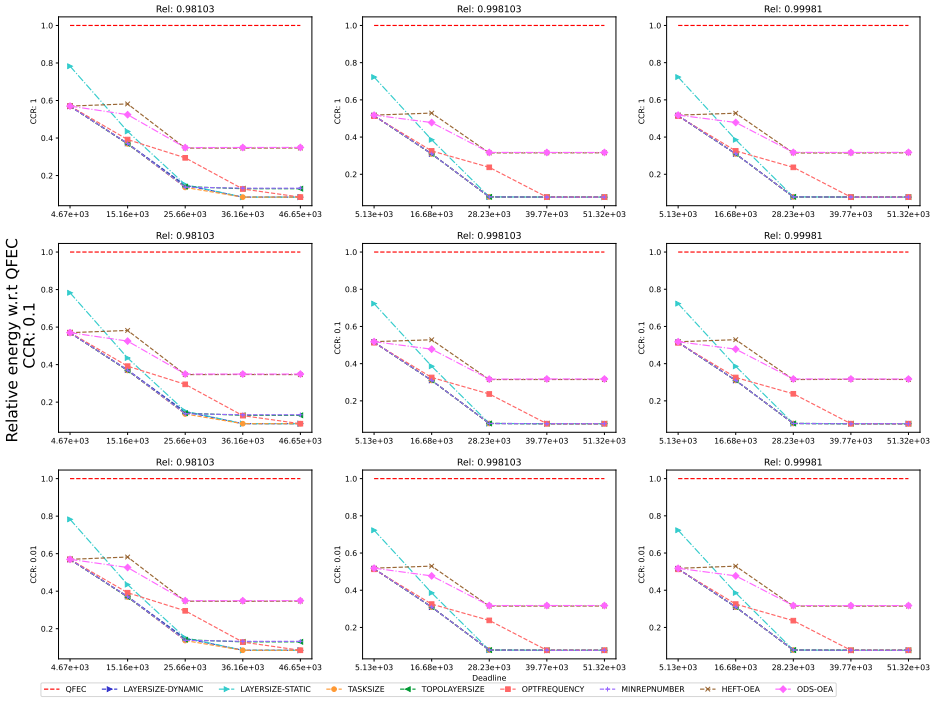

Figure 306: Performance of the different heuristics on the QR workflow.

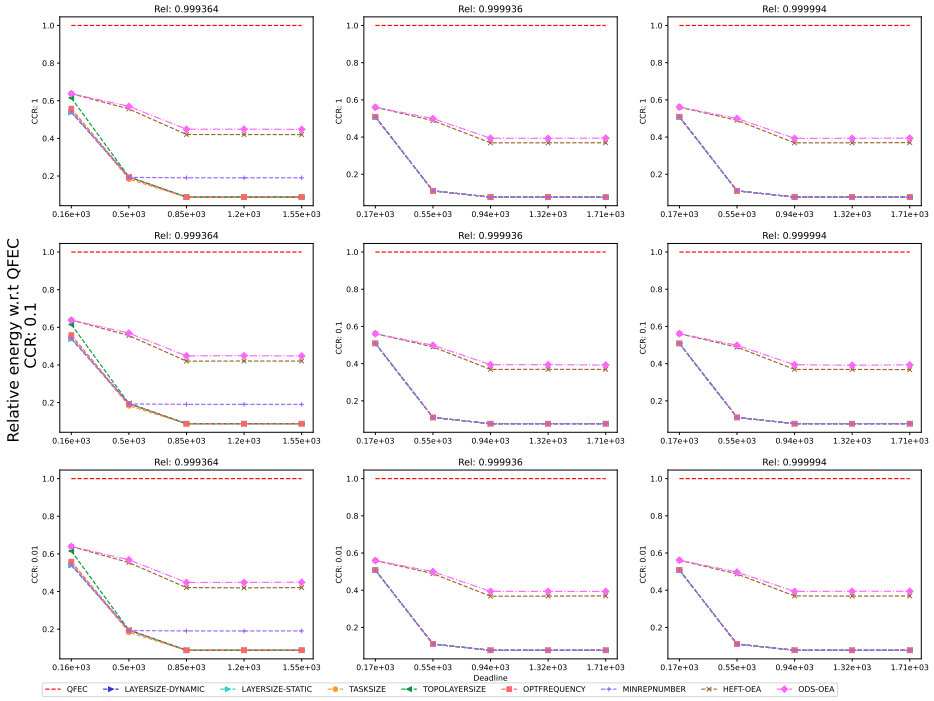

Figure 307: Performance of the different heuristics on the Seismology workflow.

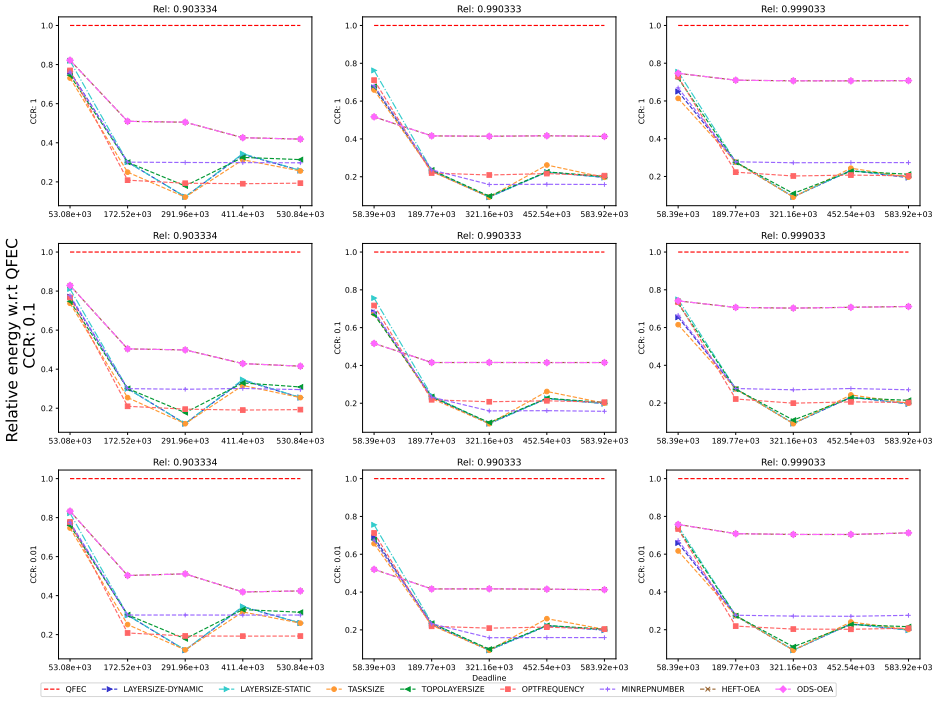

Figure 308: Performance of the different heuristics on the SoyKB workflow.

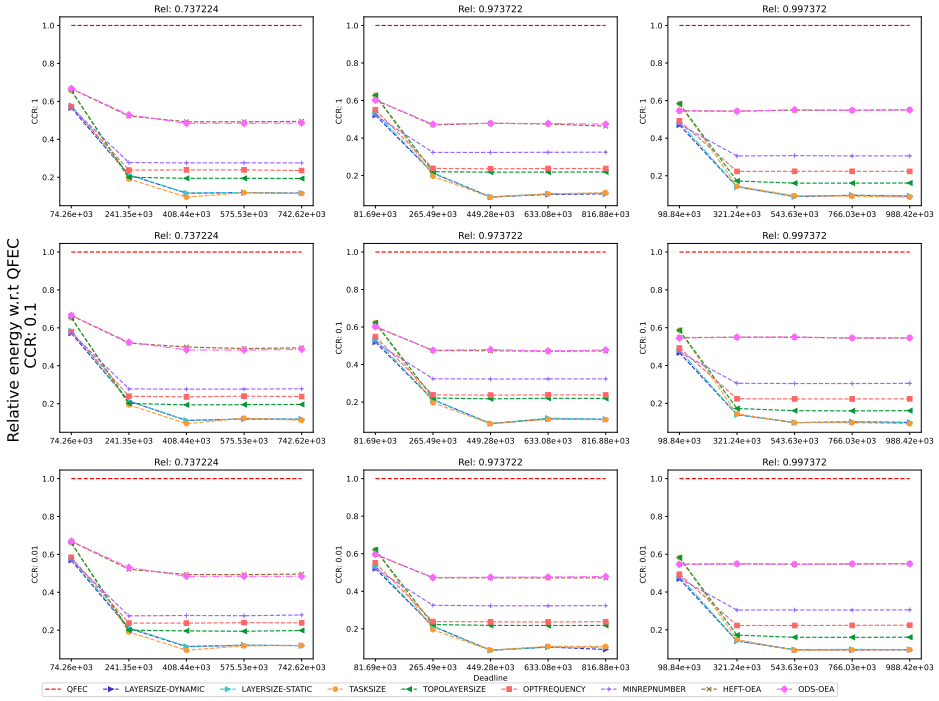

Figure 309: Performance of the different heuristics on the SRASearch.

D.7.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

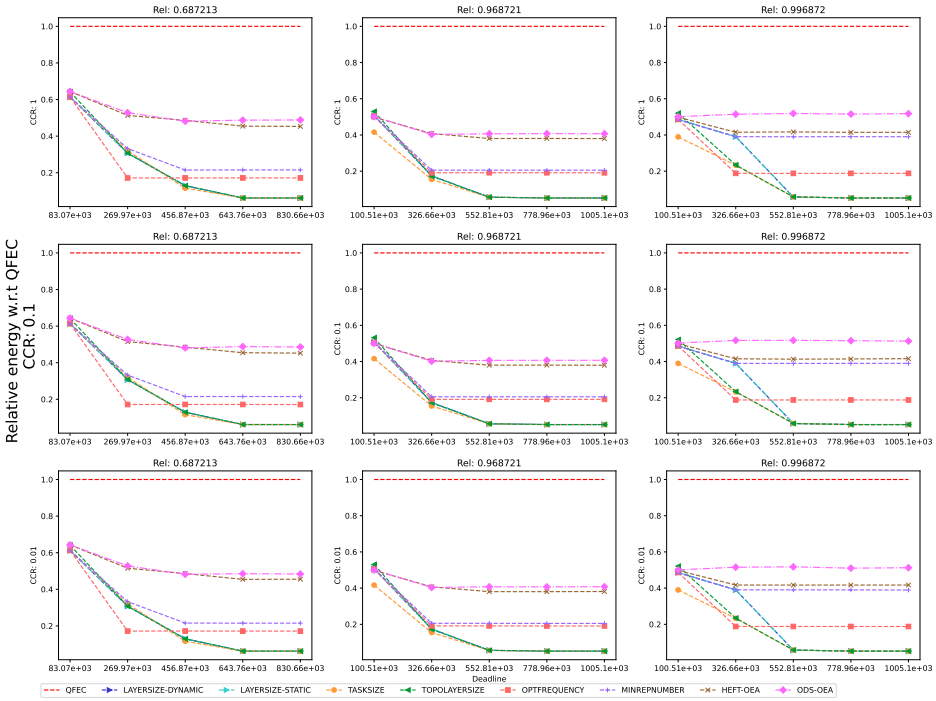

Figure 310: Performance of the different heuristics on the BLAST workflow.

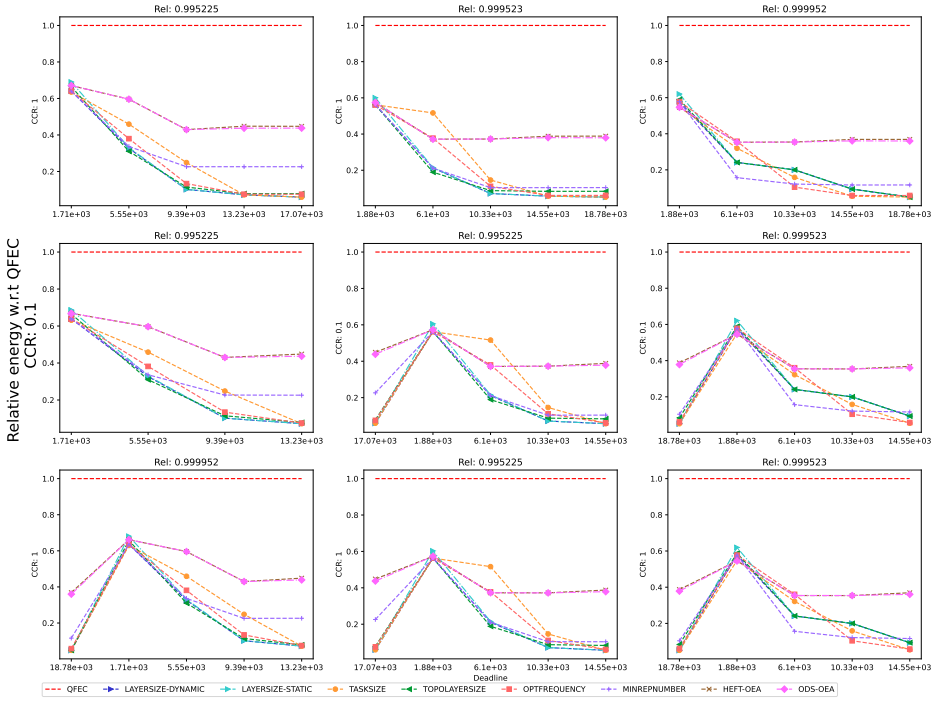

Figure 311: Performance of the different heuristics on the BWA workflow.

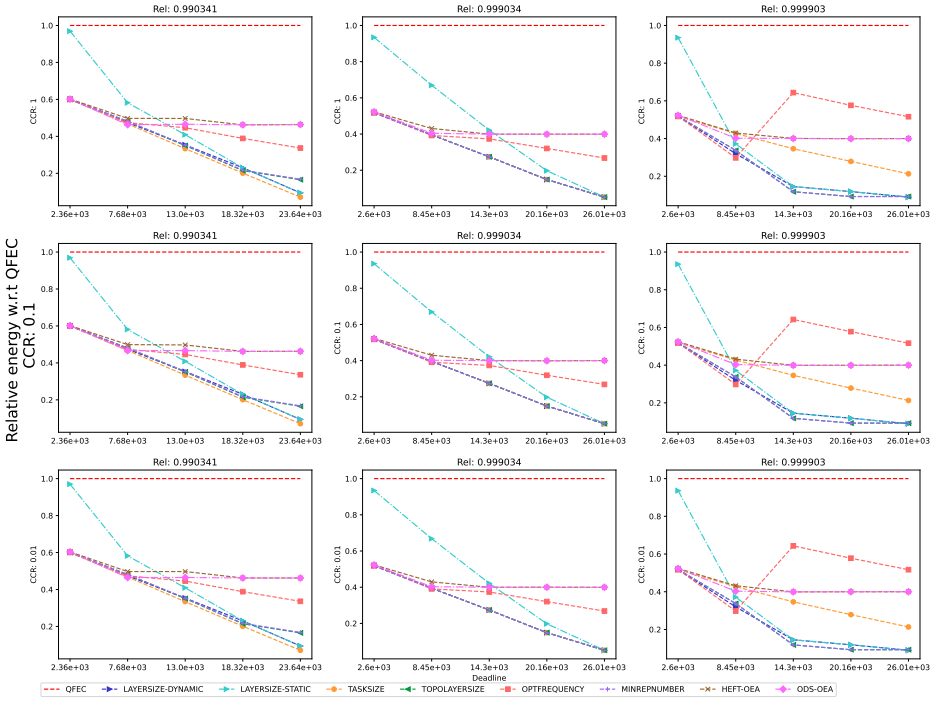

Figure 312: Performance of the different heuristics on the Cholesky workflow.

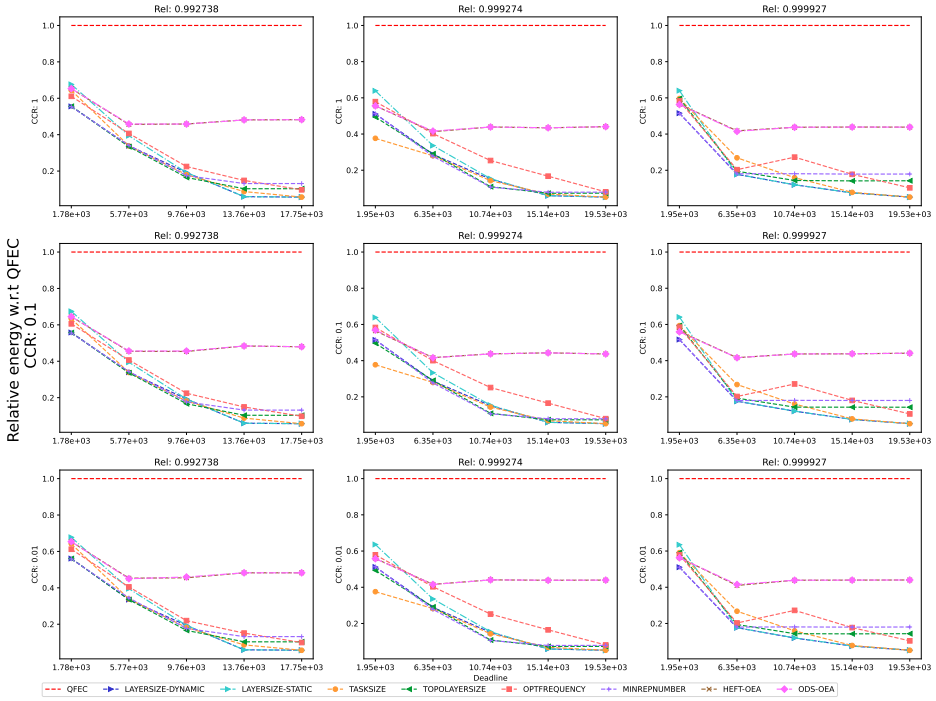

Figure 313: Performance of the different heuristics on the Cycles workflow.

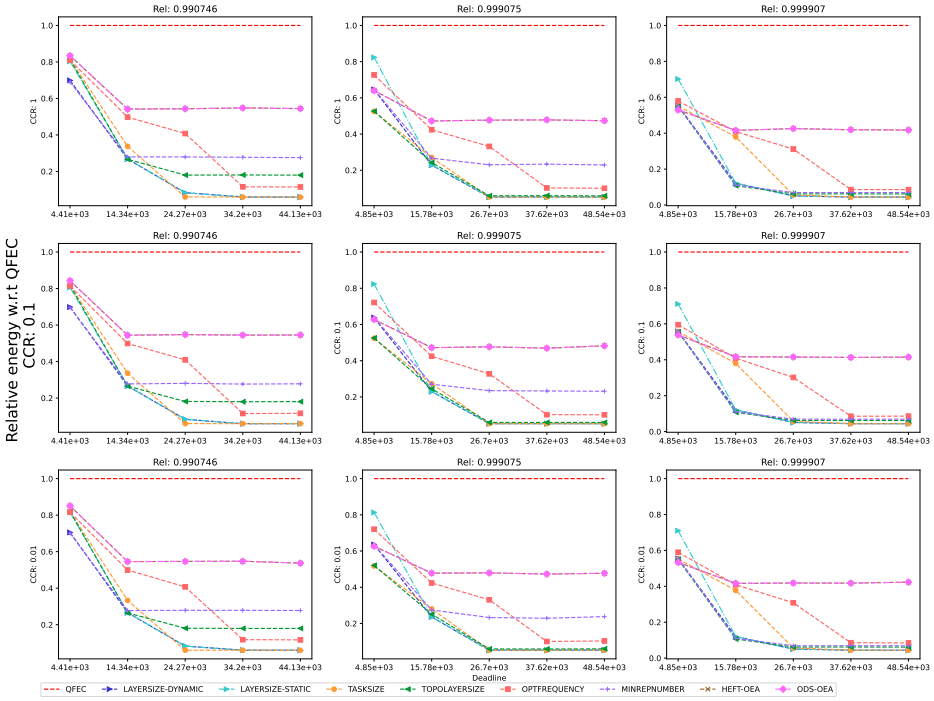

Figure 314: Performance of the different heuristics on the Epigenomics workflow.

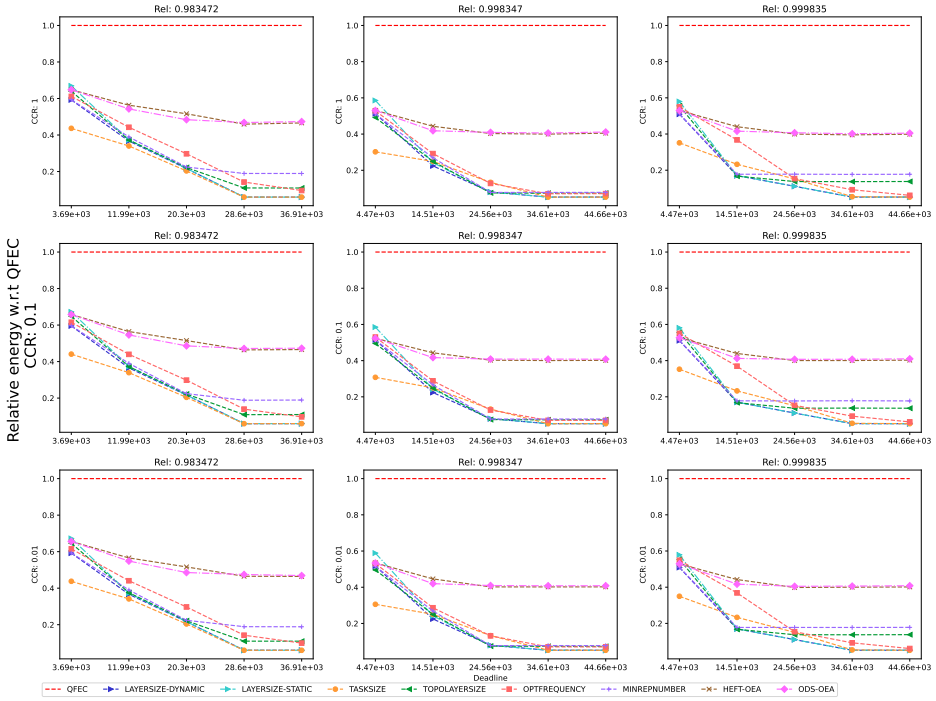

Figure 315: Performance of the different heuristics on the Genome workflow.

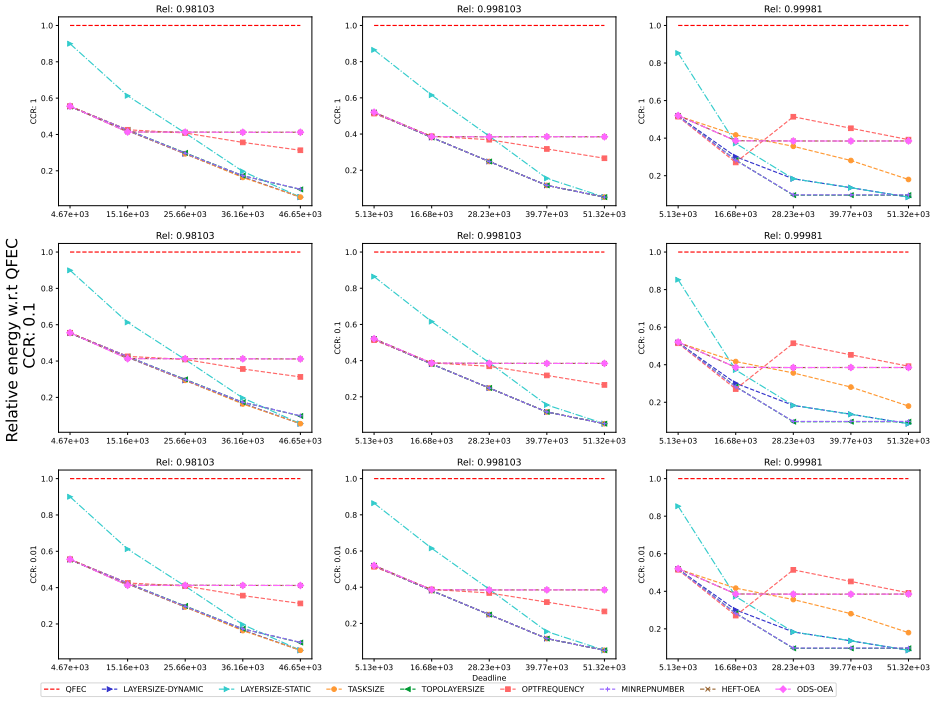

Figure 316: Performance of the different heuristics on the LU workflow.

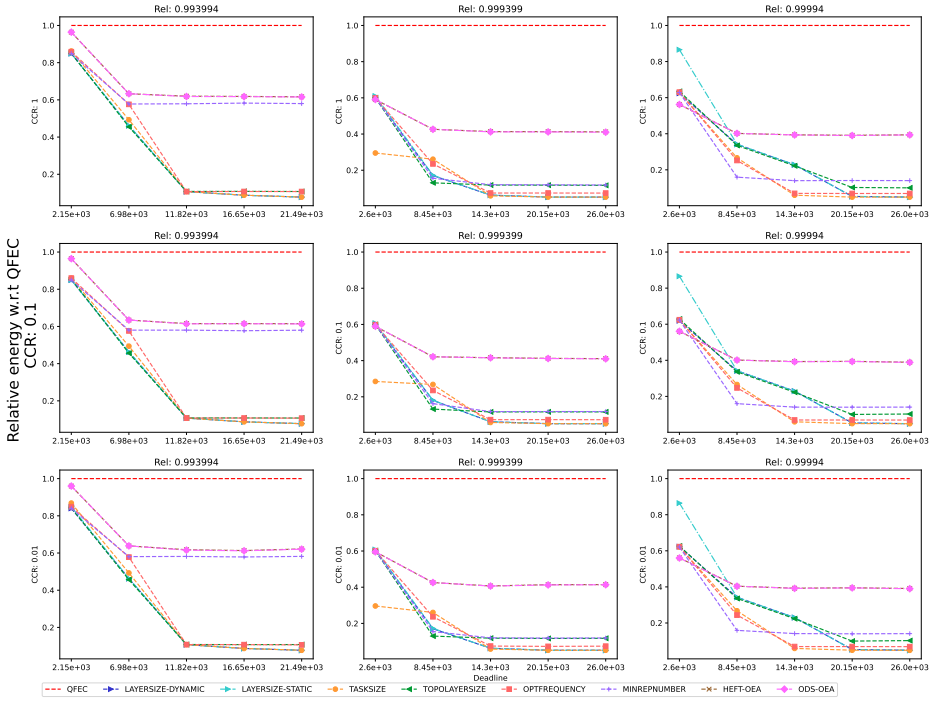

Figure 317: Performance of the different heuristics on the Montage workflow.

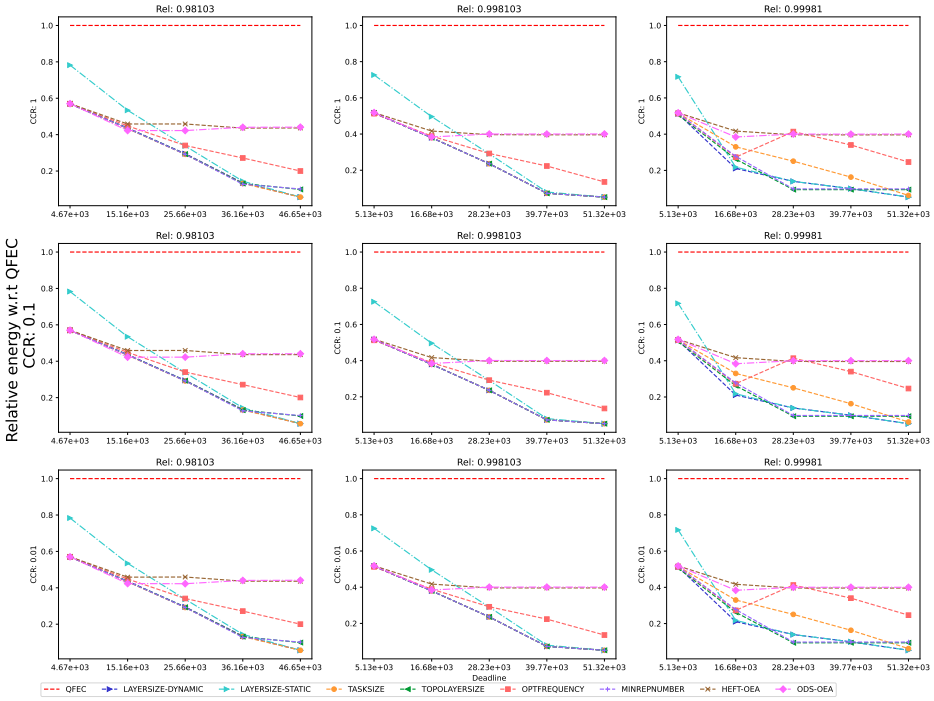

Figure 318: Performance of the different heuristics on the QR workflow.

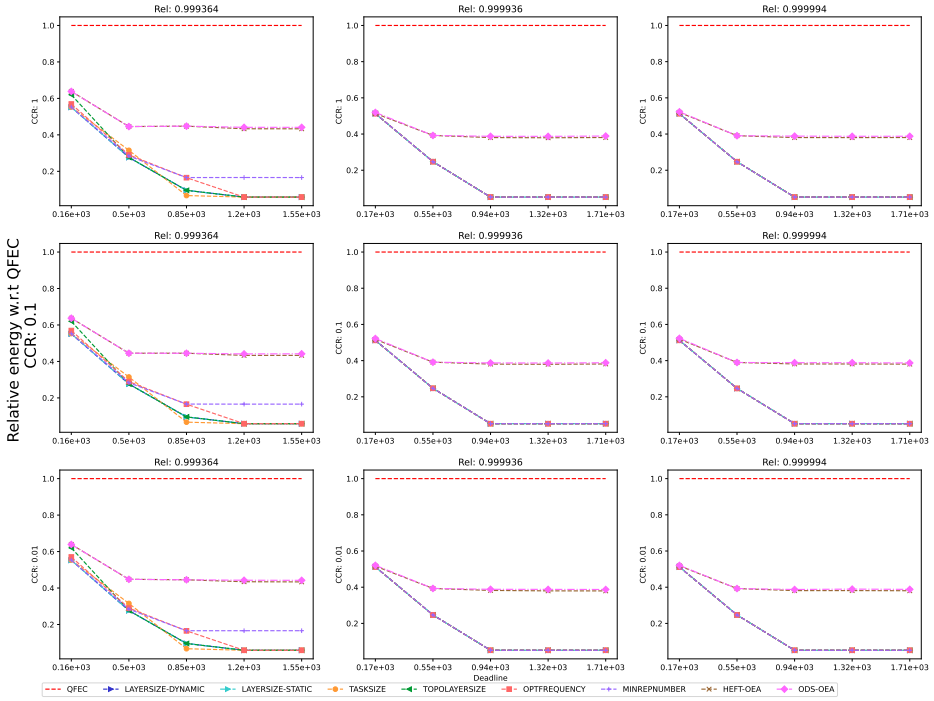

Figure 319: Performance of the different heuristics on the Seismology workflow.

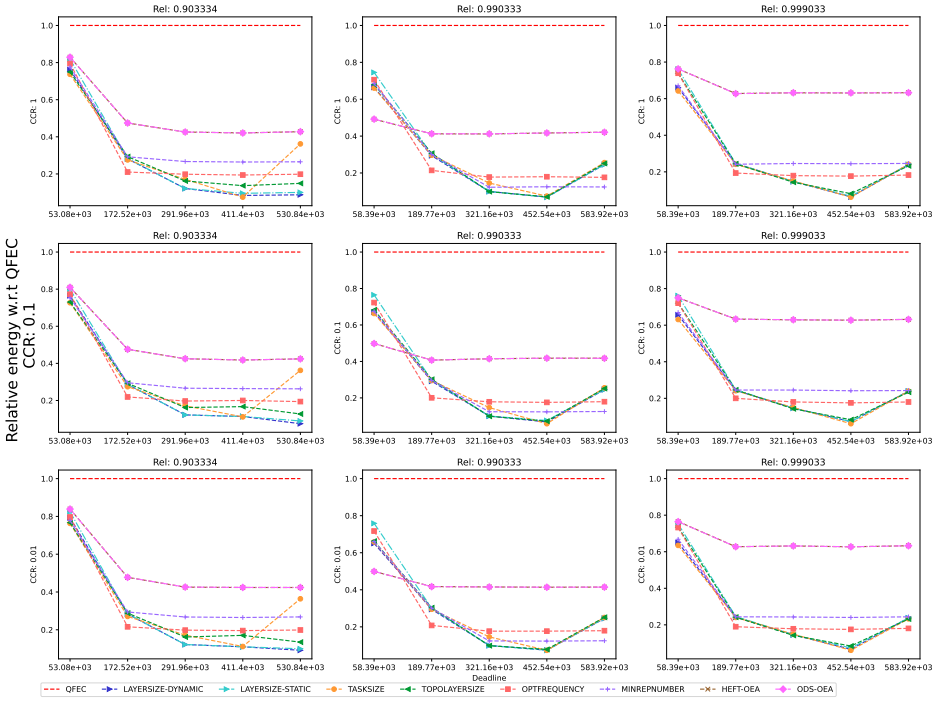

Figure 320: Performance of the different heuristics on the SoyKB workflow.

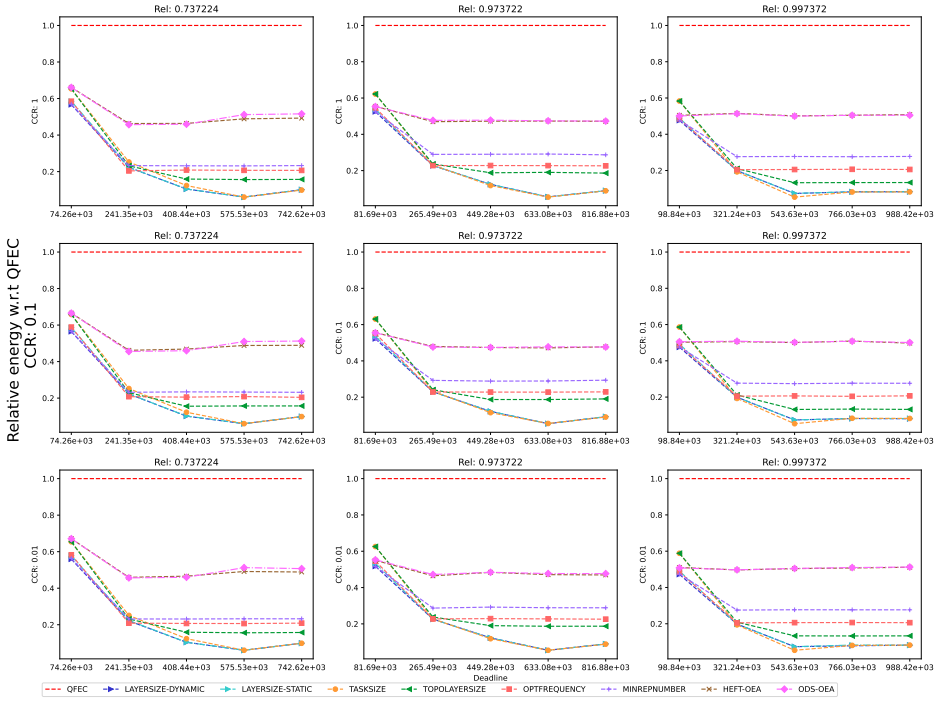

Figure 321: Performance of the different heuristics on the SRASearch.

D.7.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

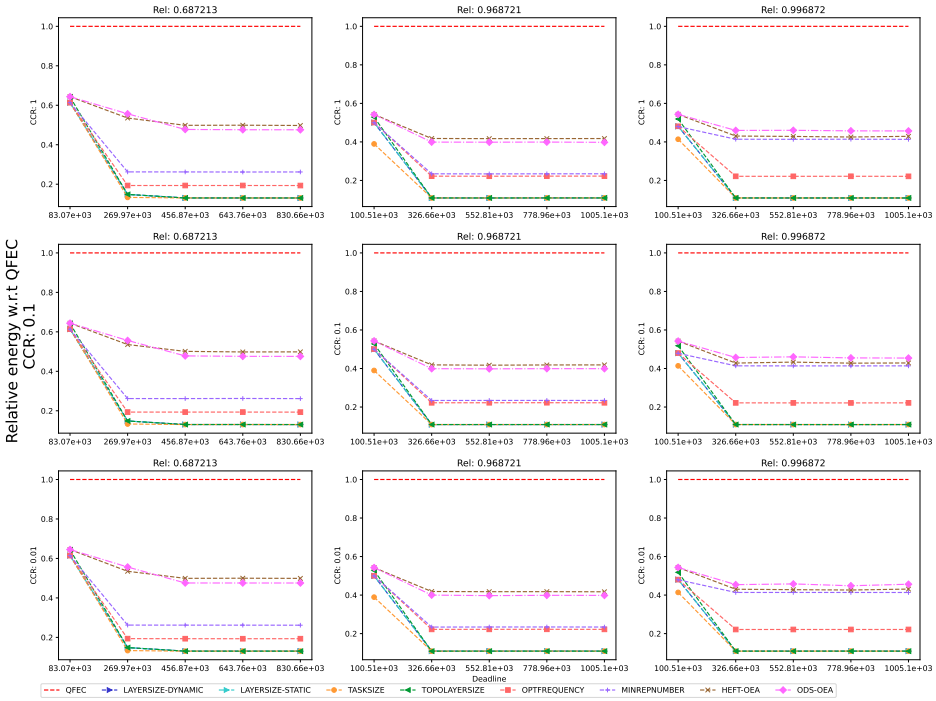

Figure 322: Performance of the different heuristics on the BLAST workflow.

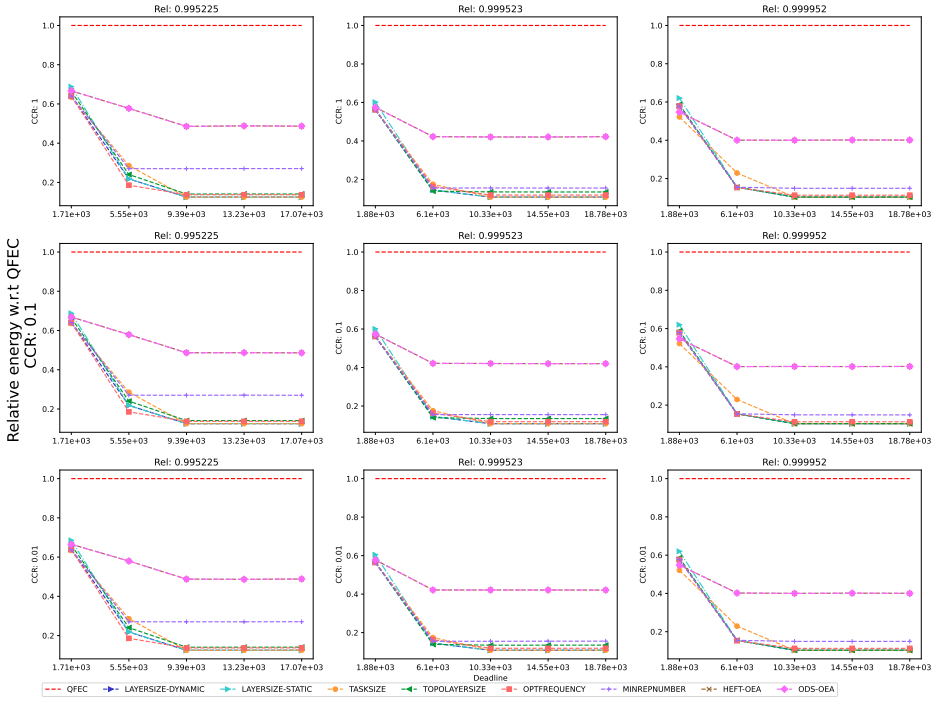

Figure 323: Performance of the different heuristics on the BWA workflow.

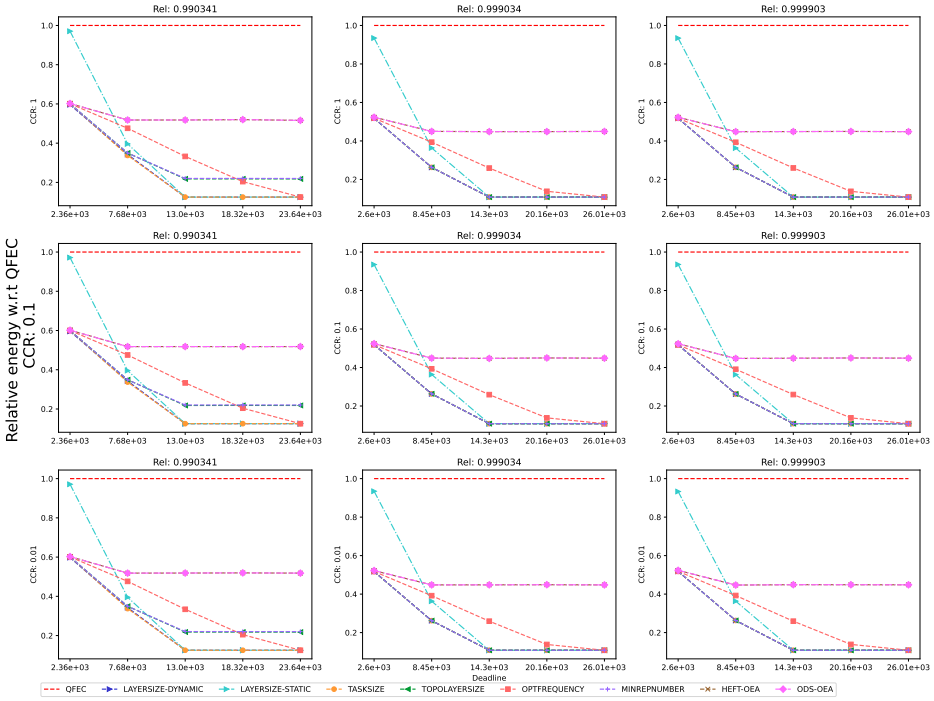

Figure 324: Performance of the different heuristics on the Cholesky workflow.

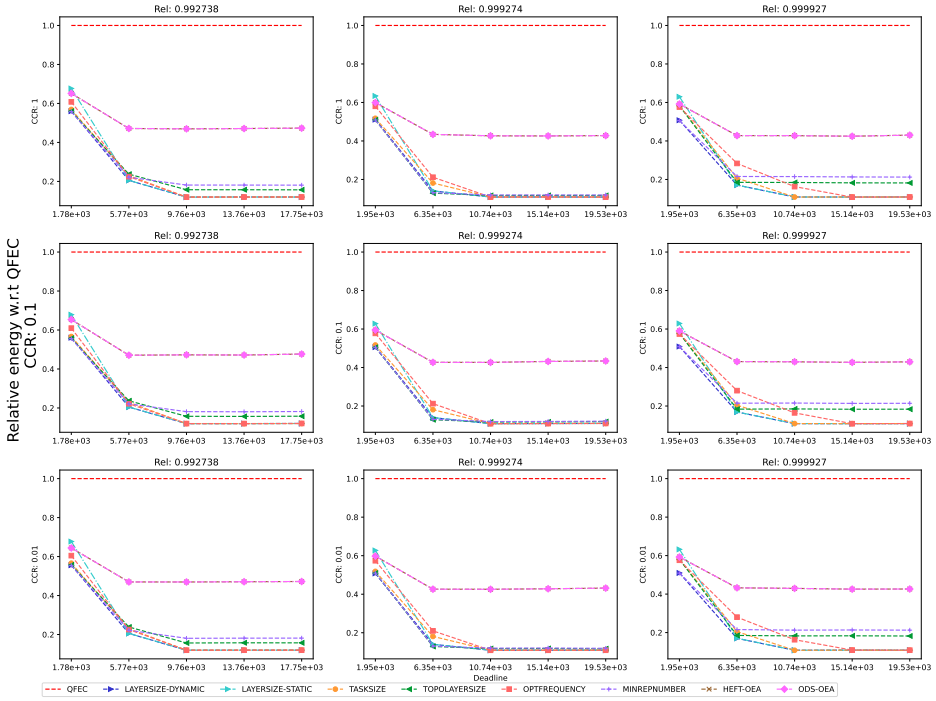

Figure 325: Performance of the different heuristics on the Cycles workflow.

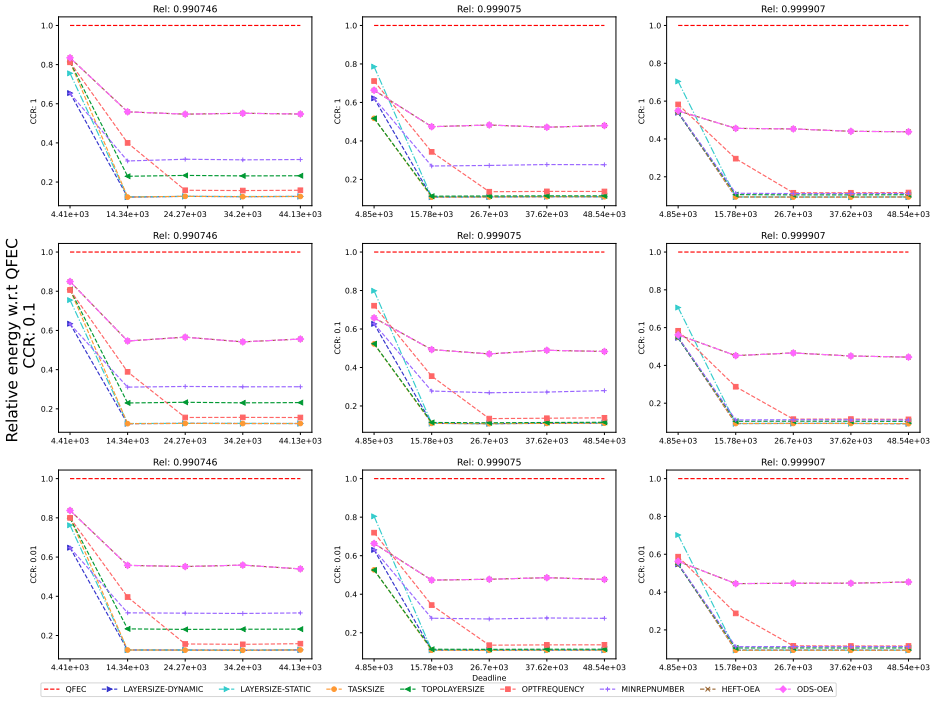

Figure 326: Performance of the different heuristics on the Epigenomics workflow.

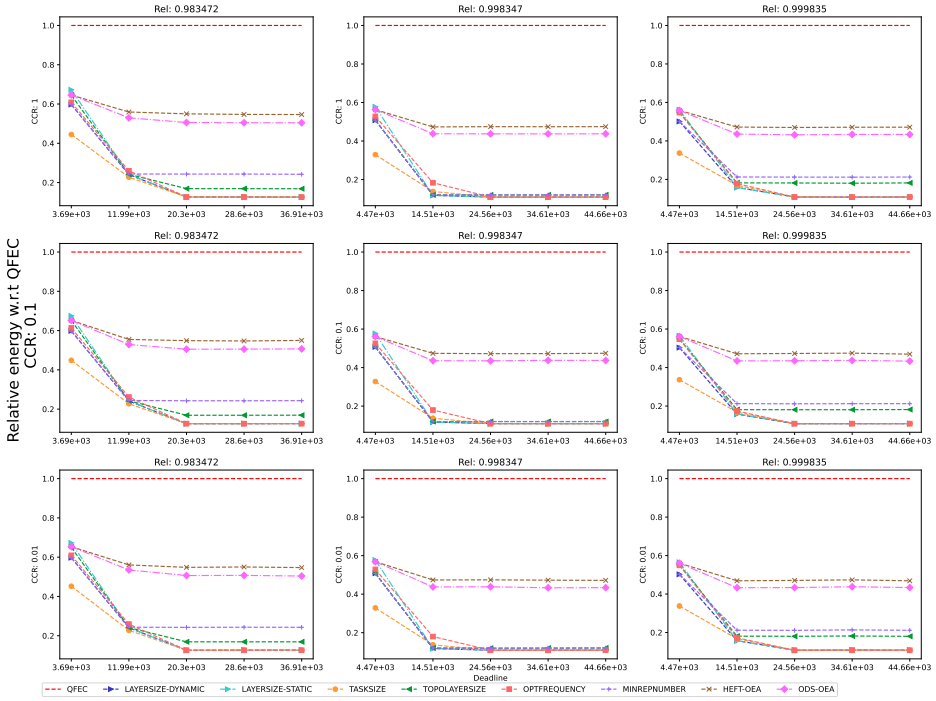

Figure 327: Performance of the different heuristics on the Genome workflow.

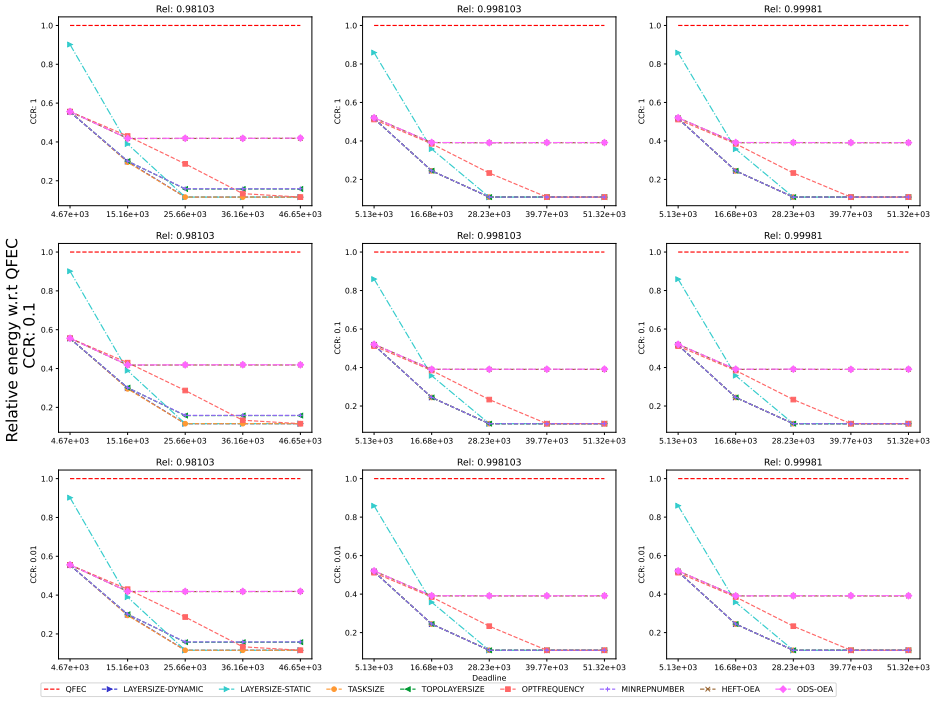

Figure 328: Performance of the different heuristics on the LU workflow.

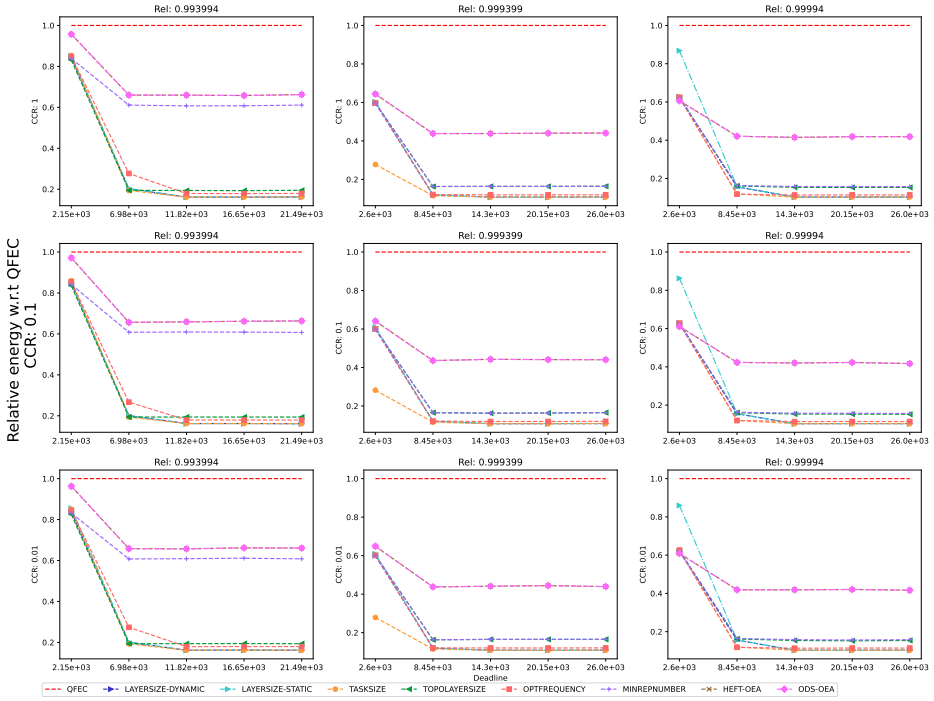

Figure 329: Performance of the different heuristics on the Montage workflow.

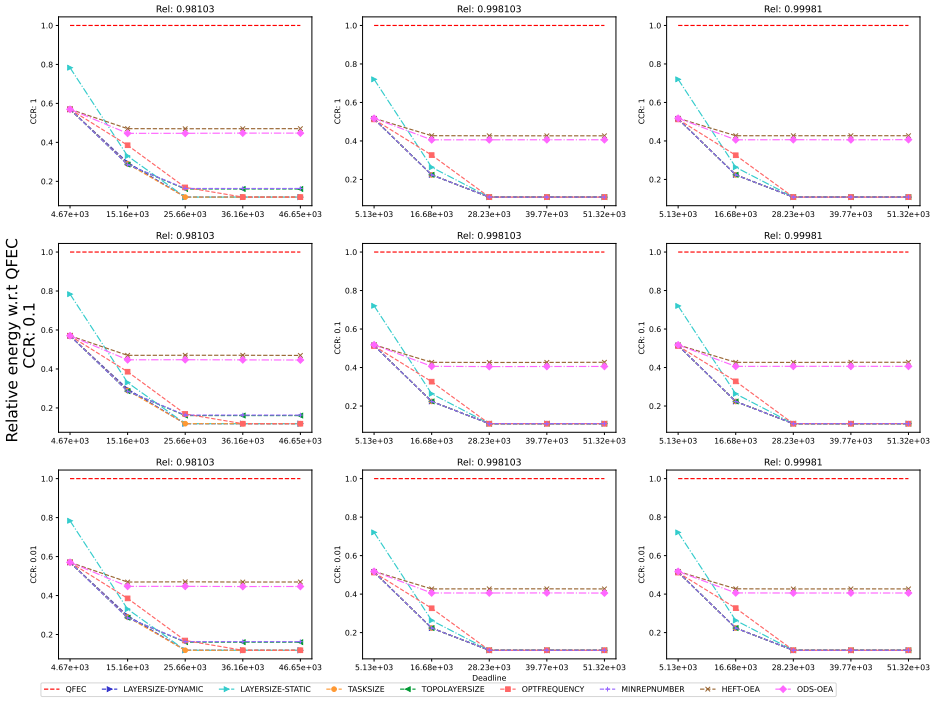

Figure 330: Performance of the different heuristics on the QR workflow.

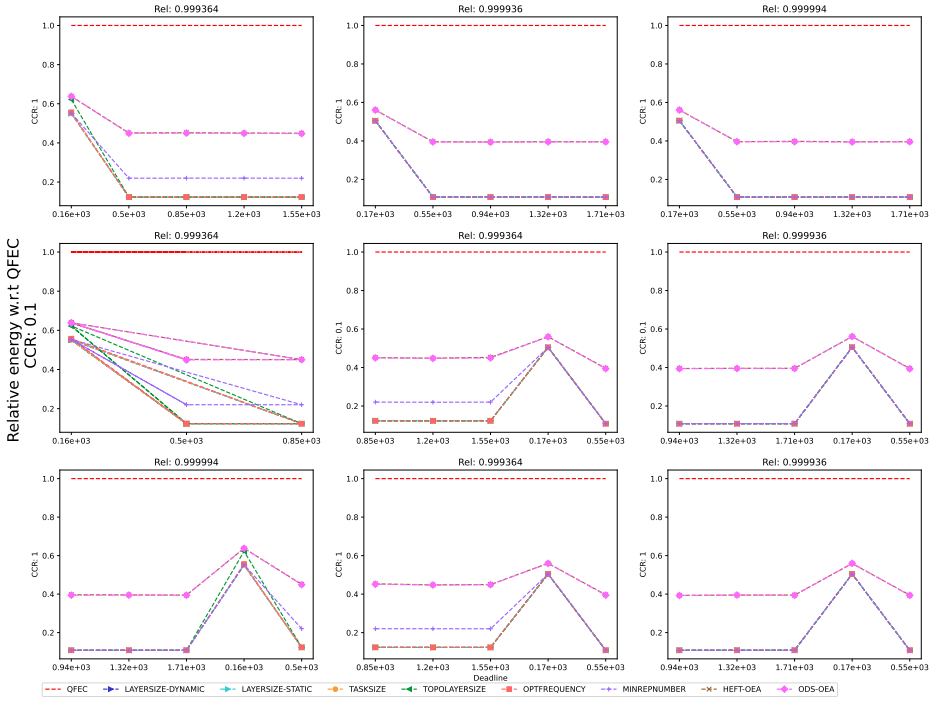

Figure 331: Performance of the different heuristics on the Seismology workflow.

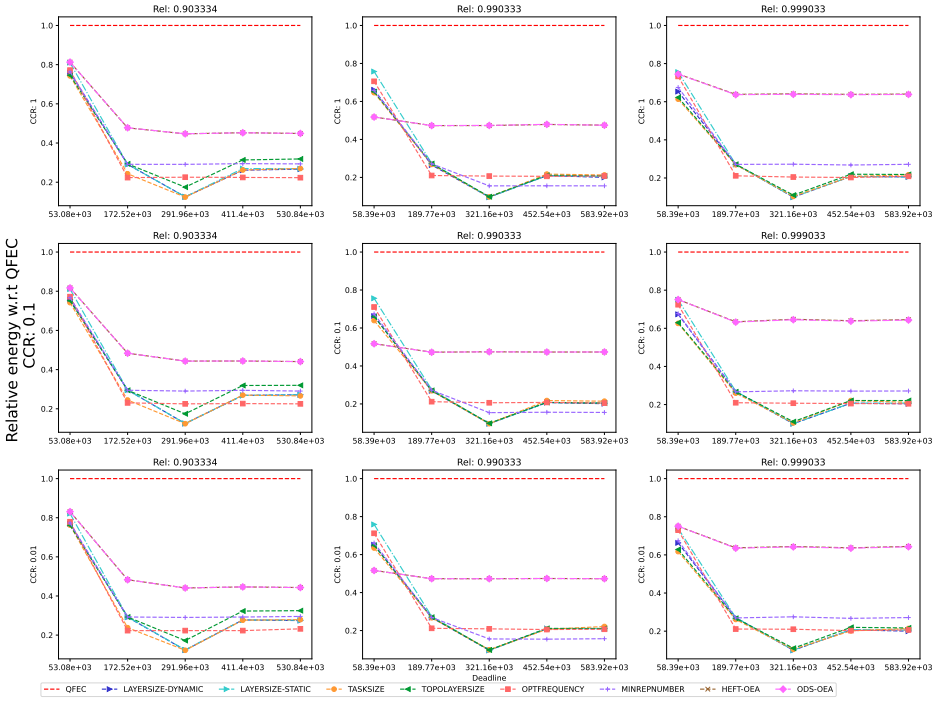

Figure 332: Performance of the different heuristics on the SoyKB workflow.

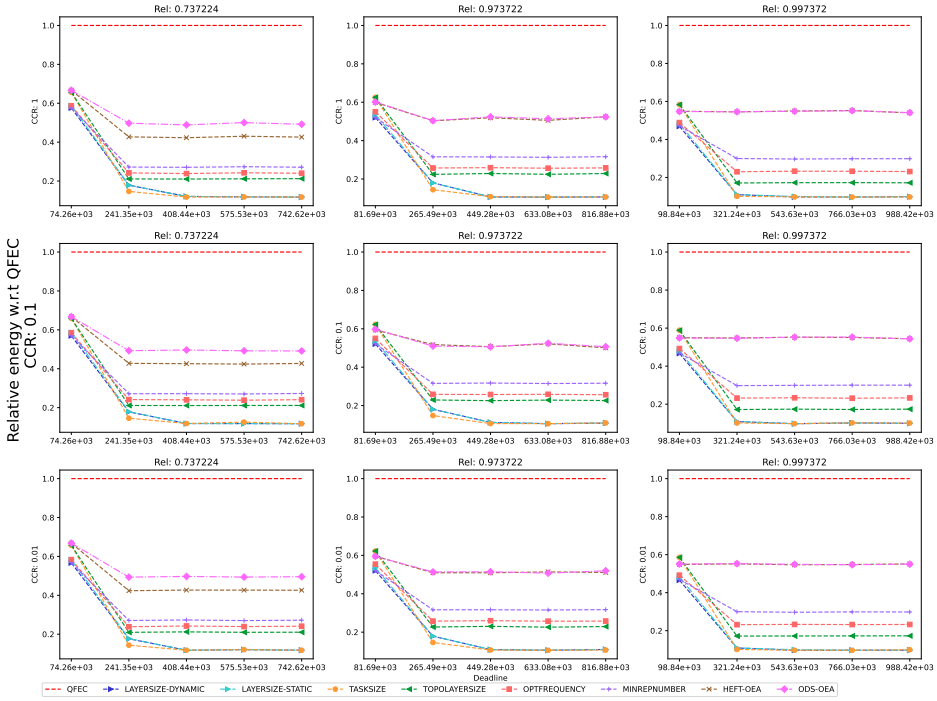

Figure 333: Performance of the different heuristics on the SRASearch.

D.7.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

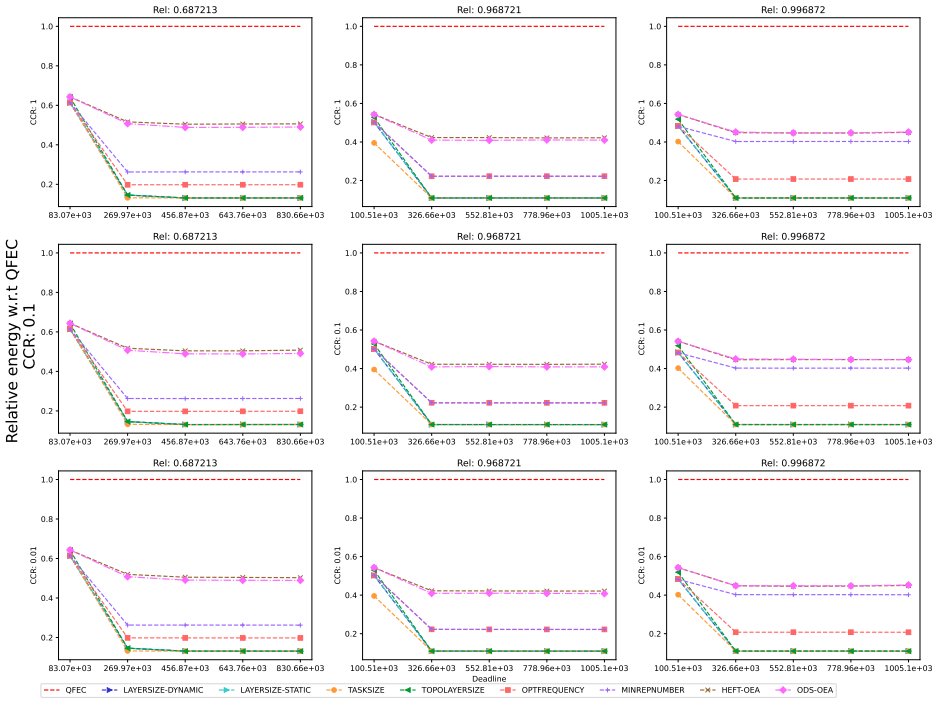

Figure 334: Performance of the different heuristics on the BLAST workflow.

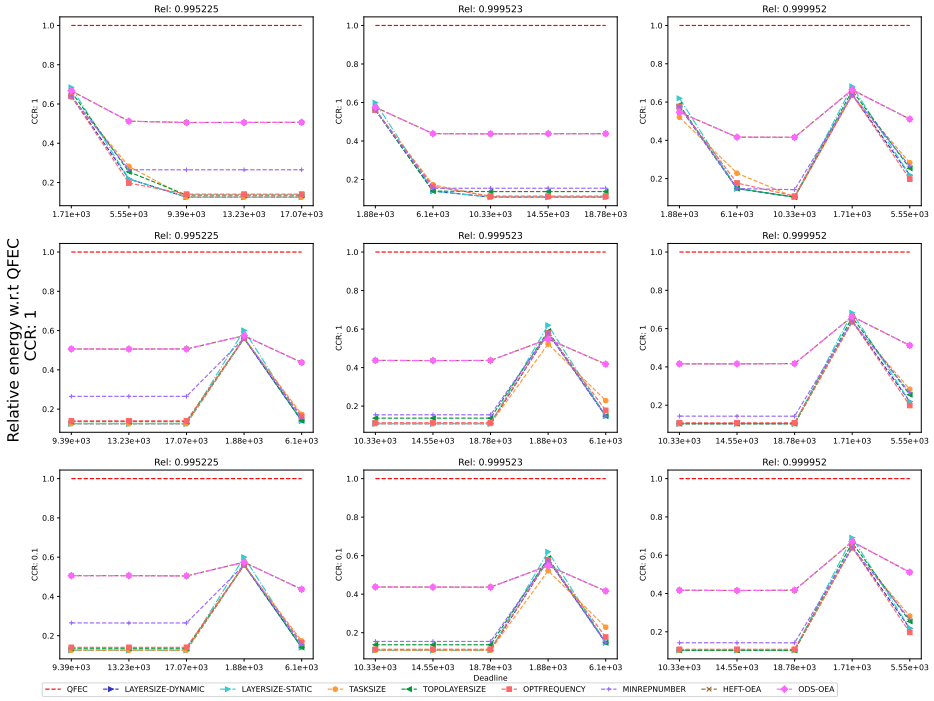

Figure 335: Performance of the different heuristics on the BWA workflow.

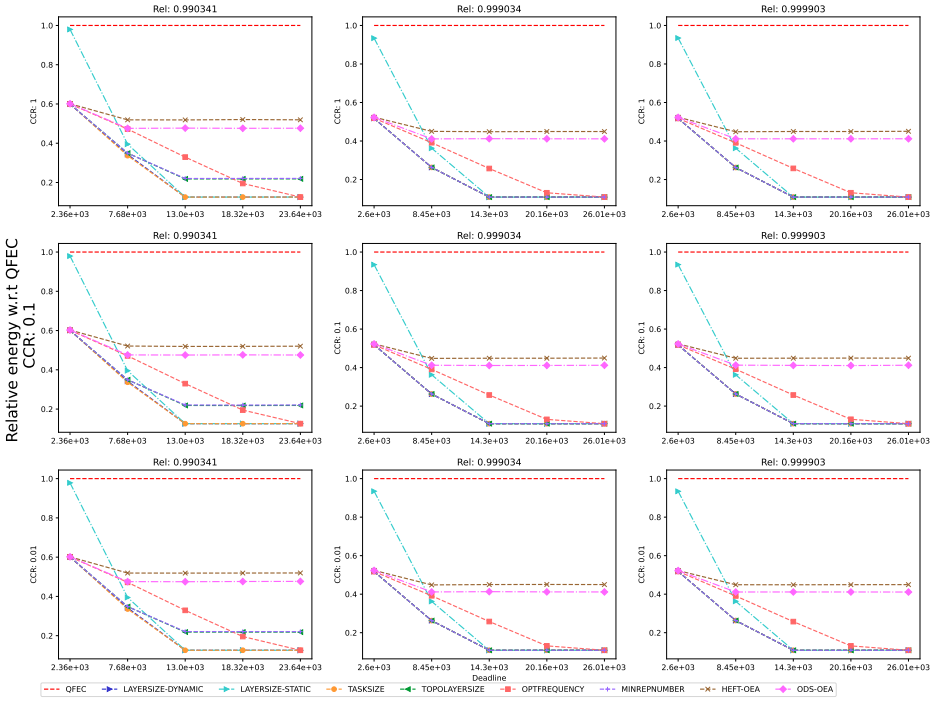

Figure 336: Performance of the different heuristics on the Cholesky workflow.

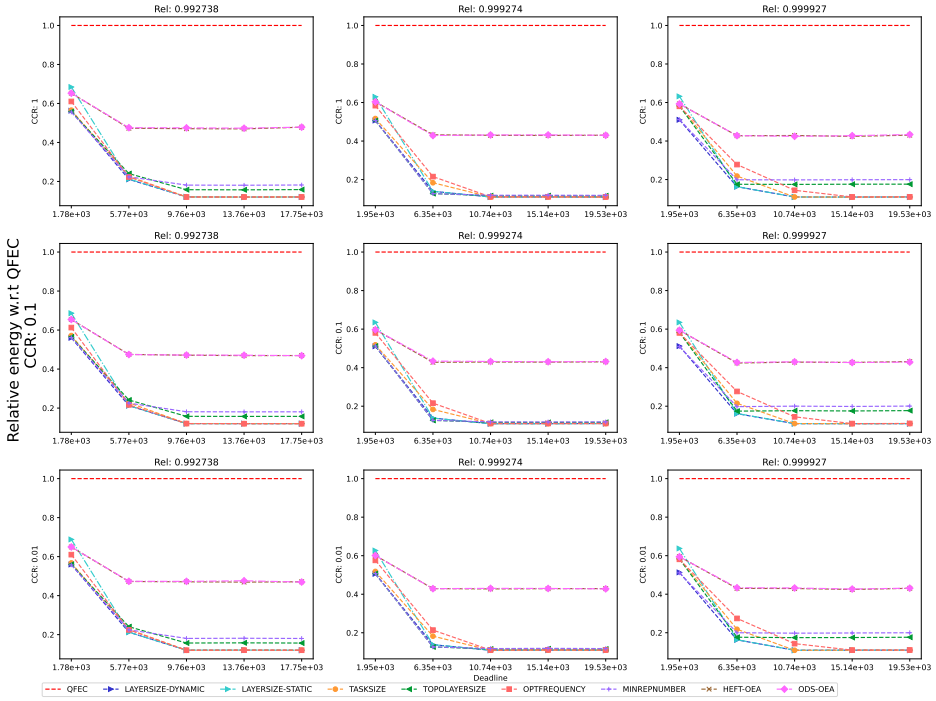

Figure 337: Performance of the different heuristics on the Cycles workflow.

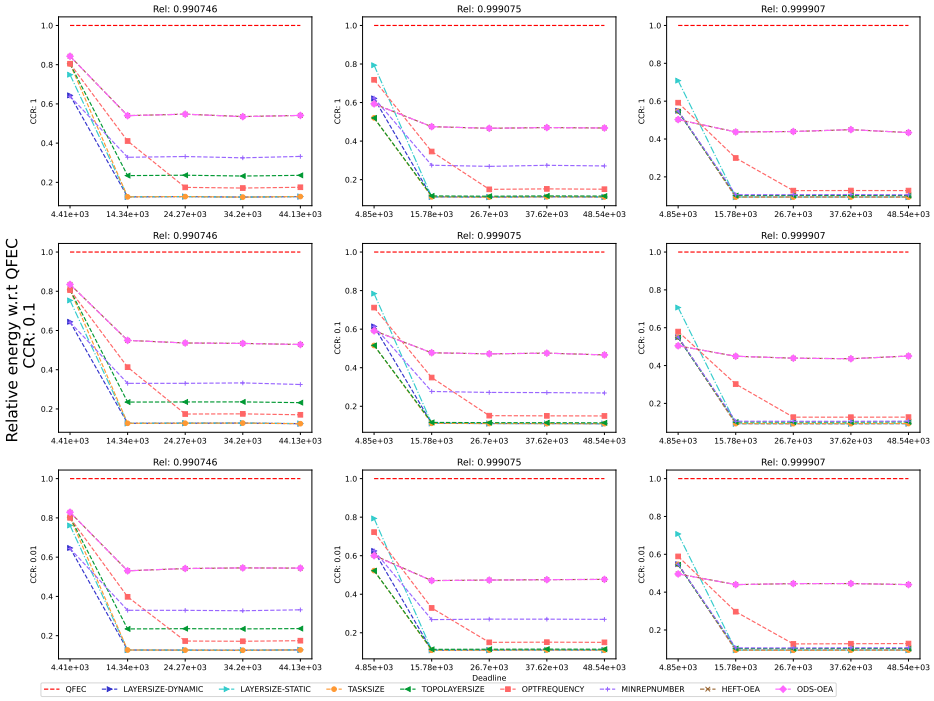

Figure 338: Performance of the different heuristics on the Epigenomics workflow.

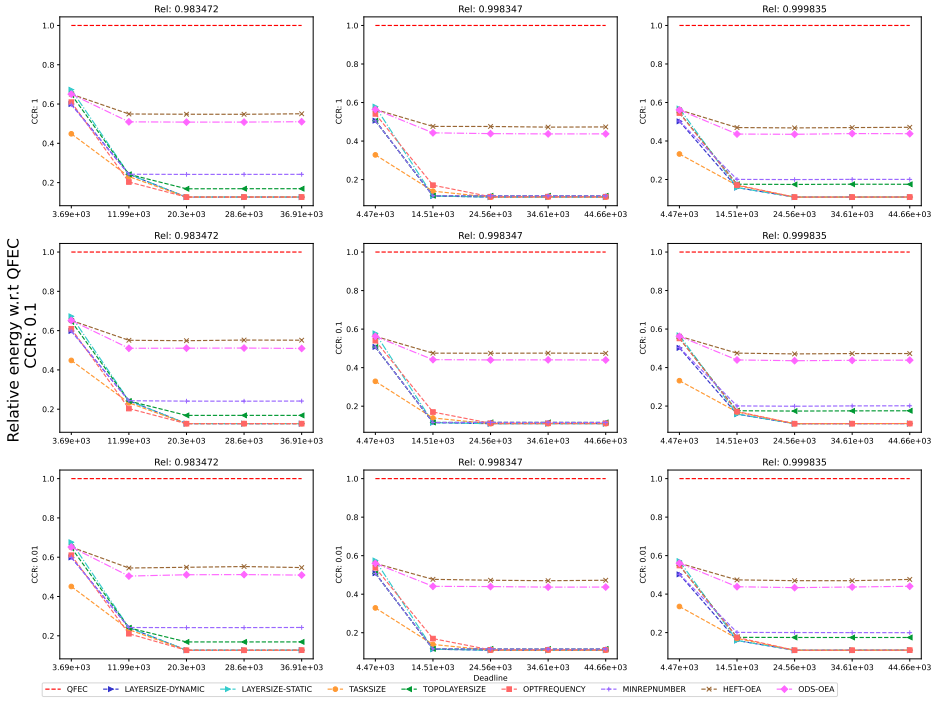

Figure 339: Performance of the different heuristics on the Genome workflow.

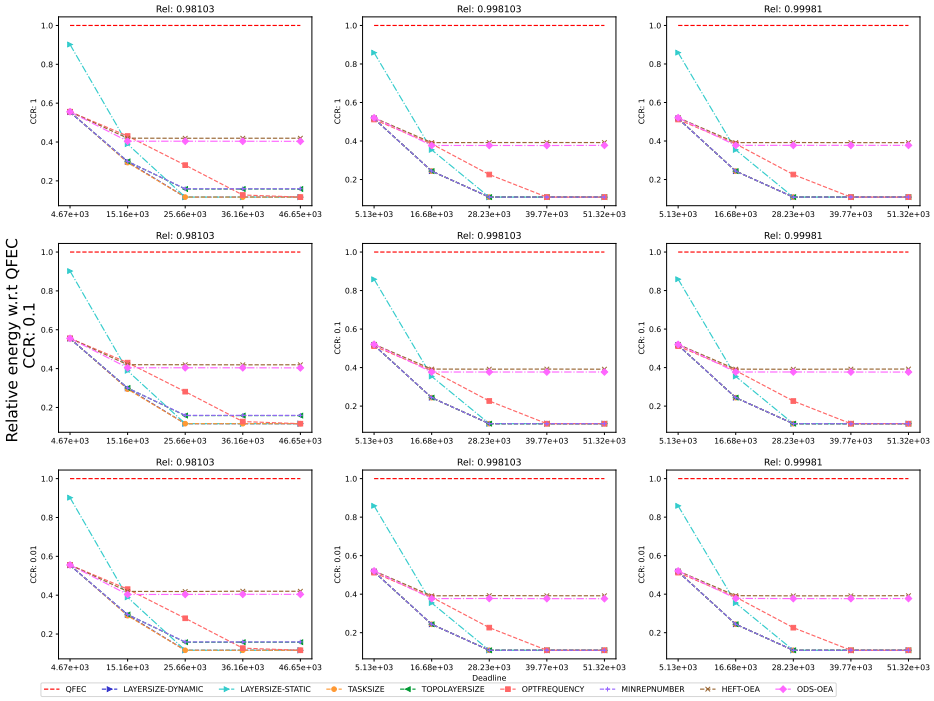

Figure 340: Performance of the different heuristics on the LU workflow.

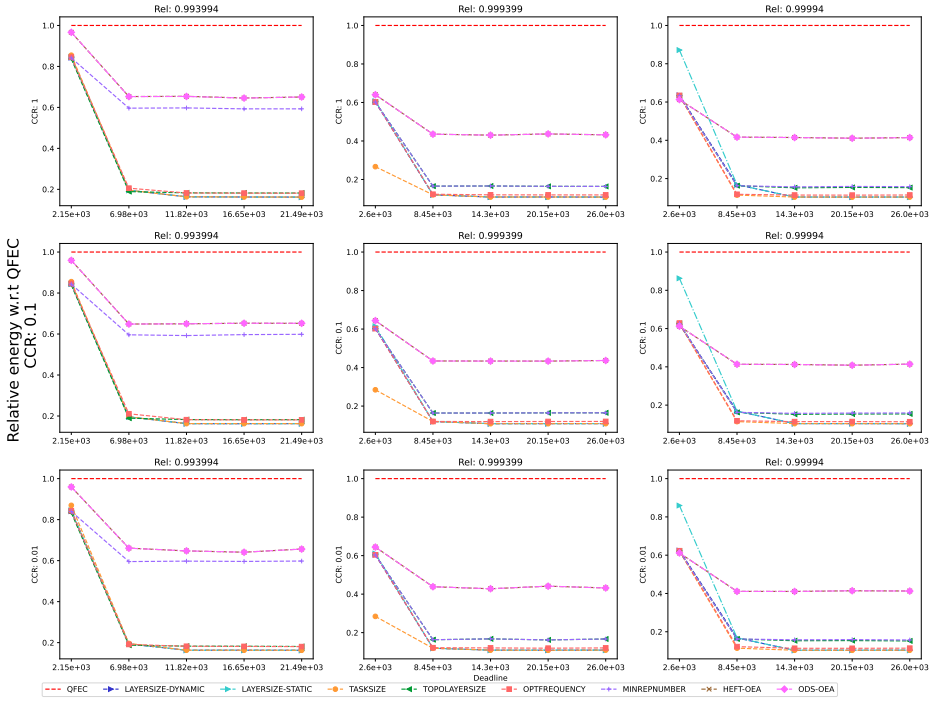

Figure 341: Performance of the different heuristics on the Montage workflow.

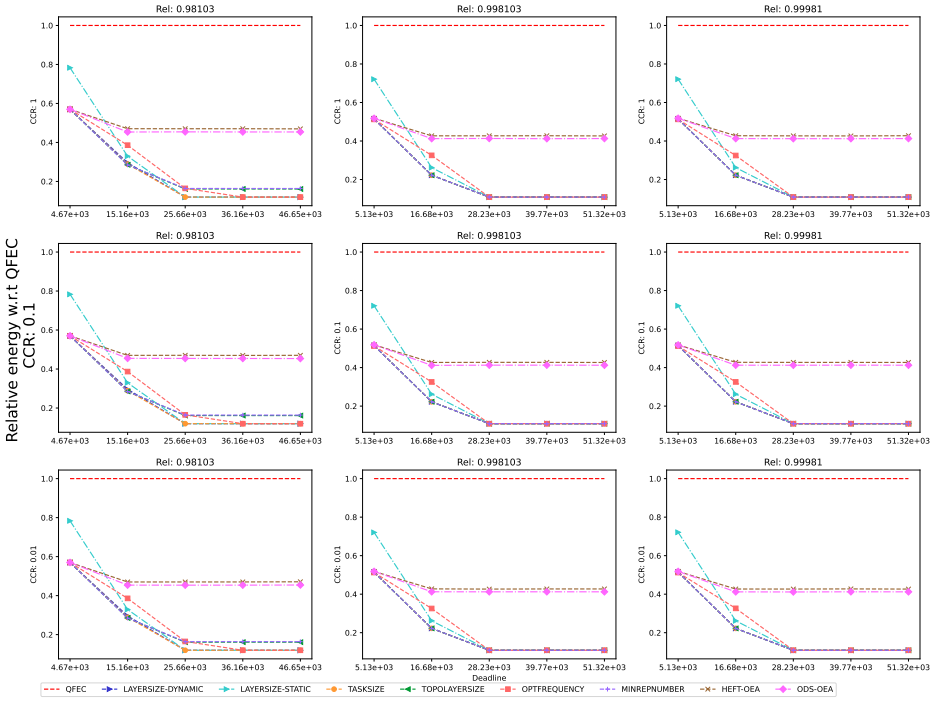

Figure 342: Performance of the different heuristics on the QR workflow.

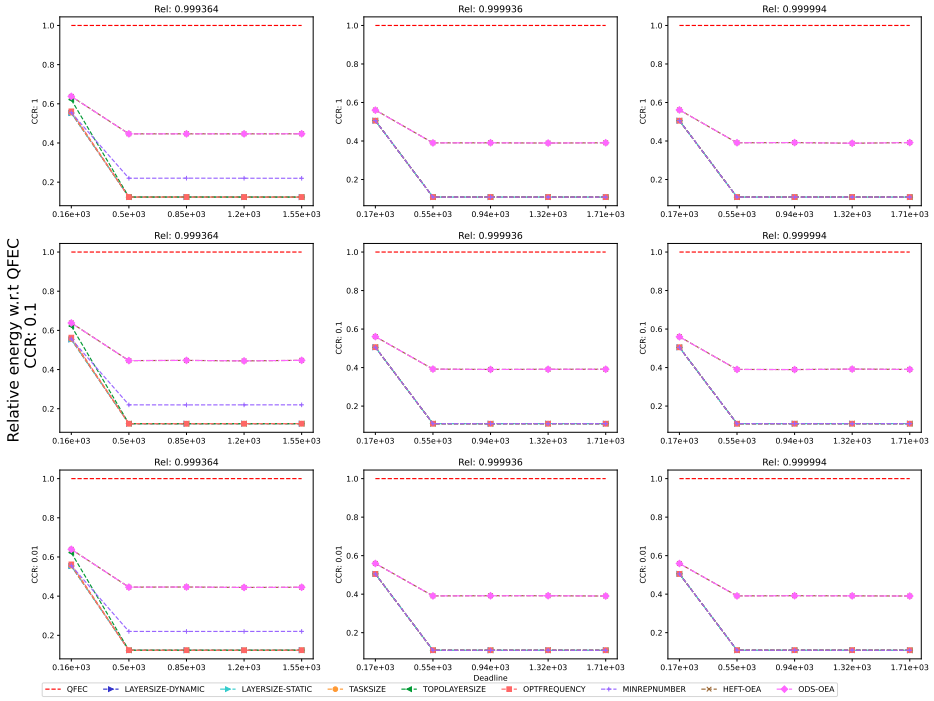

Figure 343: Performance of the different heuristics on the Seismology workflow.

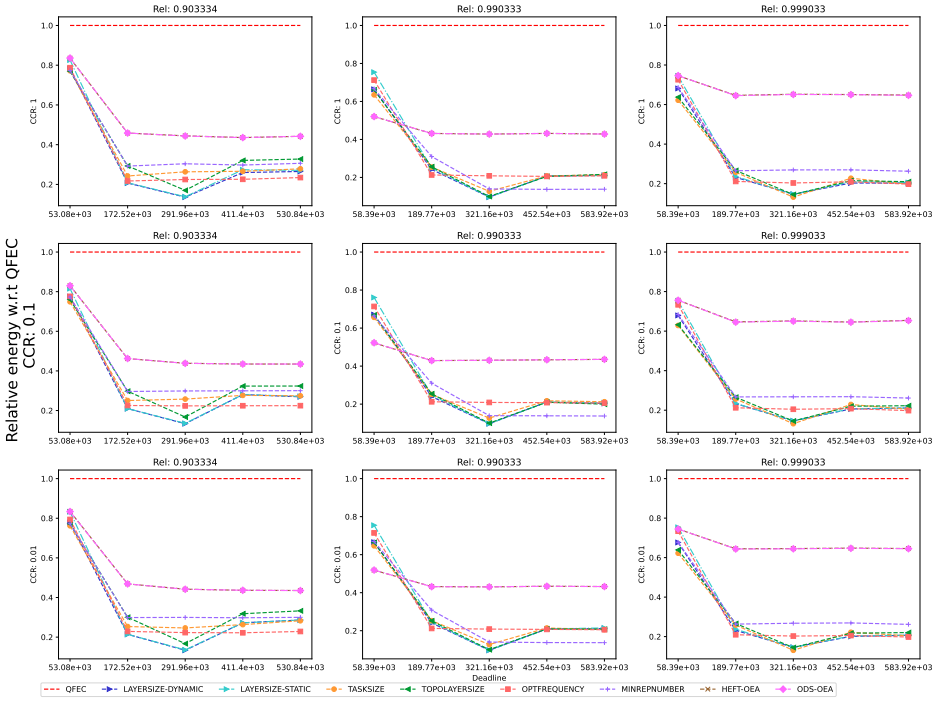

Figure 344: Performance of the different heuristics on the SoyKB workflow.

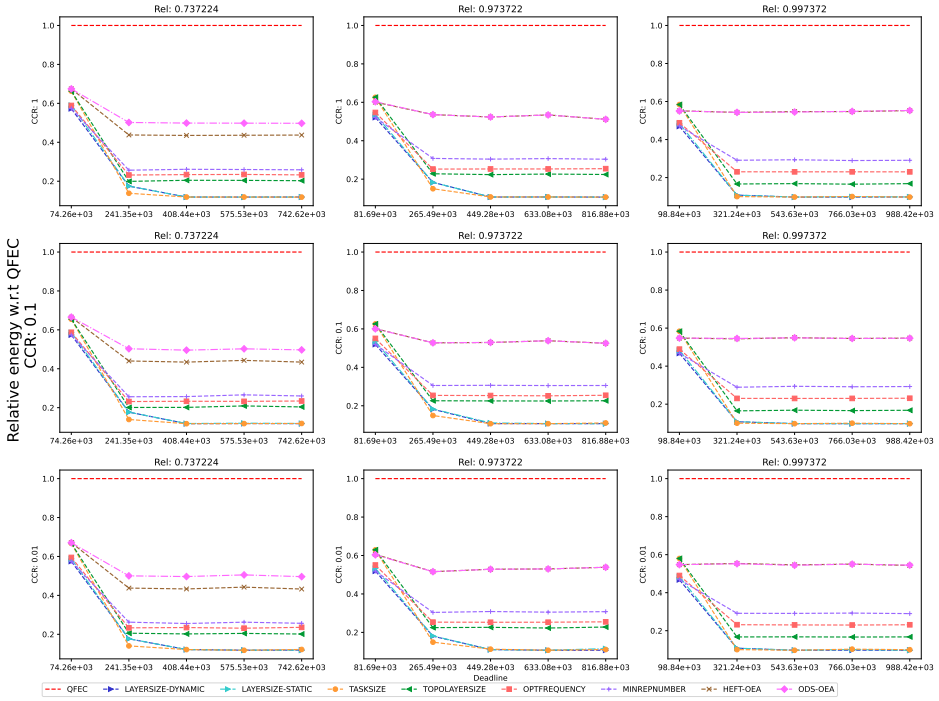

Figure 345: Performance of the different heuristics on the SRASearch.

D.8 $BC/WC = 0.8$

D.8.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

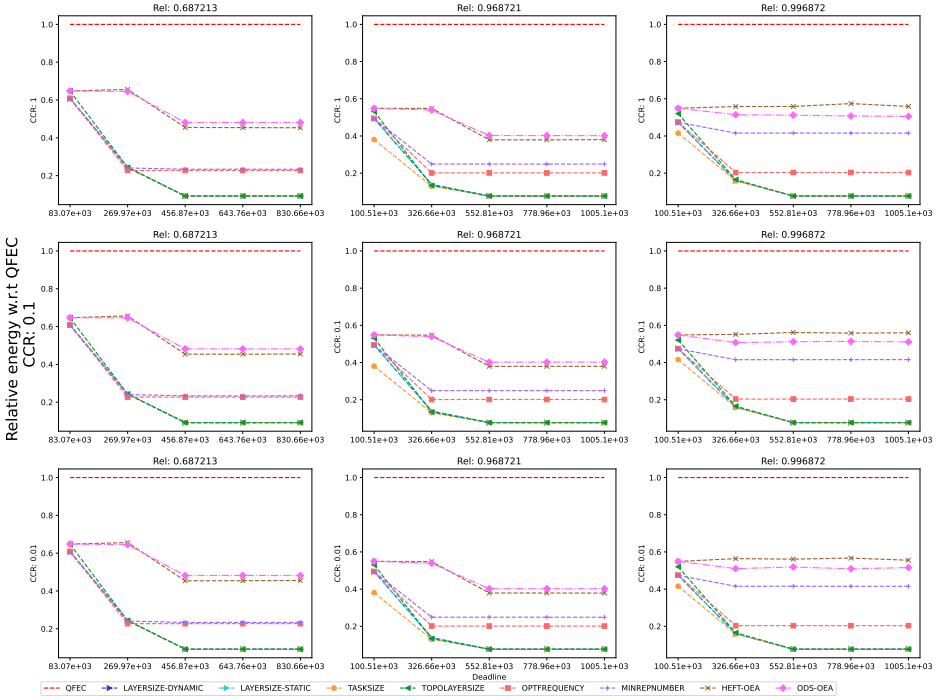

Figure 346: Performance of the different heuristics on the BLAST workflow.

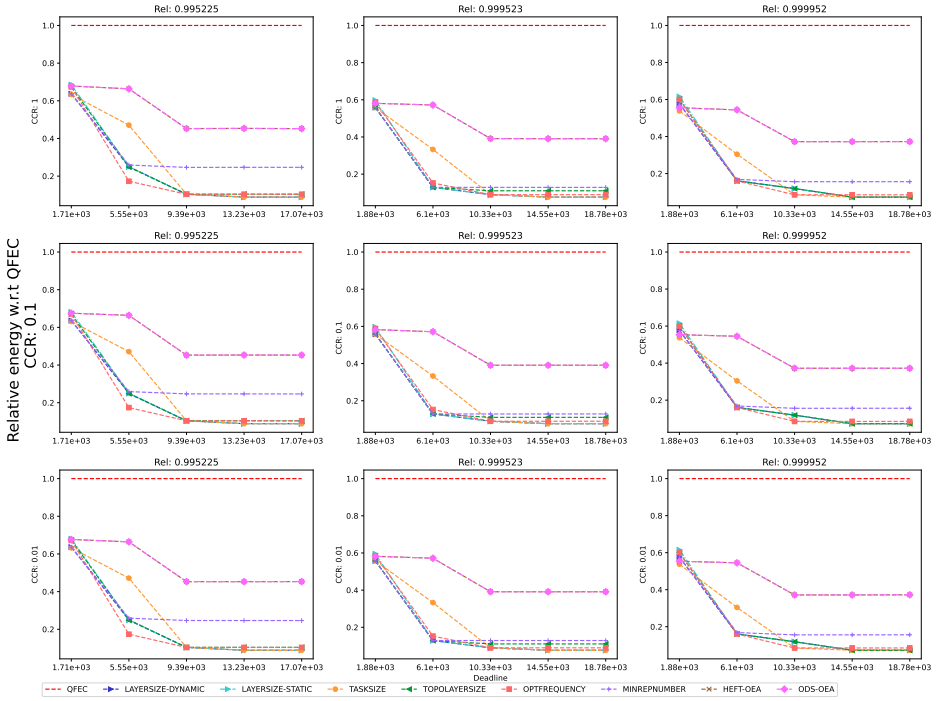

Figure 347: Performance of the different heuristics on the BWA workflow.

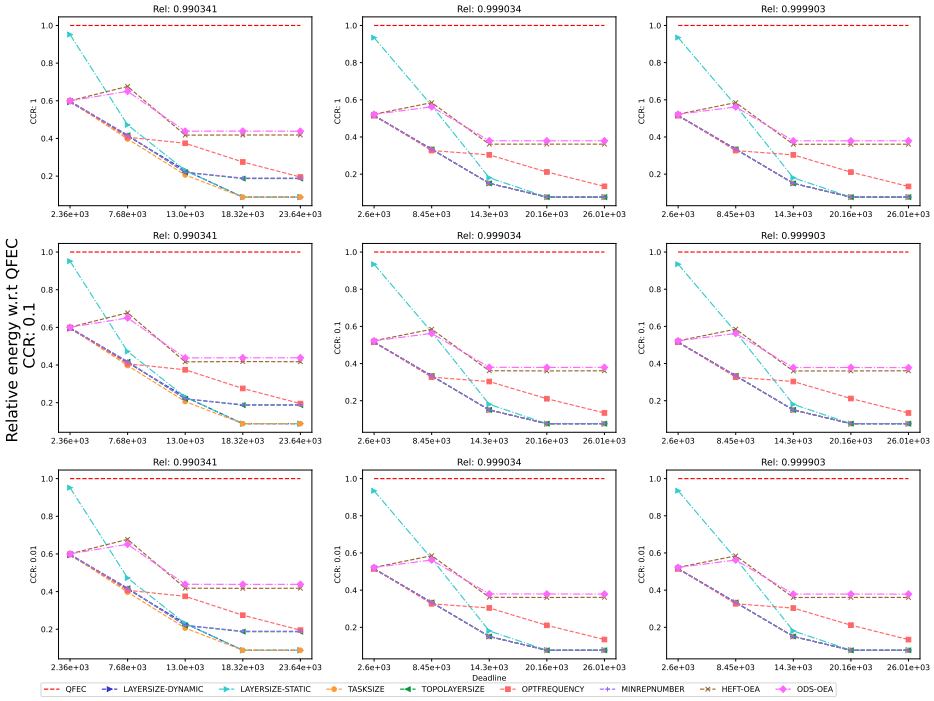

Figure 348: Performance of the different heuristics on the Cholesky workflow.

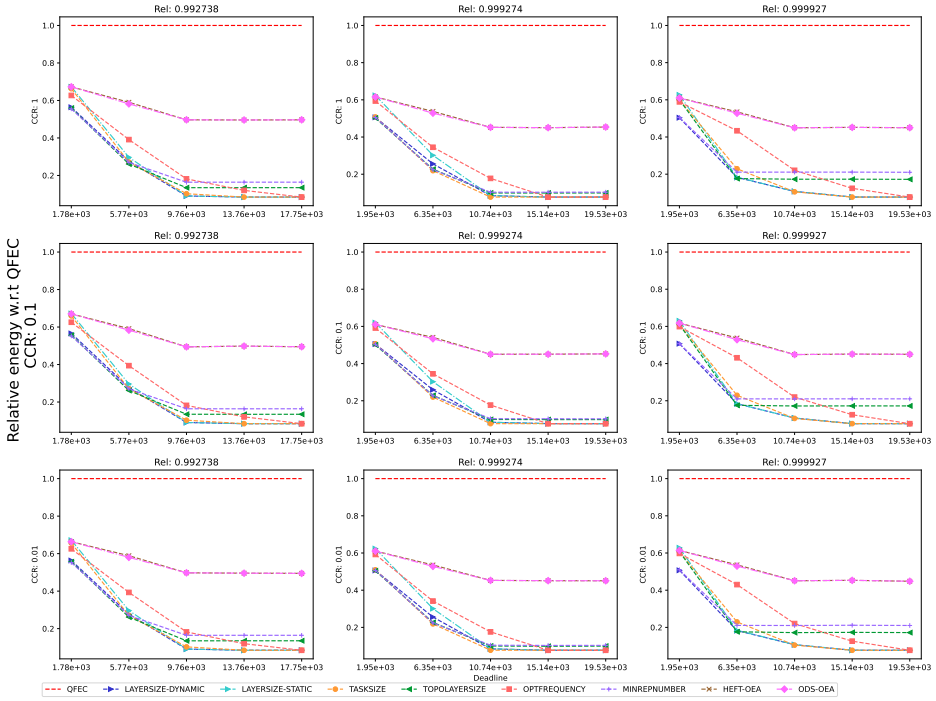

Figure 349: Performance of the different heuristics on the Cycles workflow.

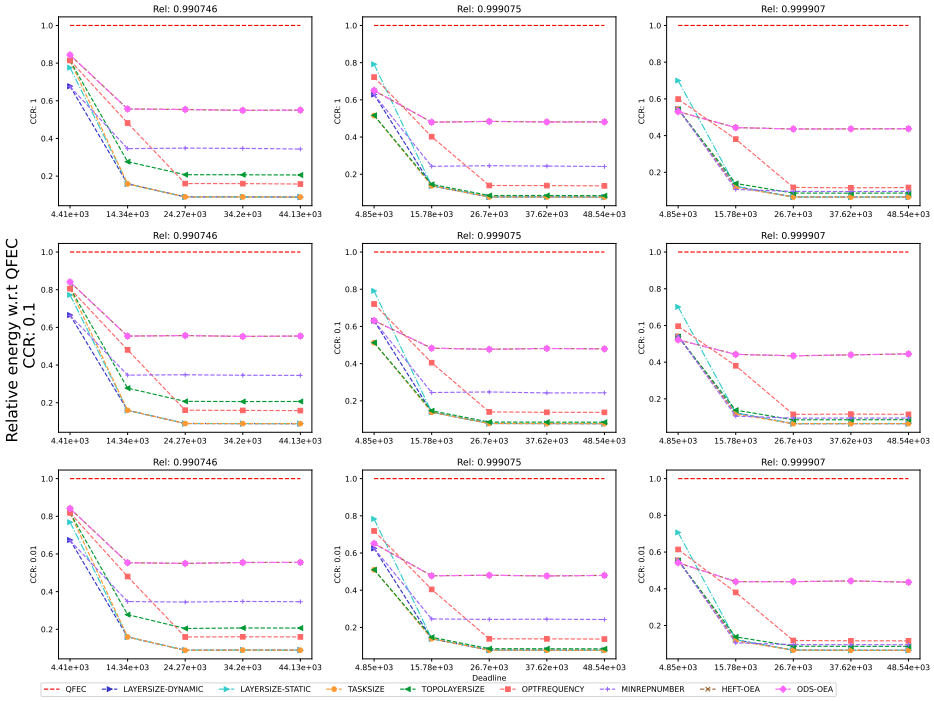

Figure 350: Performance of the different heuristics on the Epigenomics workflow.

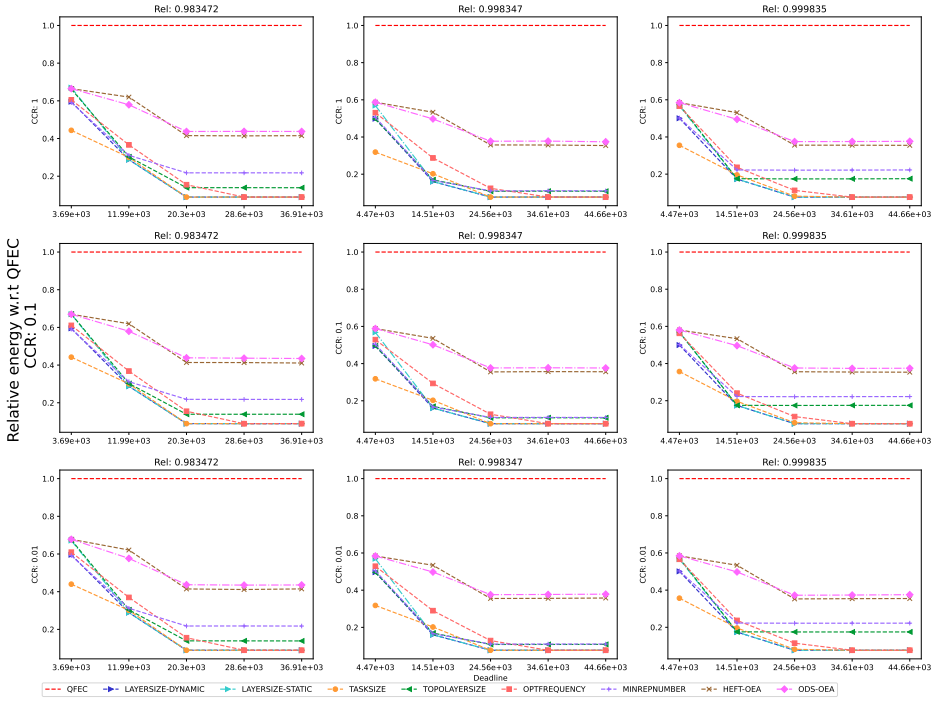

Figure 351: Performance of the different heuristics on the Genome workflow.

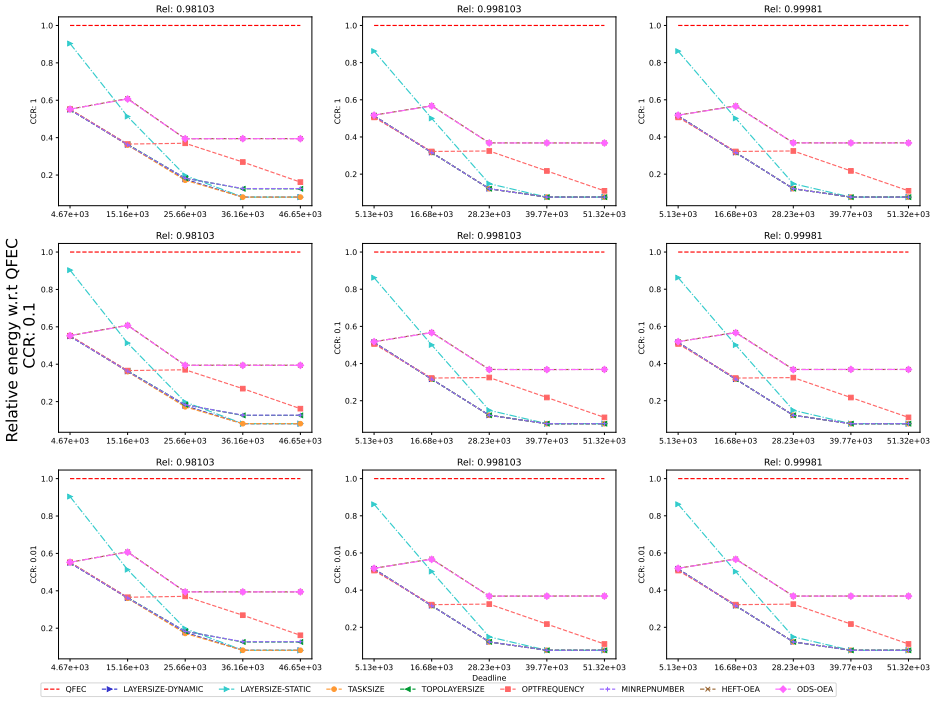

Figure 352: Performance of the different heuristics on the LU workflow.

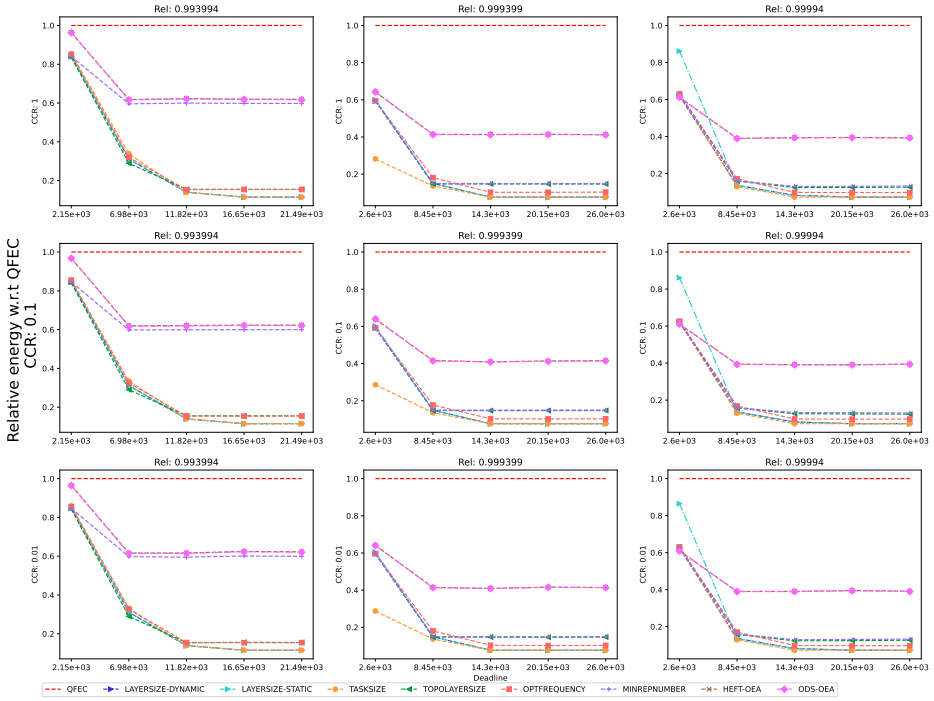

Figure 353: Performance of the different heuristics on the Montage workflow.

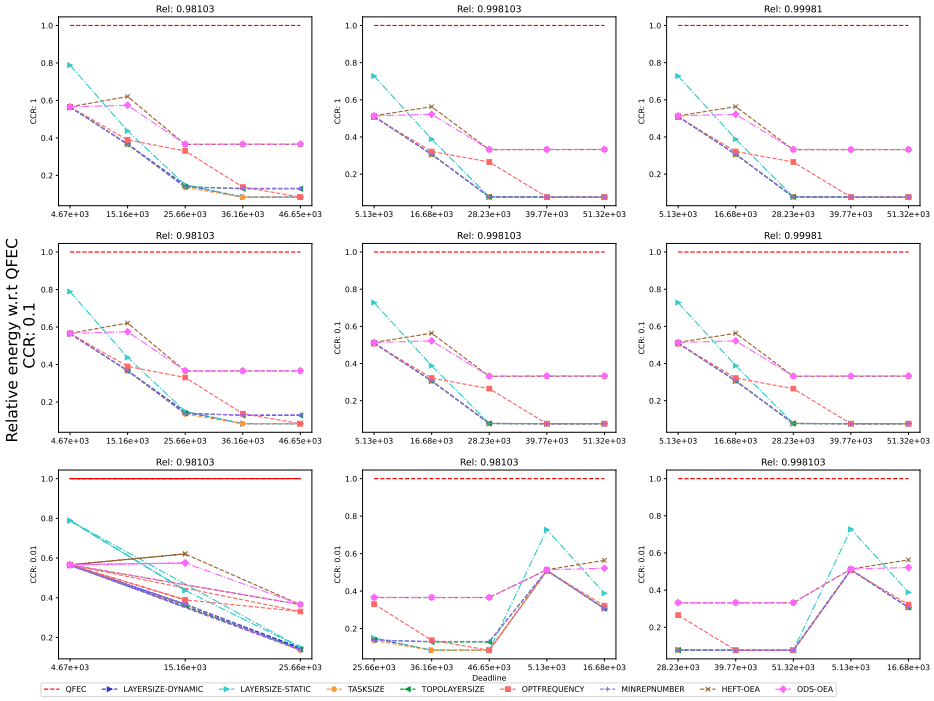

Figure 354: Performance of the different heuristics on the QR workflow.

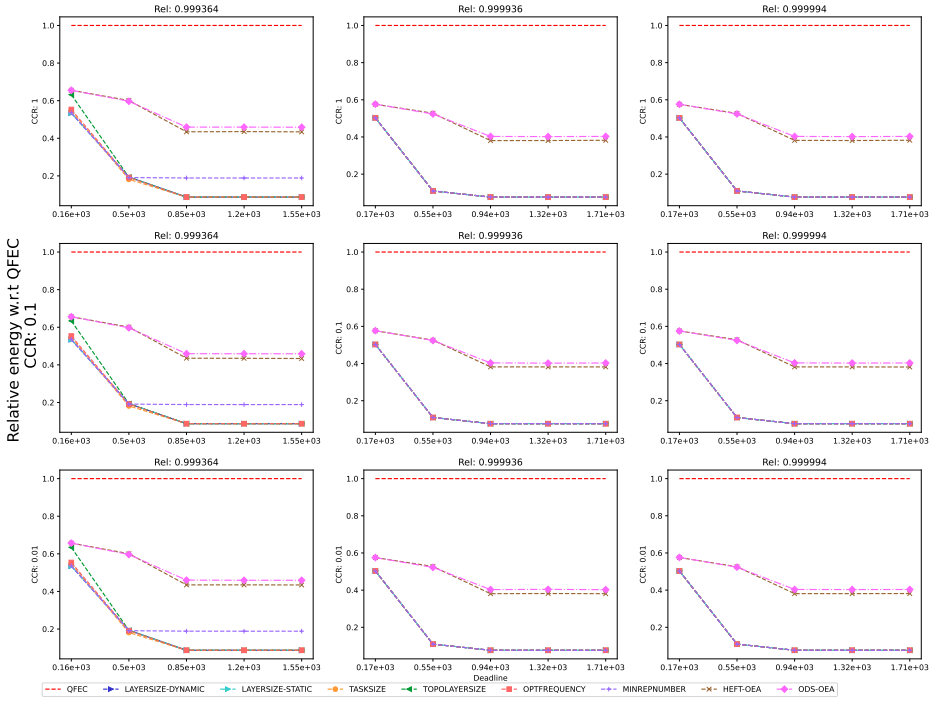

Figure 355: Performance of the different heuristics on the Seismology workflow.

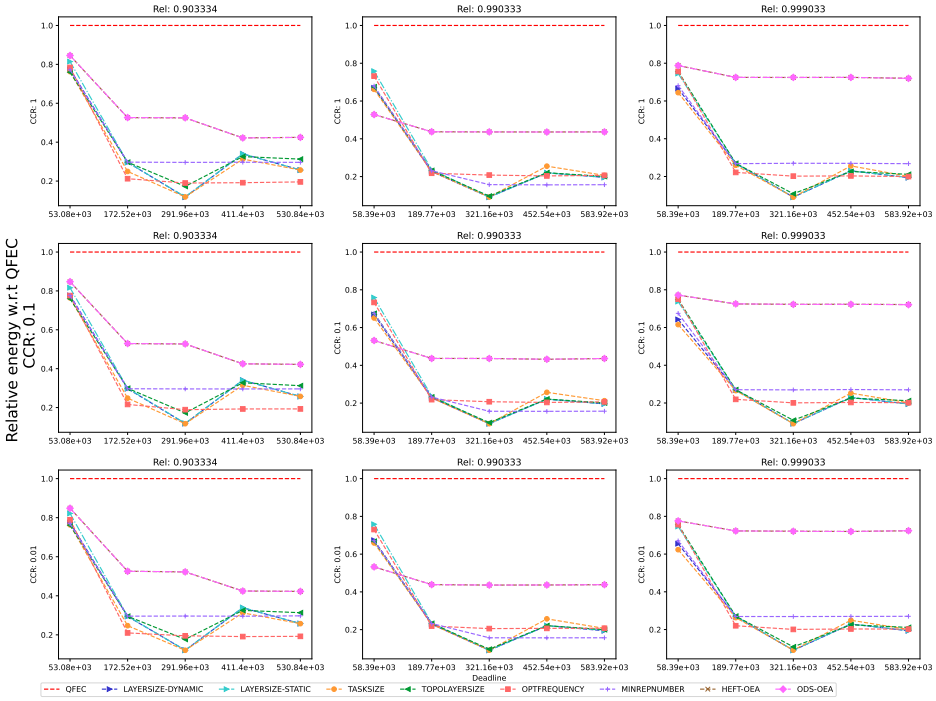

Figure 356: Performance of the different heuristics on the SoyKB workflow.

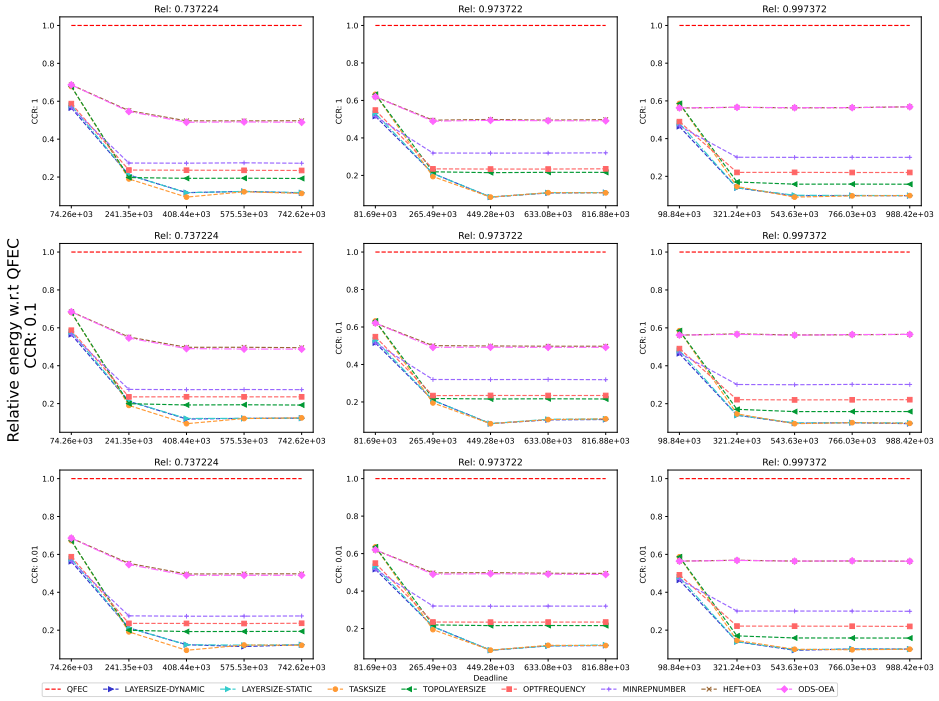

Figure 357: Performance of the different heuristics on the SRASearch.

D.8.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

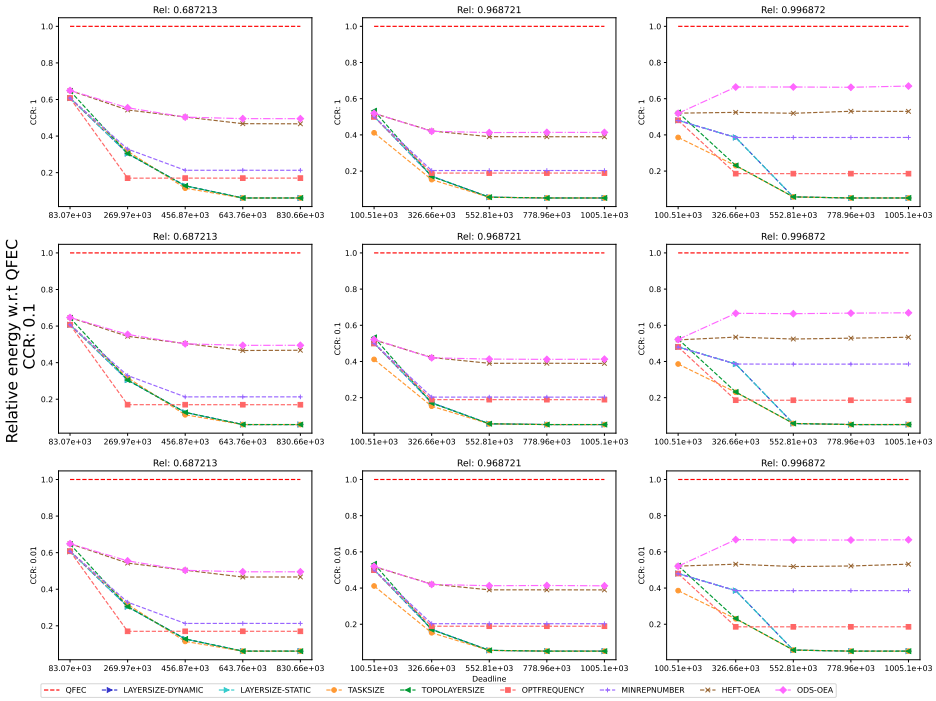

Figure 358: Performance of the different heuristics on the BLAST workflow.

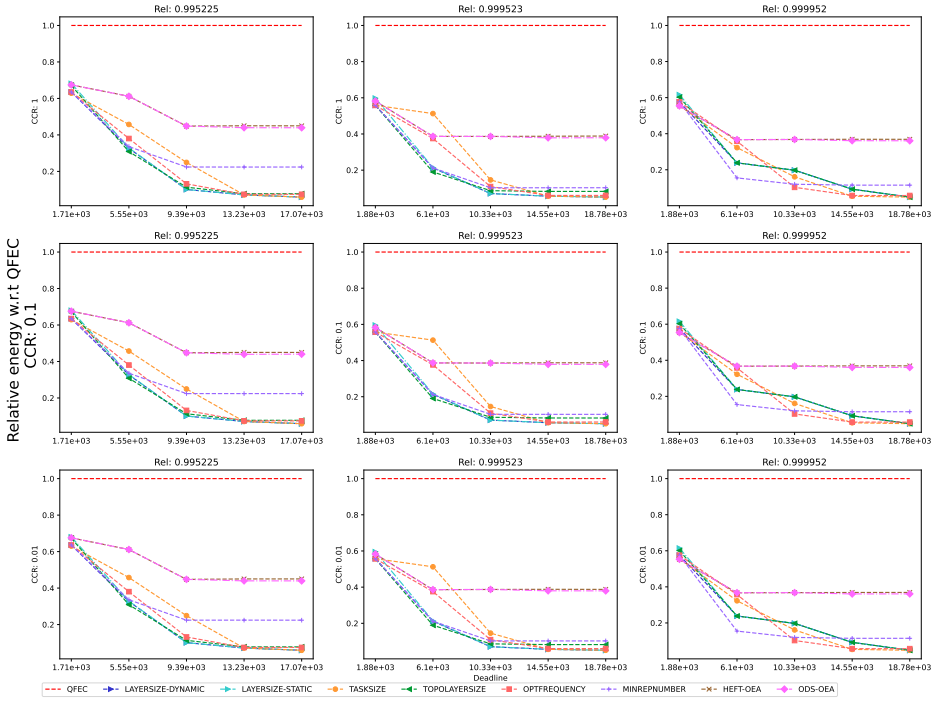

Figure 359: Performance of the different heuristics on the BWA workflow.

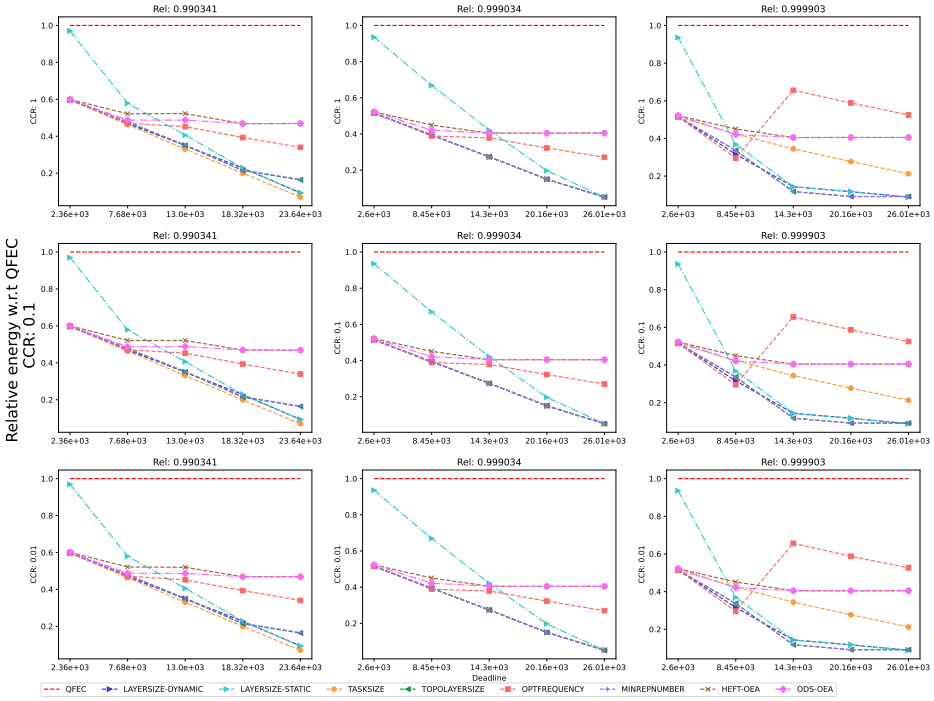

Figure 360: Performance of the different heuristics on the Cholesky workflow.

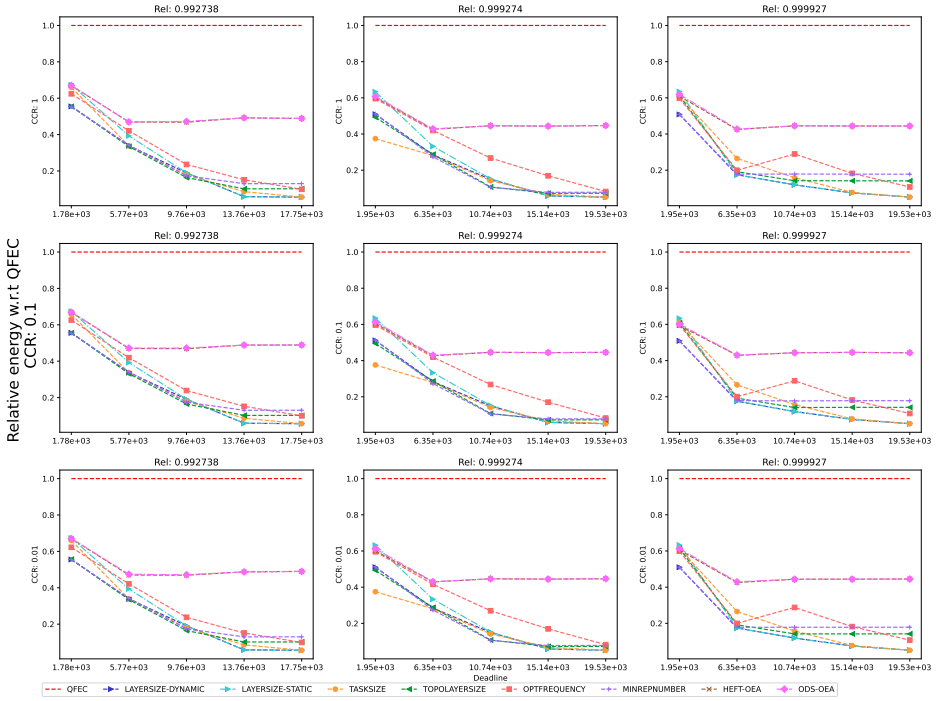

Figure 361: Performance of the different heuristics on the Cycles workflow.

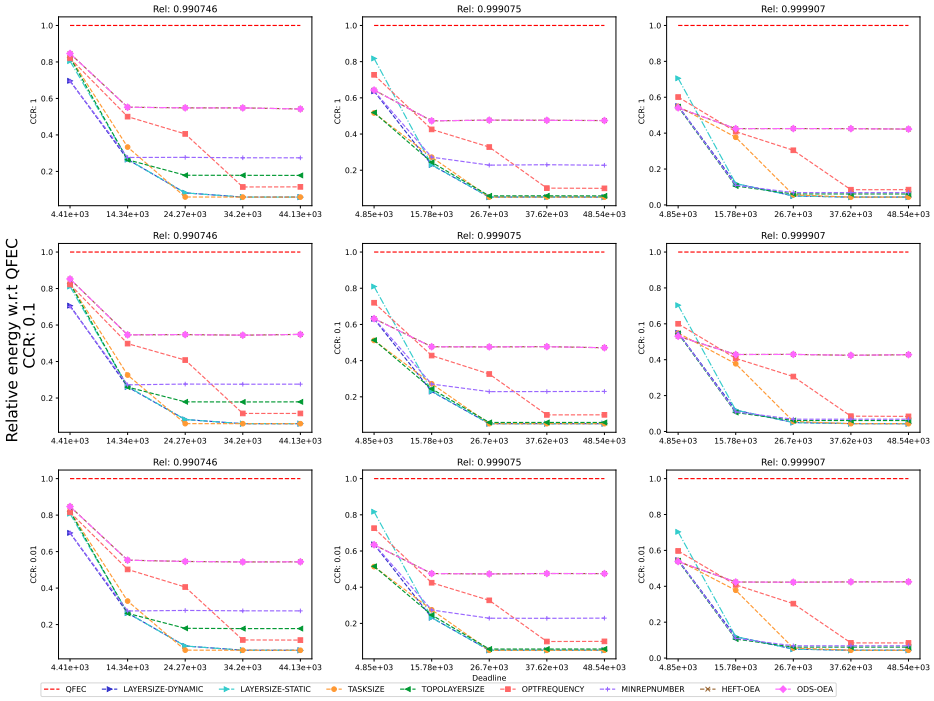

Figure 362: Performance of the different heuristics on the Epigenomics workflow.

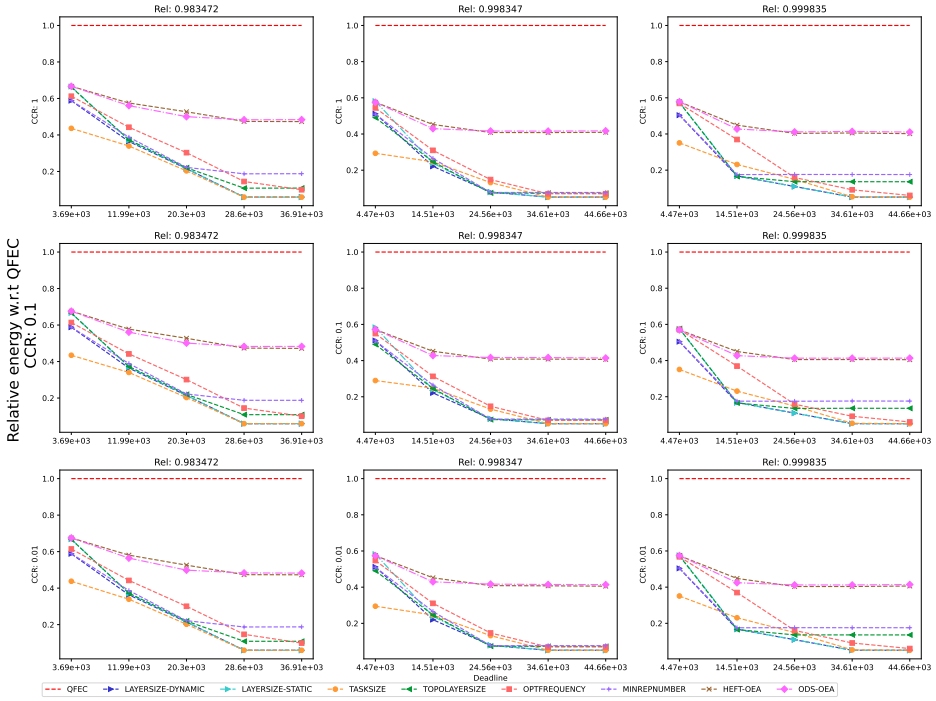

Figure 363: Performance of the different heuristics on the Genome workflow.

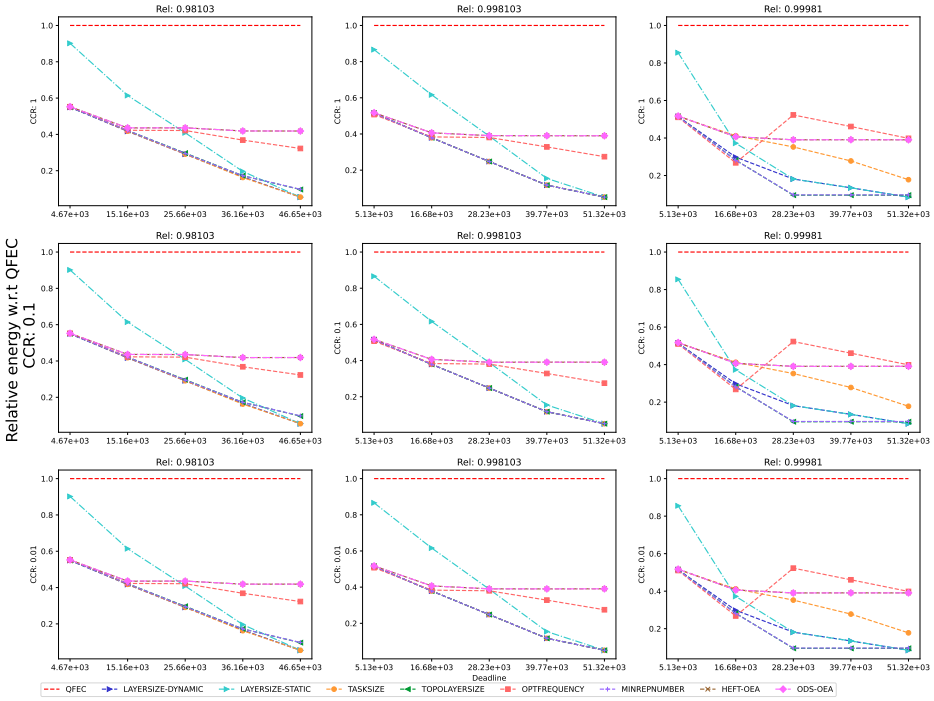

Figure 364: Performance of the different heuristics on the LU workflow.

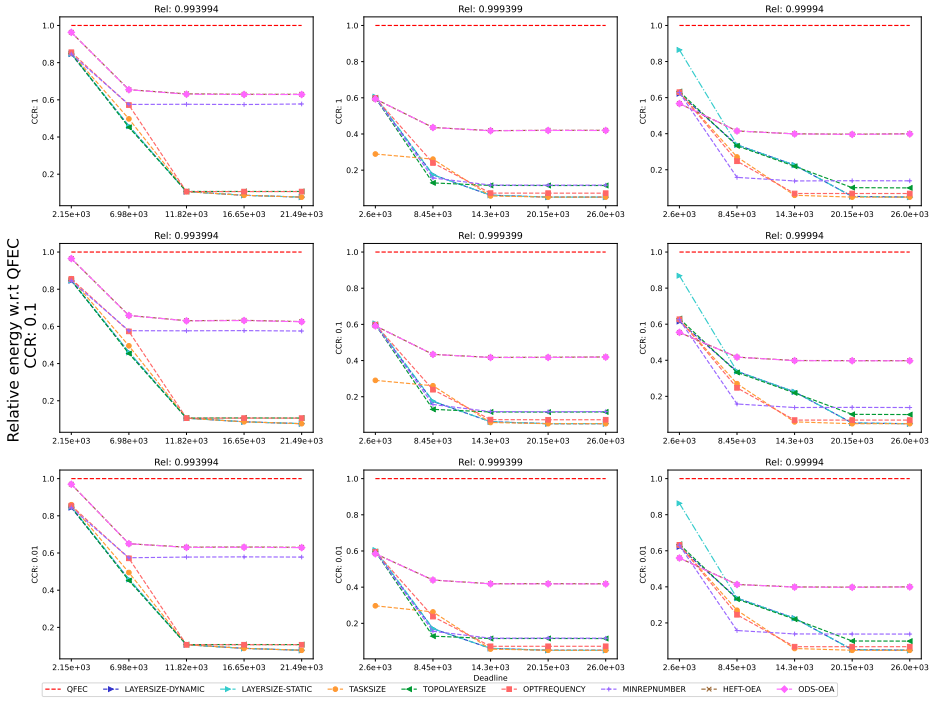

Figure 365: Performance of the different heuristics on the Montage workflow.

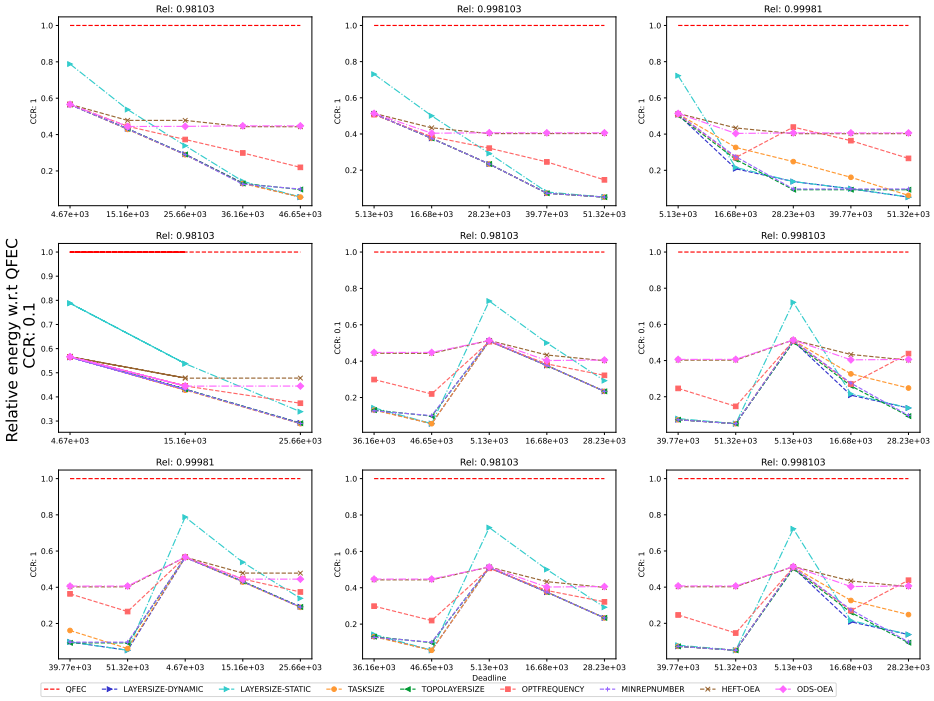

Figure 366: Performance of the different heuristics on the QR workflow.

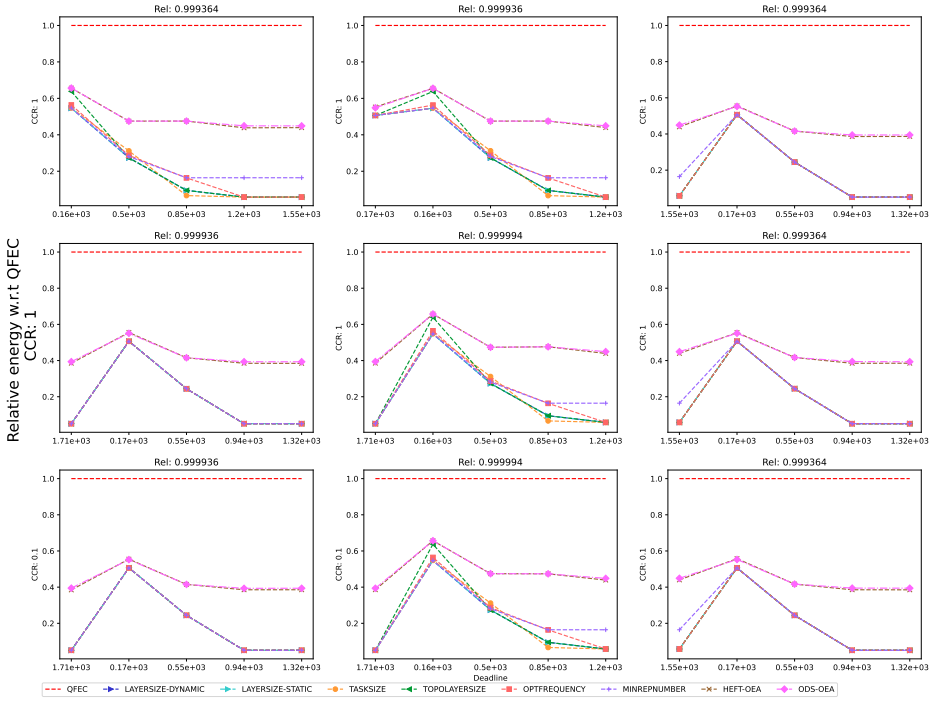

Figure 367: Performance of the different heuristics on the Seismology workflow.

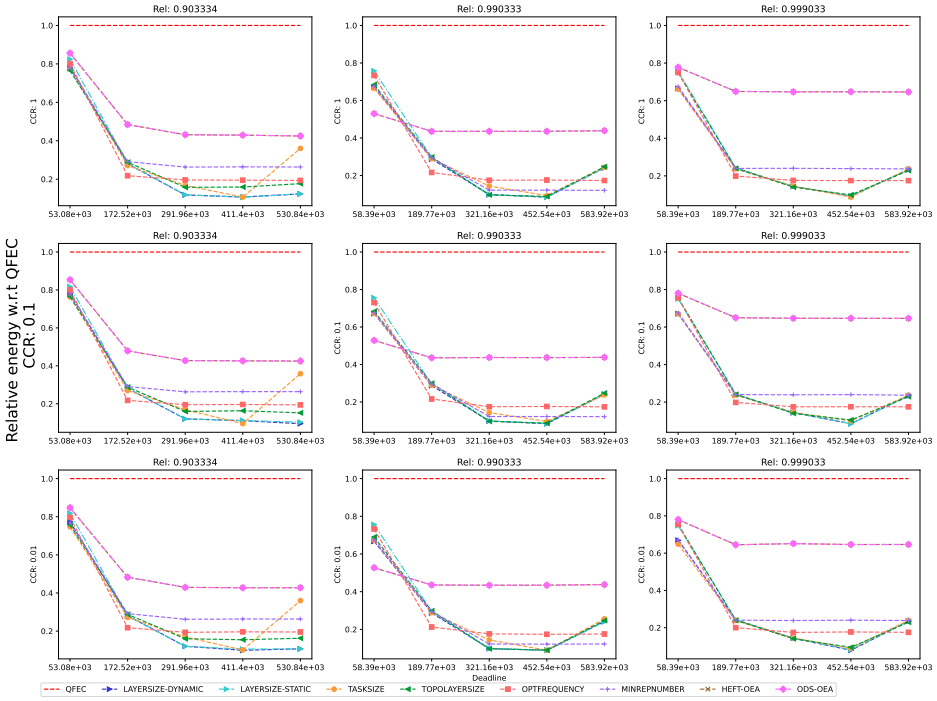

Figure 368: Performance of the different heuristics on the SoyKB workflow.

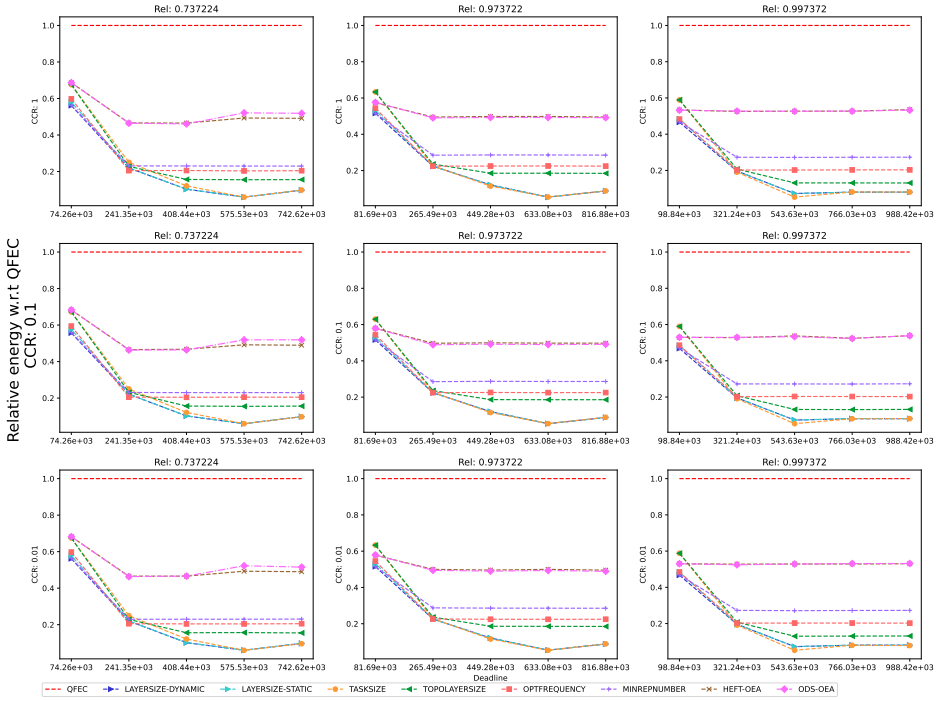

Figure 369: Performance of the different heuristics on the SRASearch.

D.8.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

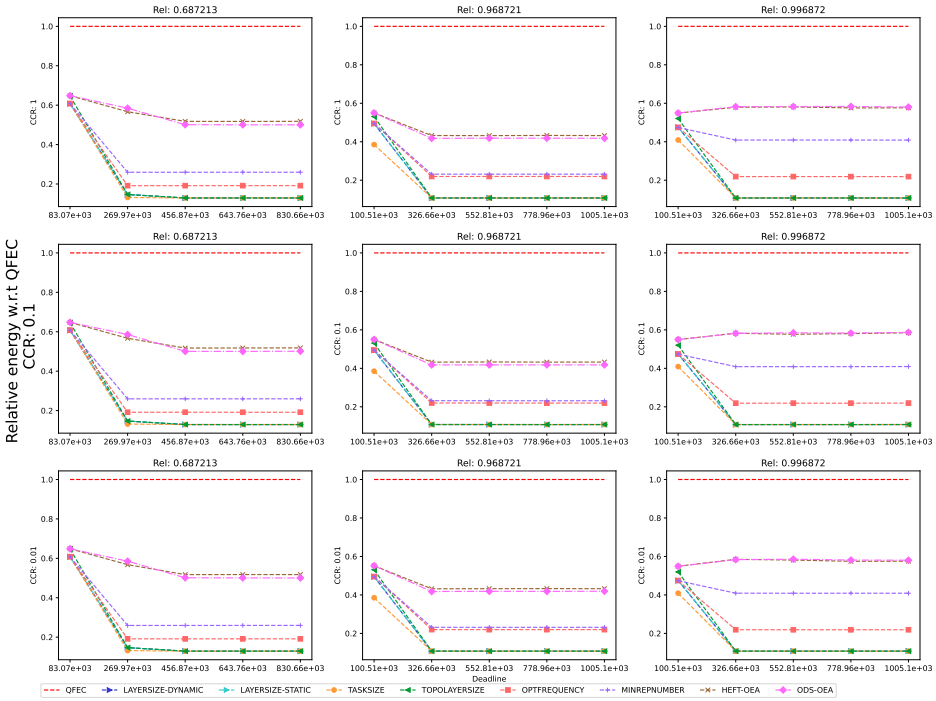

Figure 370: Performance of the different heuristics on the BLAST workflow.

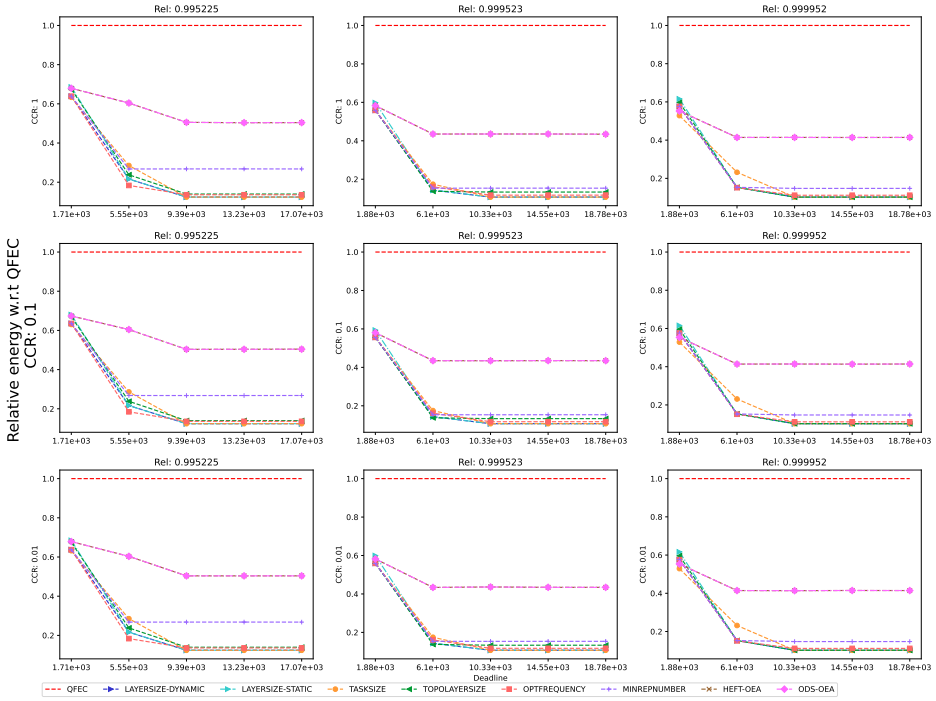

Figure 371: Performance of the different heuristics on the BWA workflow.

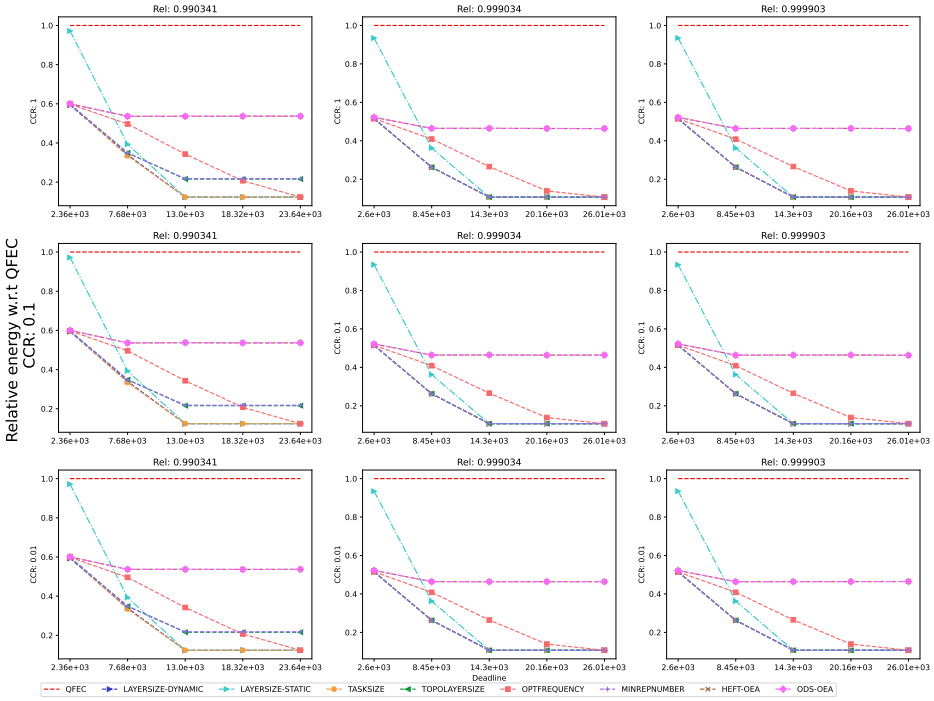

Figure 372: Performance of the different heuristics on the Cholesky workflow.

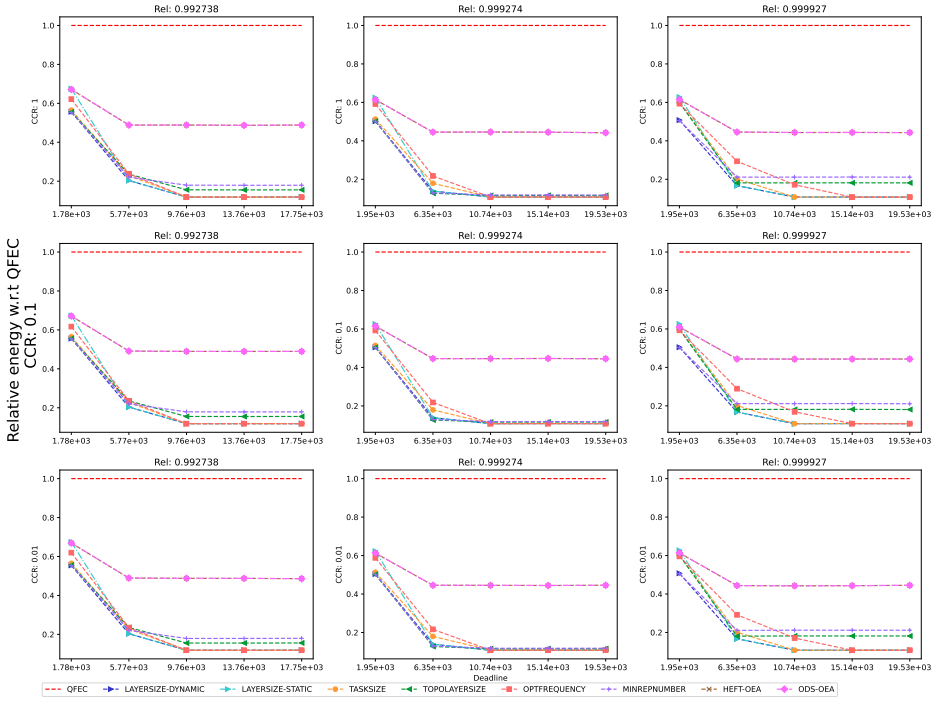

Figure 373: Performance of the different heuristics on the Cycles workflow.

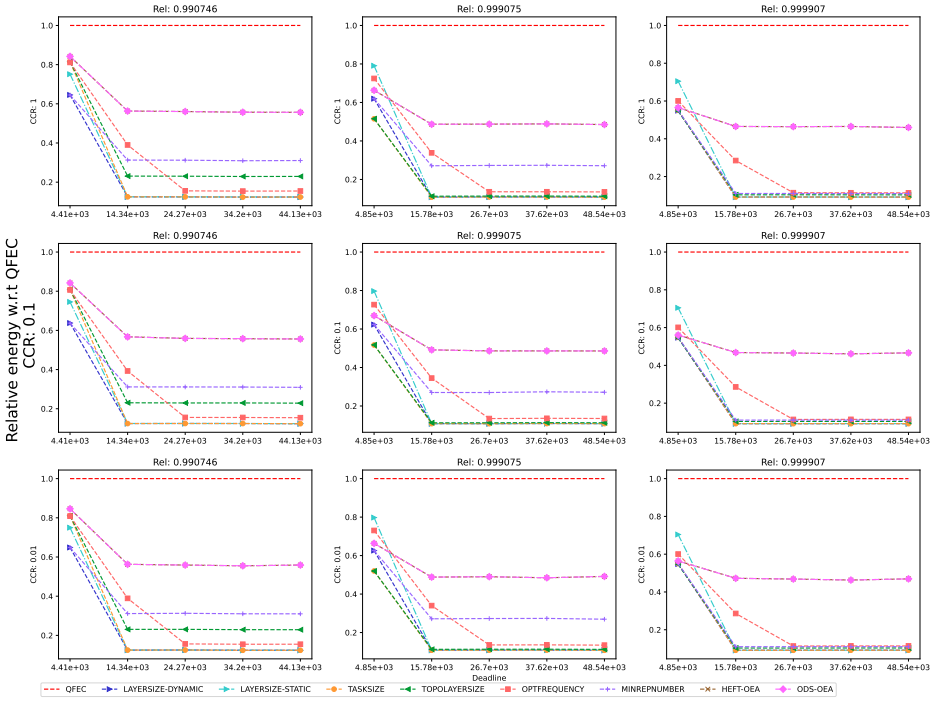

Figure 374: Performance of the different heuristics on the Epigenomics workflow.

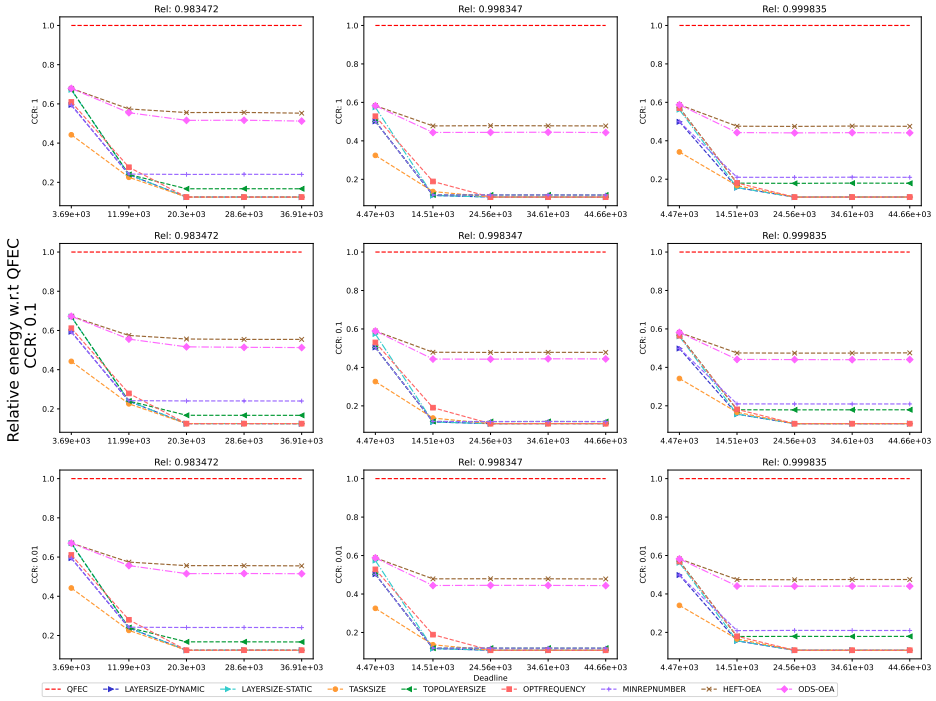

Figure 375: Performance of the different heuristics on the Genome workflow.

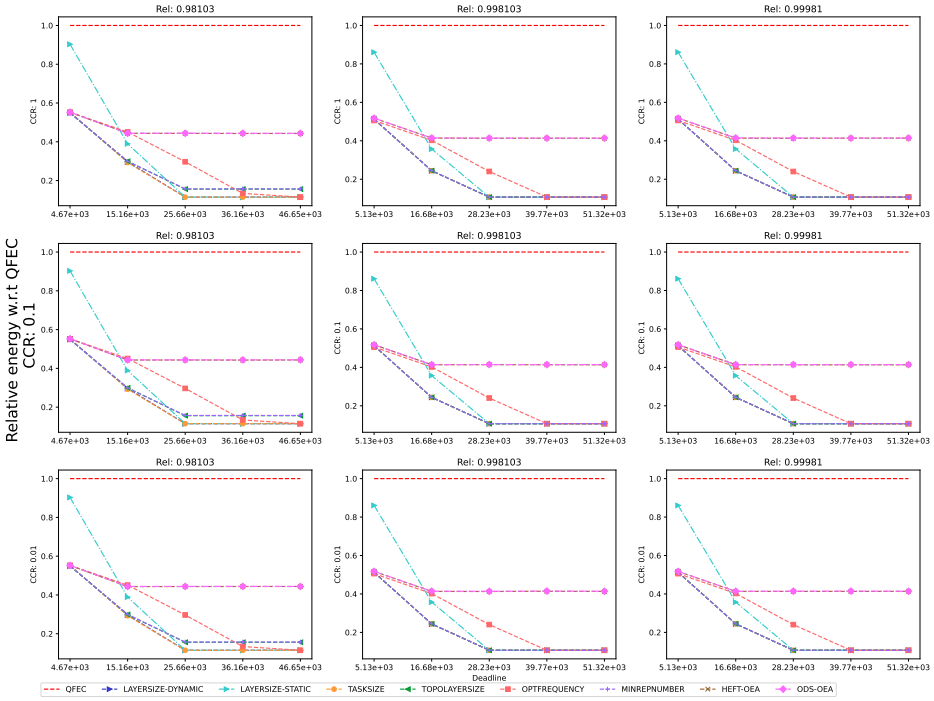

Figure 376: Performance of the different heuristics on the LU workflow.

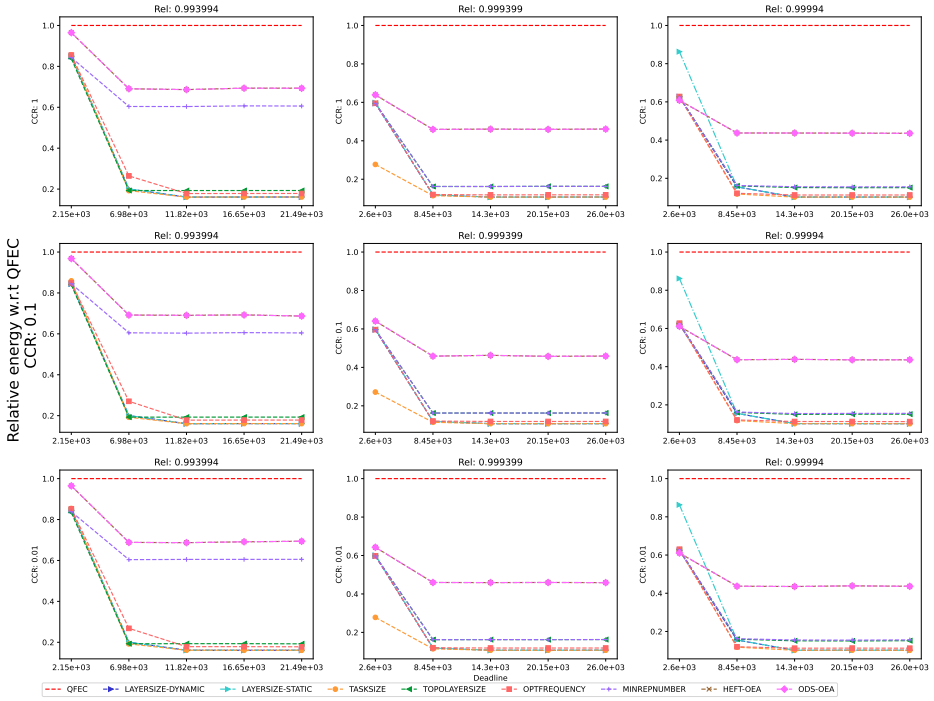

Figure 377: Performance of the different heuristics on the Montage workflow.

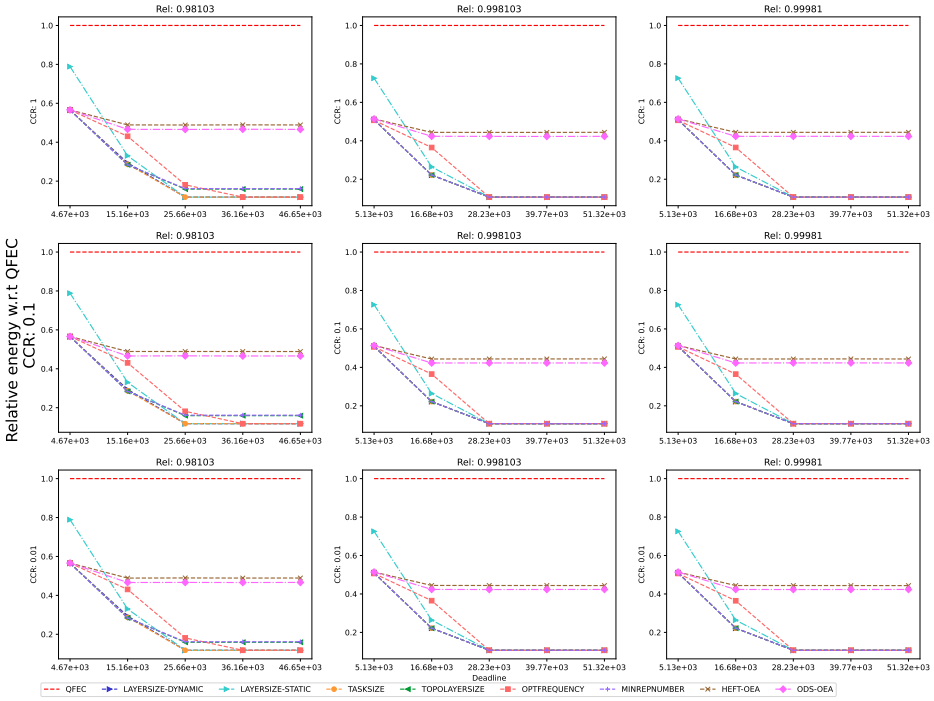

Figure 378: Performance of the different heuristics on the QR workflow.

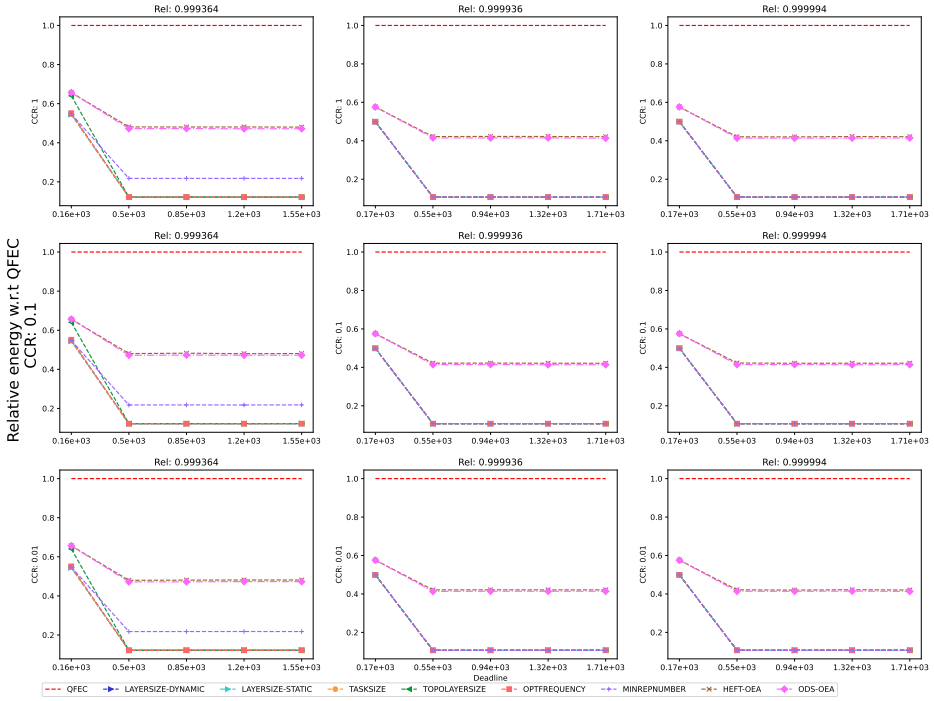

Figure 379: Performance of the different heuristics on the Seismology workflow.

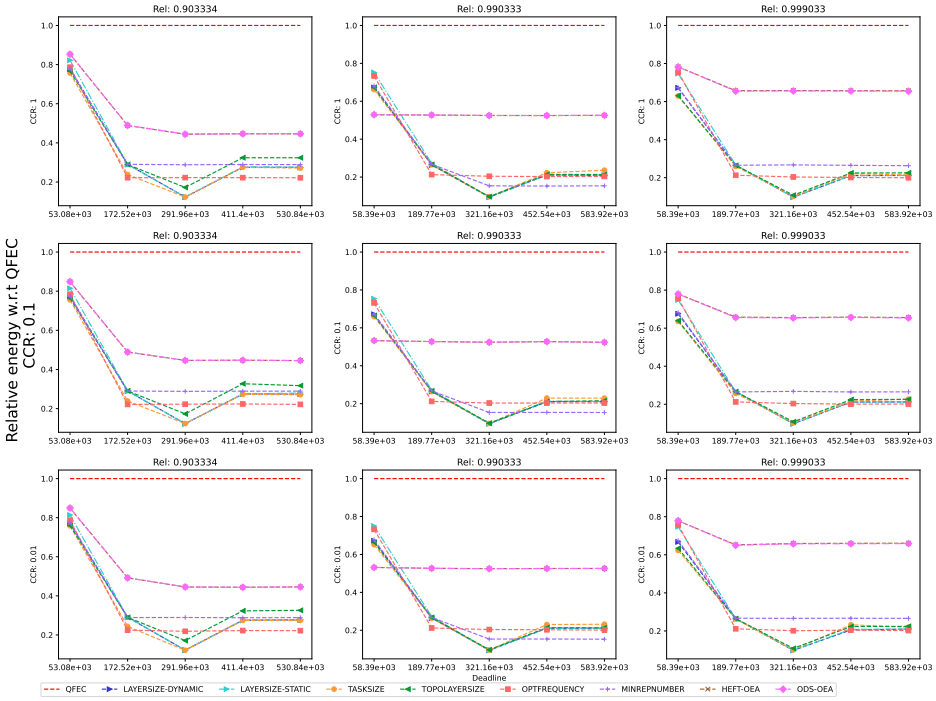

Figure 380: Performance of the different heuristics on the SoyKB workflow.

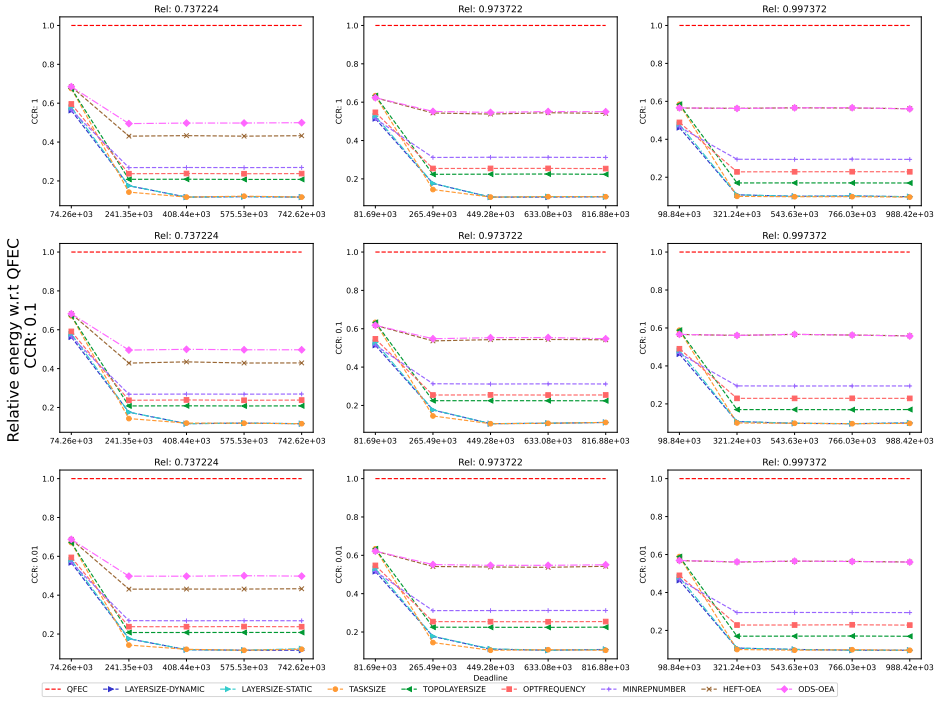

Figure 381: Performance of the different heuristics on the SRASearch.

D.8.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

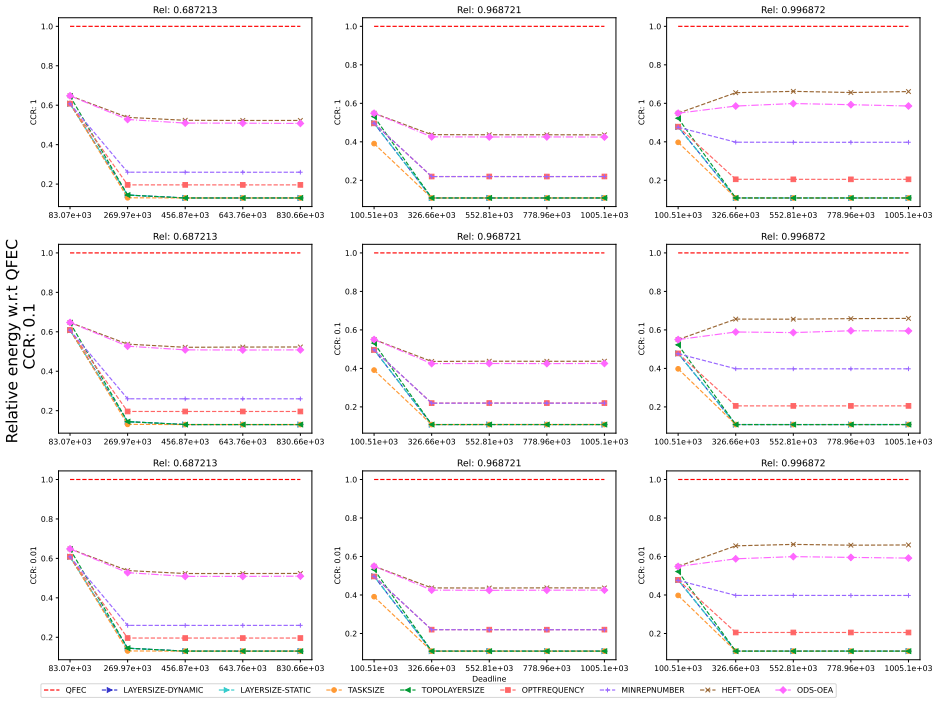

Figure 382: Performance of the different heuristics on the BLAST workflow.

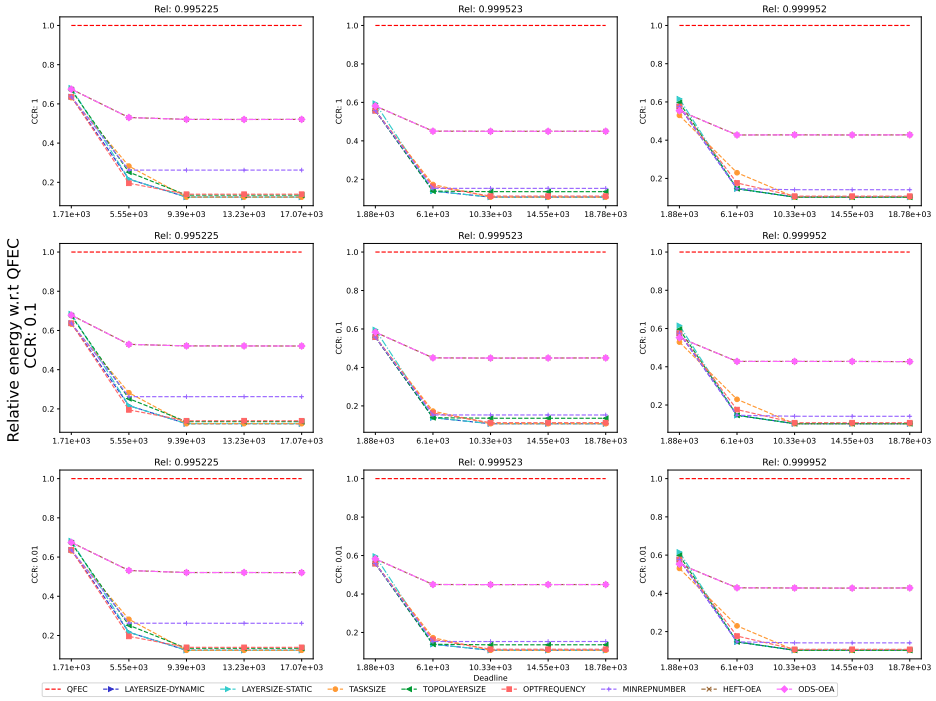

Figure 383: Performance of the different heuristics on the BWA workflow.

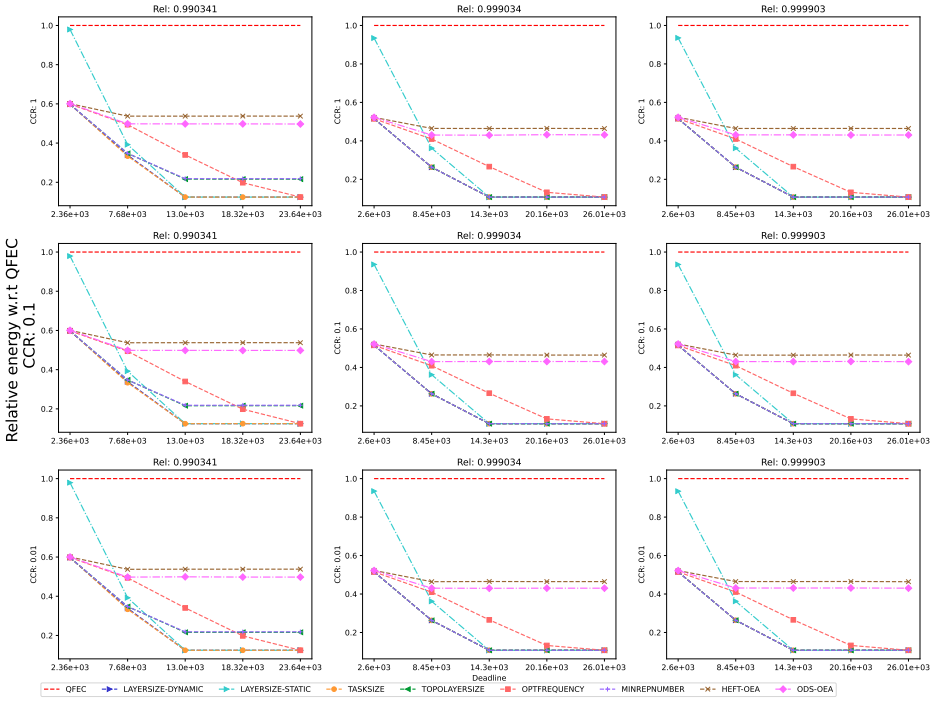

Figure 384: Performance of the different heuristics on the Cholesky workflow.

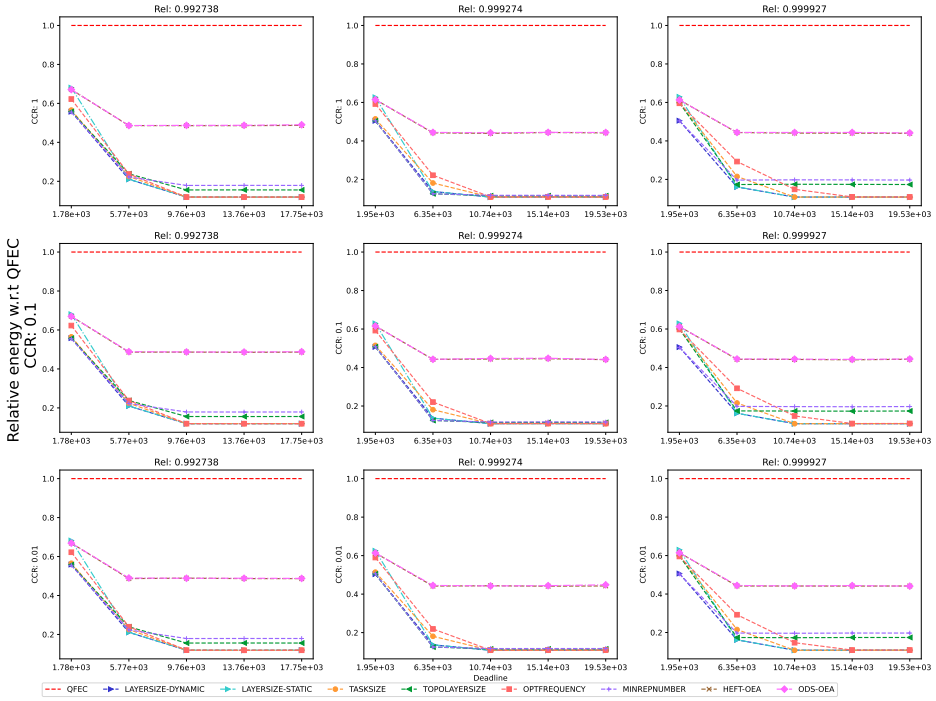

Figure 385: Performance of the different heuristics on the Cycles workflow.

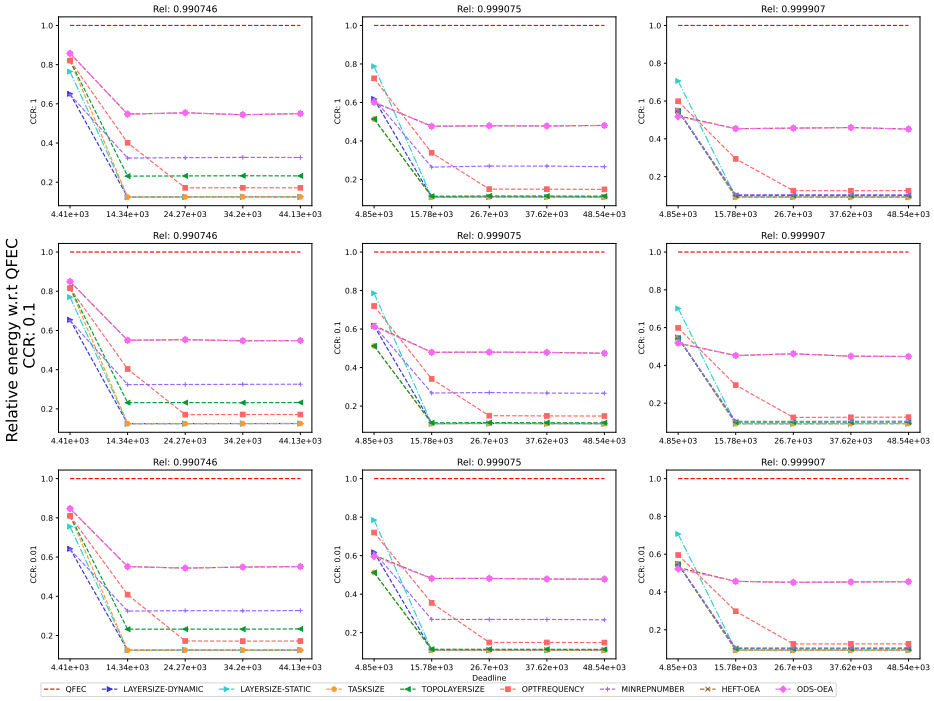

Figure 386: Performance of the different heuristics on the Epigenomics workflow.

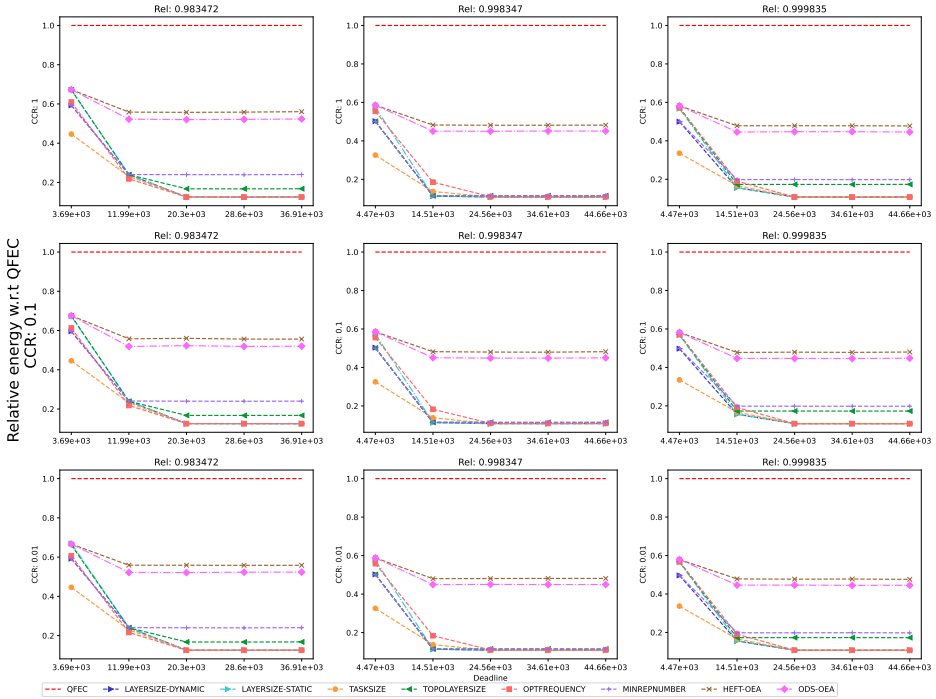

Figure 387: Performance of the different heuristics on the Genome workflow.

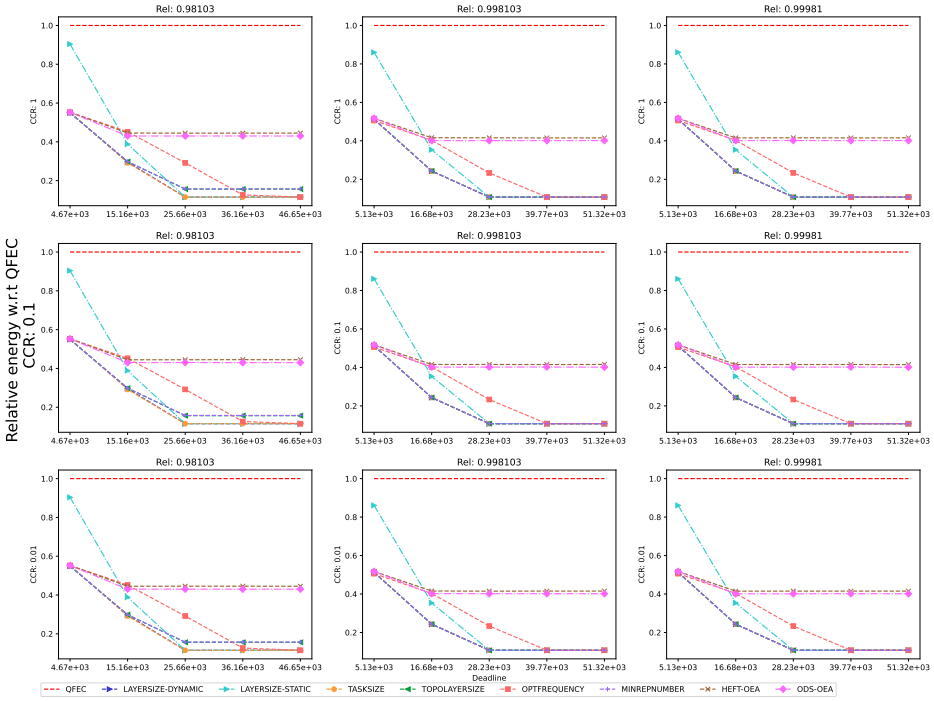

Figure 388: Performance of the different heuristics on the LU workflow.

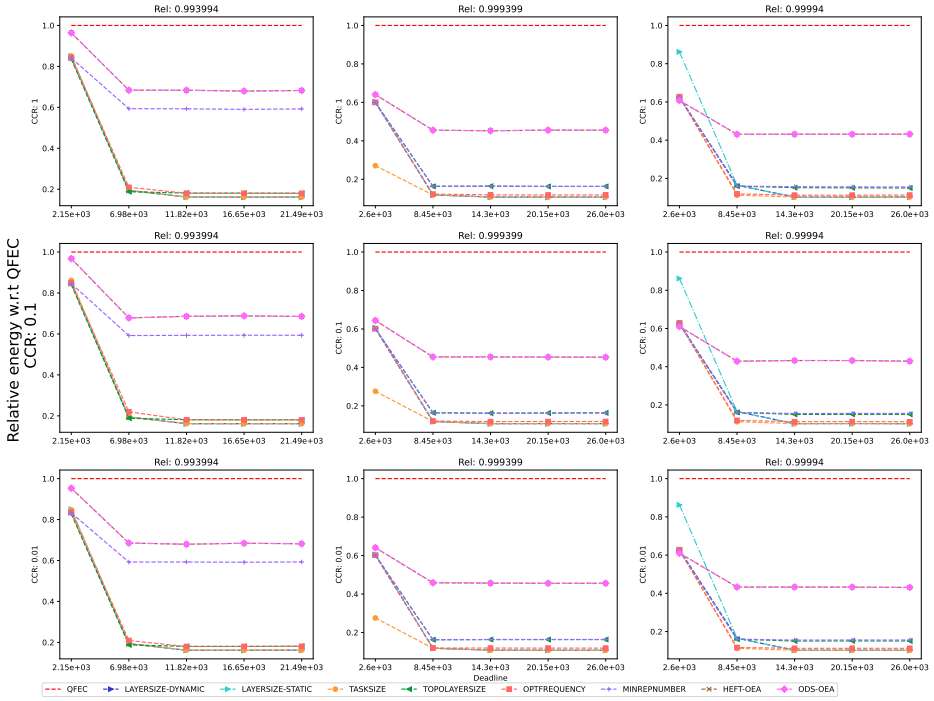

Figure 389: Performance of the different heuristics on the Montage workflow.

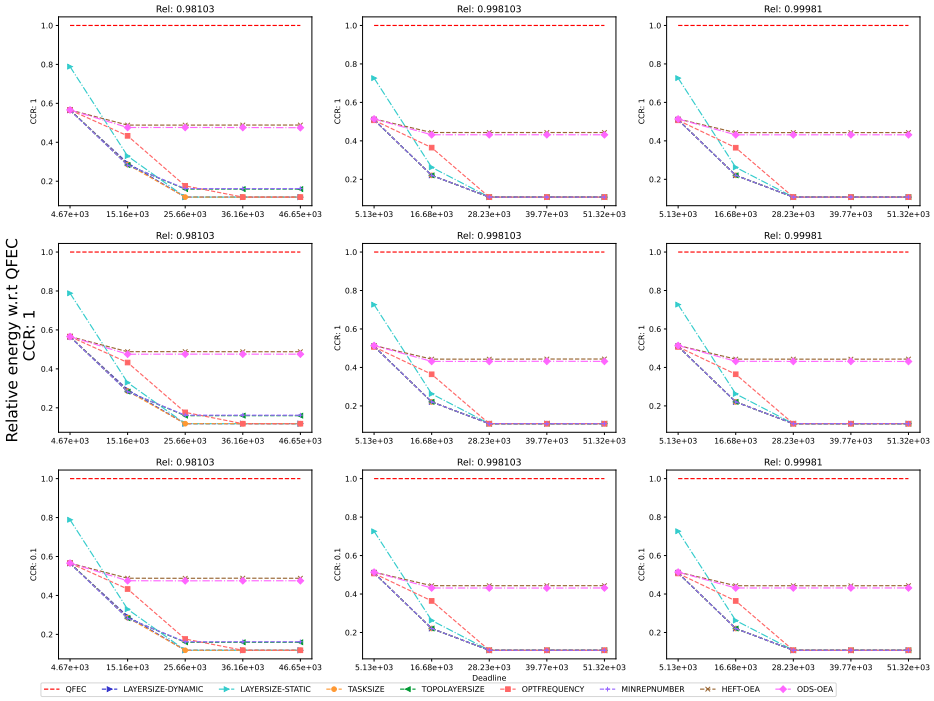

Figure 390: Performance of the different heuristics on the QR workflow.

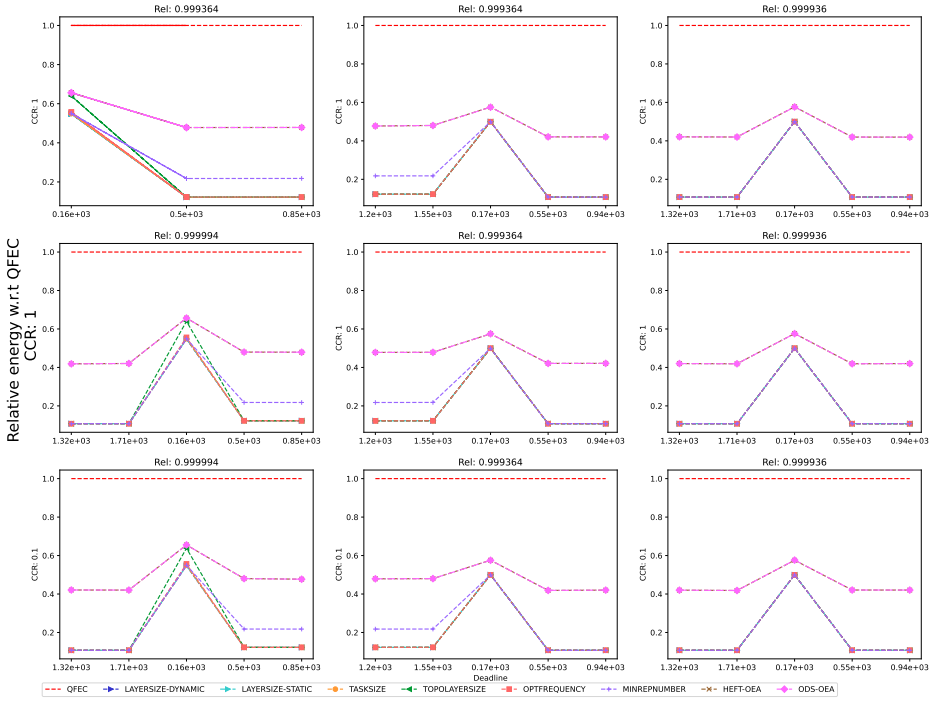

Figure 391: Performance of the different heuristics on the Seismology workflow.

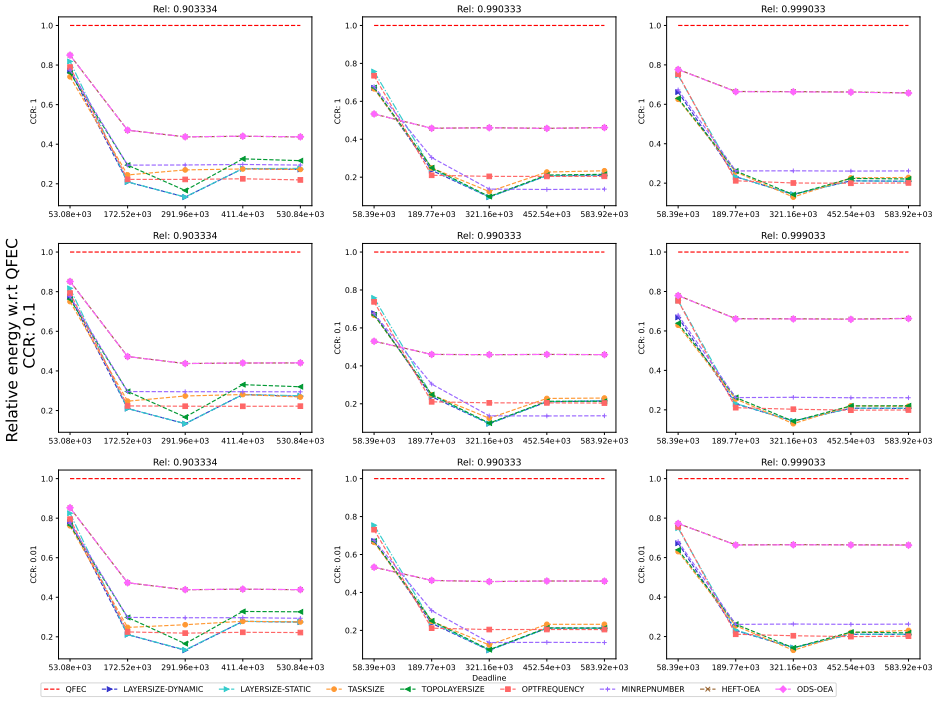

Figure 392: Performance of the different heuristics on the SoyKB workflow.

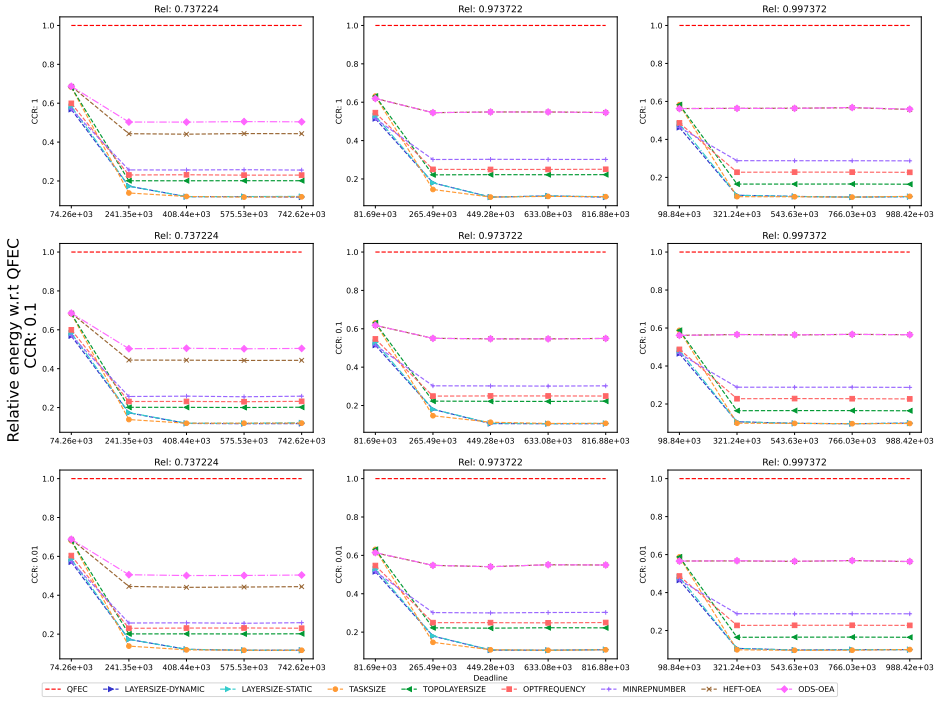

Figure 393: Performance of the different heuristics on the SRASearch.

D.9 $BC/WC = 0.9$

D.9.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

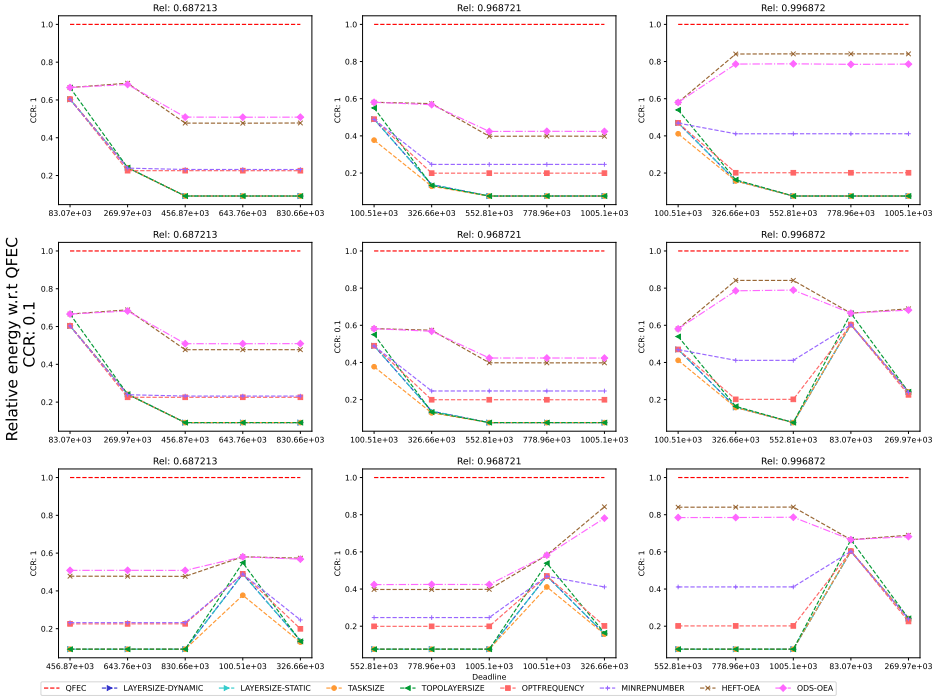

Figure 394: Performance of the different heuristics on the BLAST workflow.

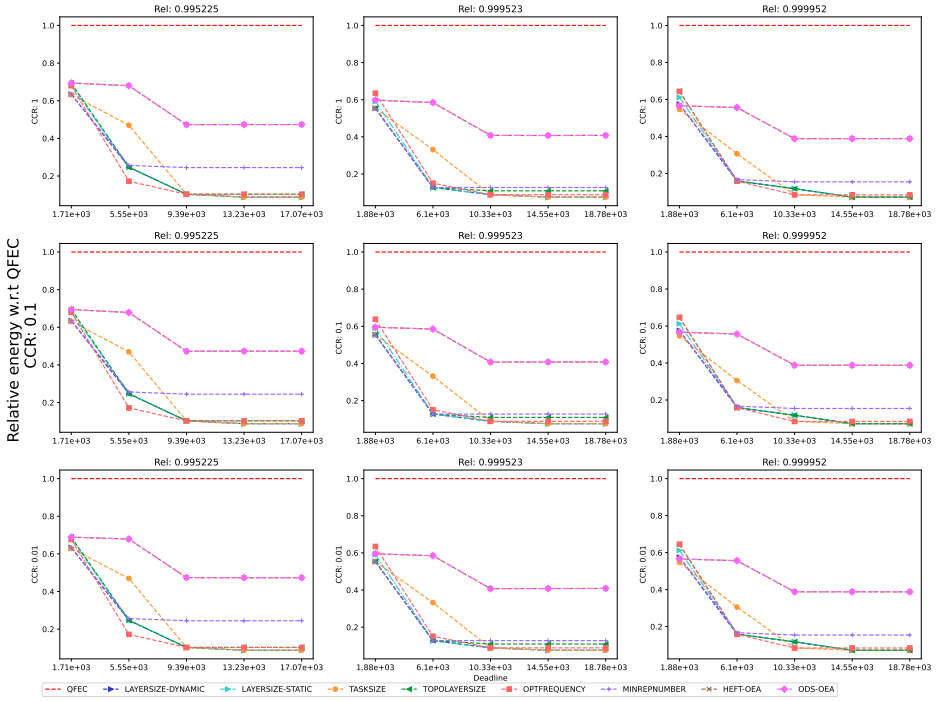

Figure 395: Performance of the different heuristics on the BWA workflow.

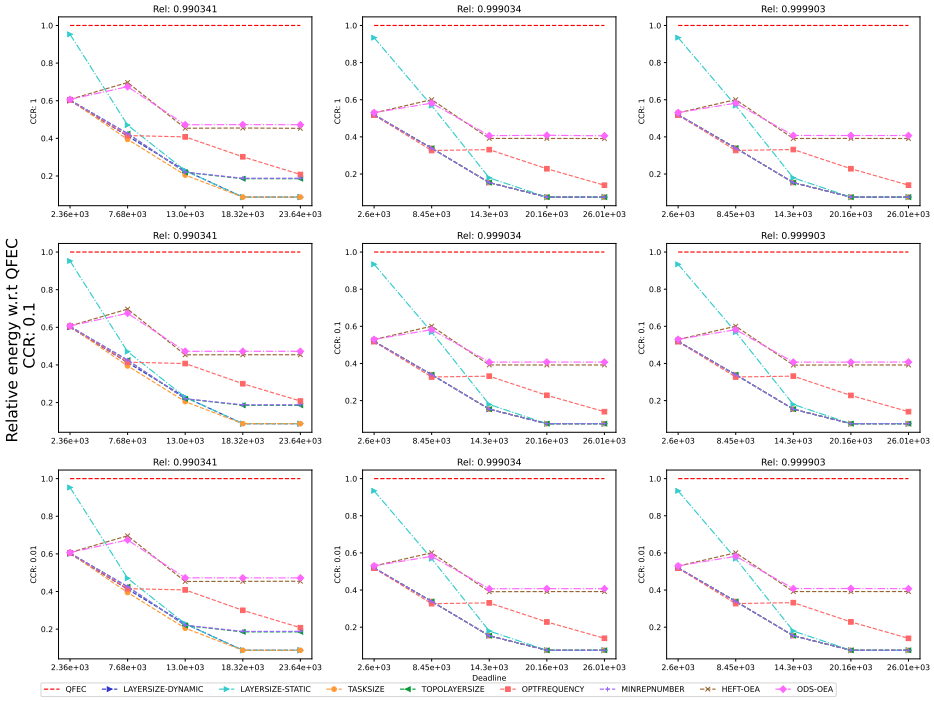

Figure 396: Performance of the different heuristics on the Cholesky workflow.

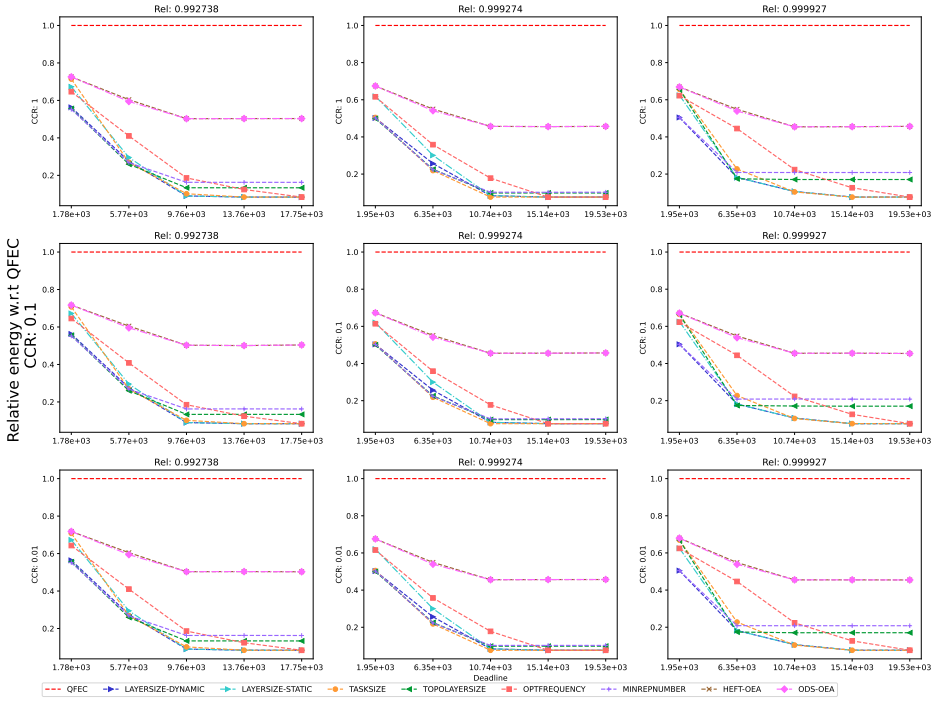

Figure 397: Performance of the different heuristics on the Cycles workflow.

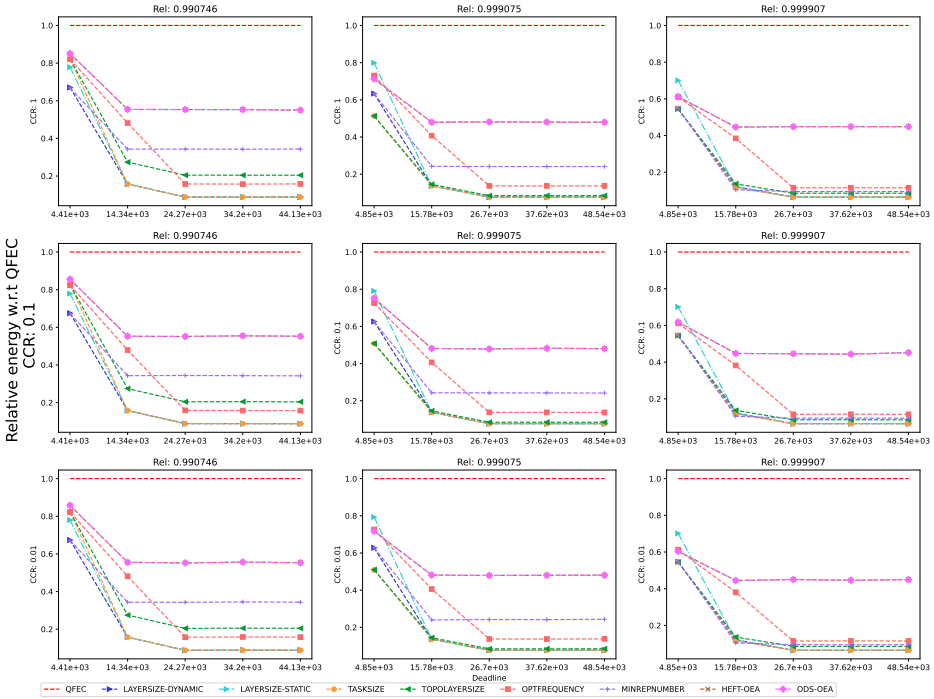

Figure 398: Performance of the different heuristics on the Epigenomics workflow.

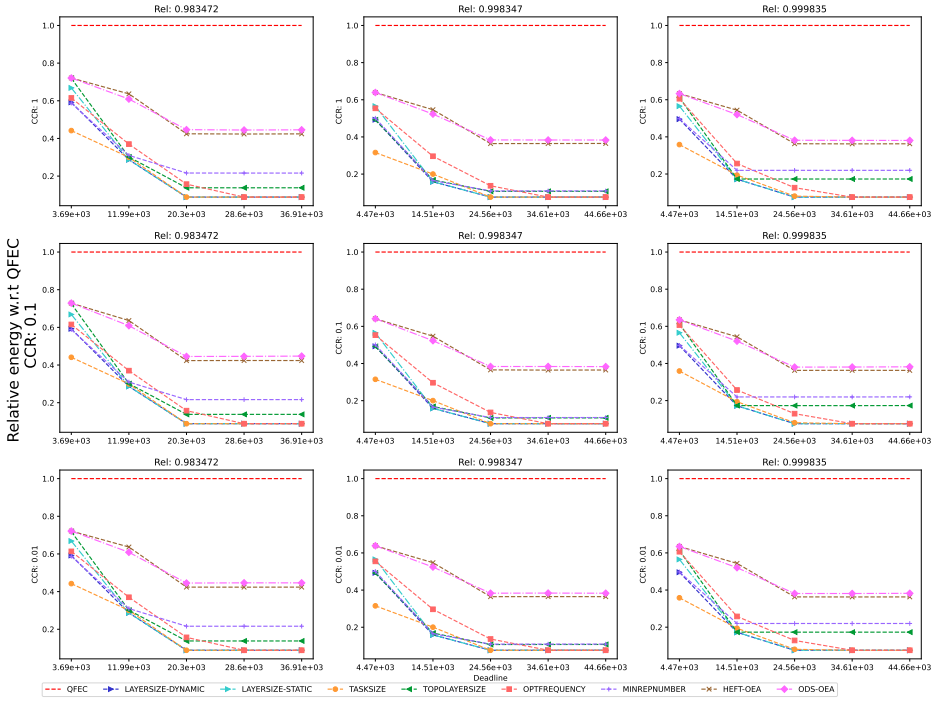

Figure 399: Performance of the different heuristics on the Genome workflow.

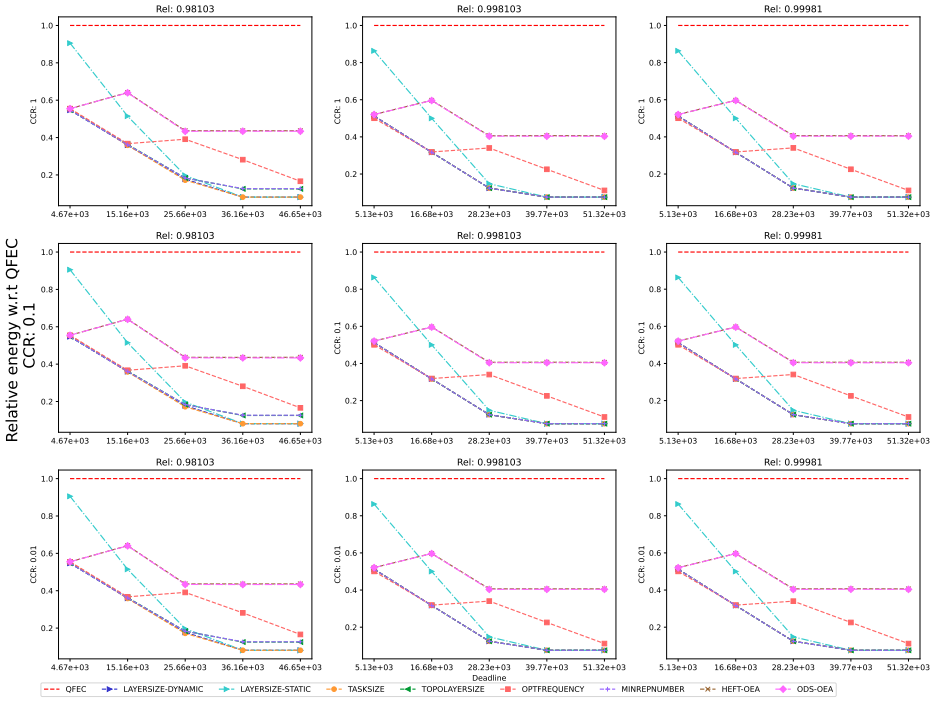

Figure 400: Performance of the different heuristics on the LU workflow.

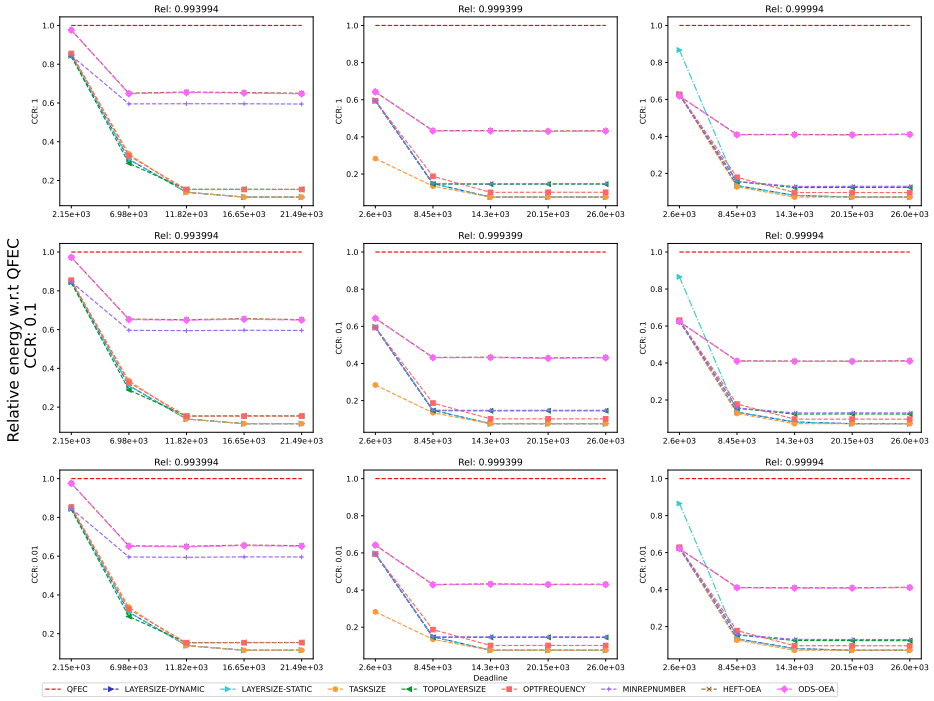

Figure 401: Performance of the different heuristics on the Montage workflow.

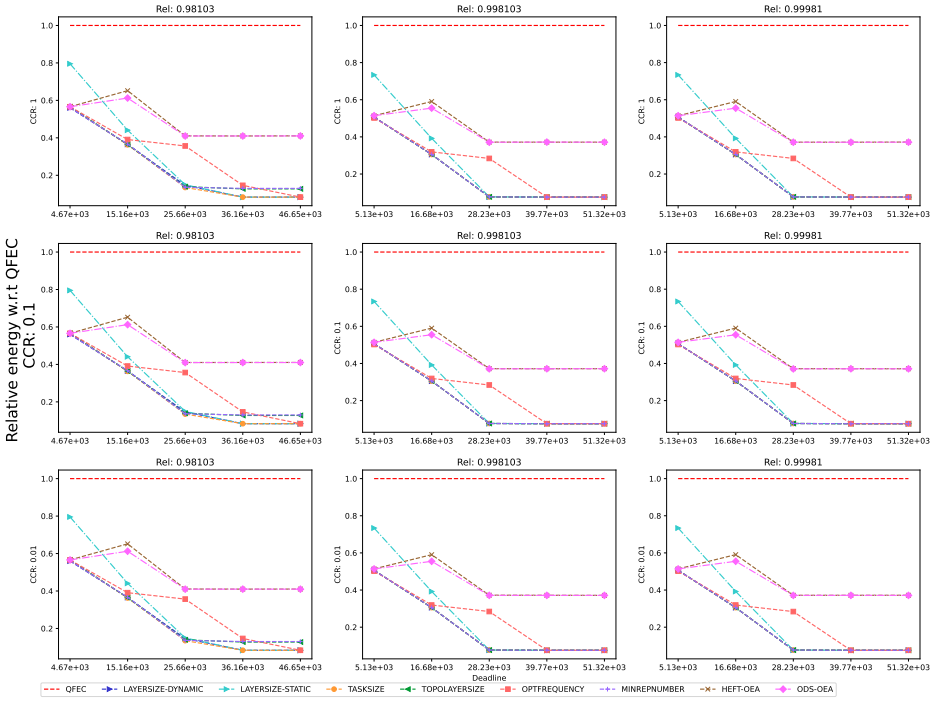

Figure 402: Performance of the different heuristics on the QR workflow.

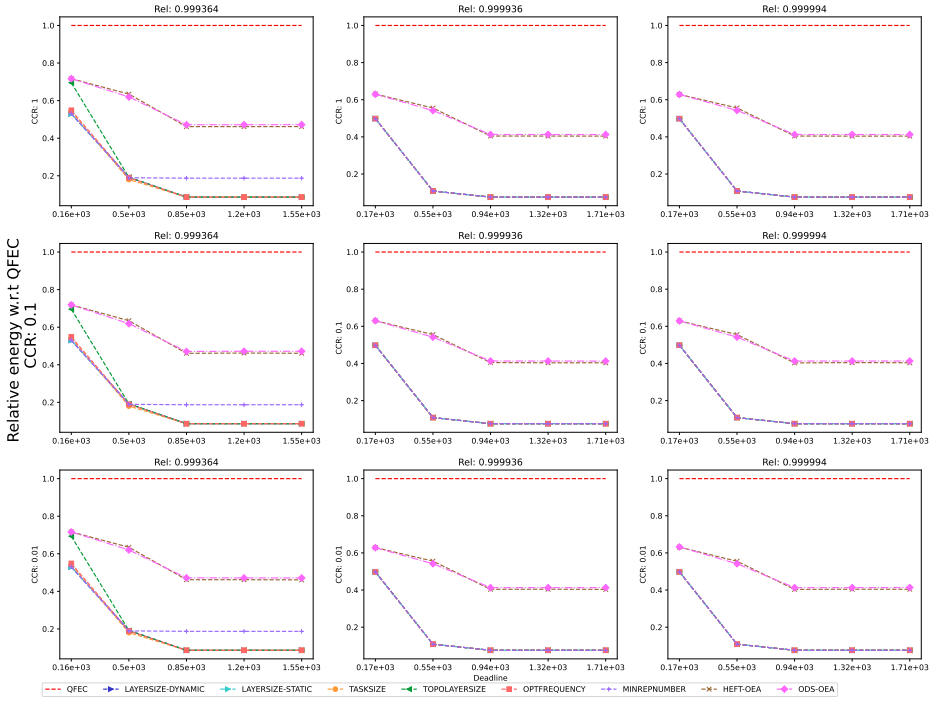

Figure 403: Performance of the different heuristics on the Seismology workflow.

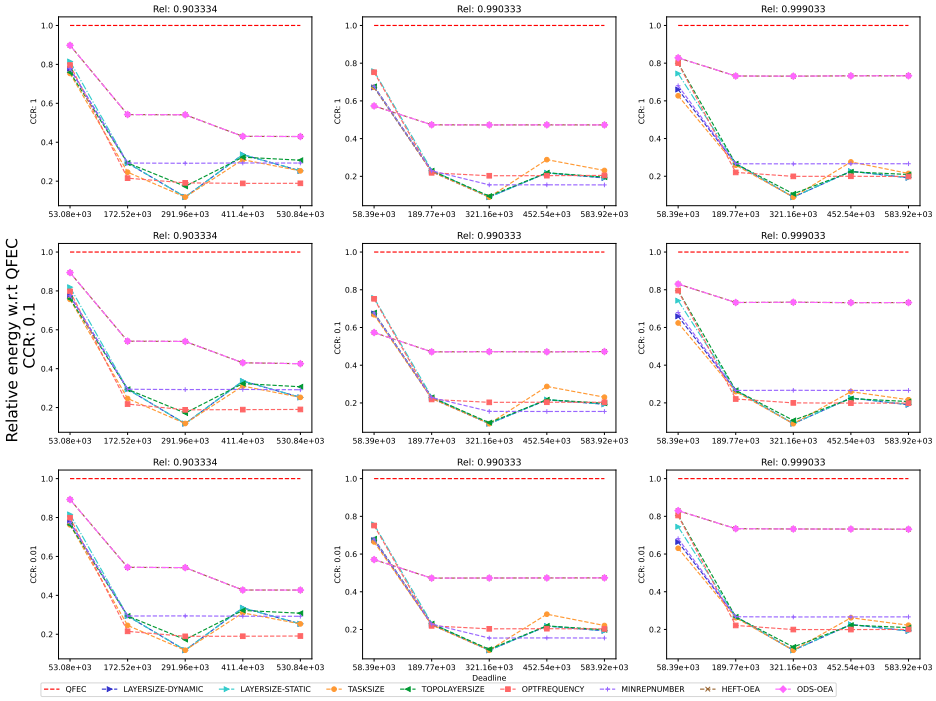

Figure 404: Performance of the different heuristics on the SoyKB workflow.

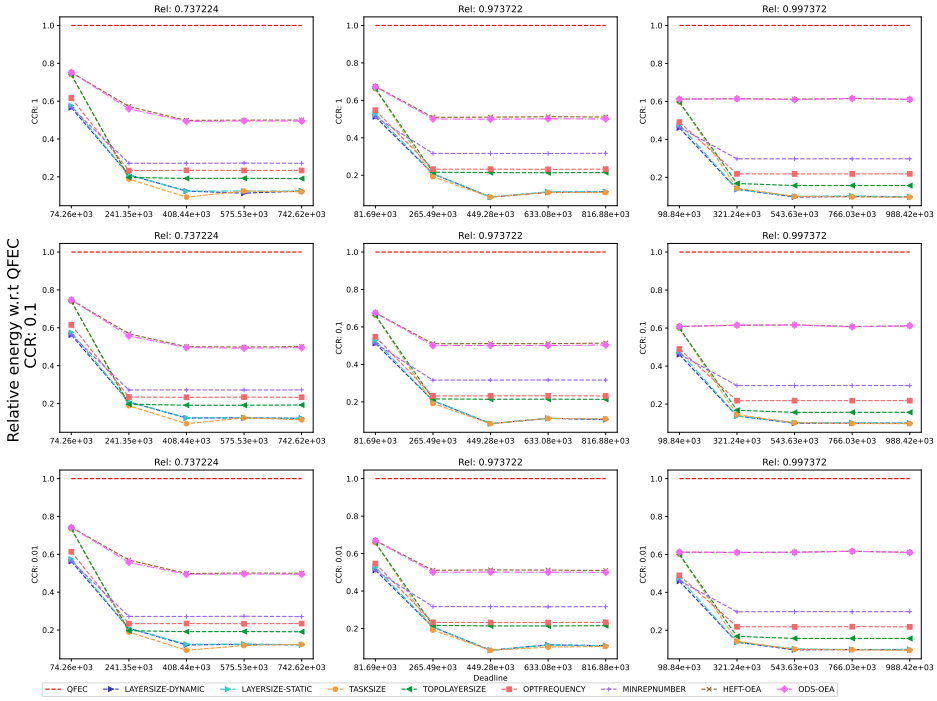

Figure 405: Performance of the different heuristics on the SRASearch.

D.9.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

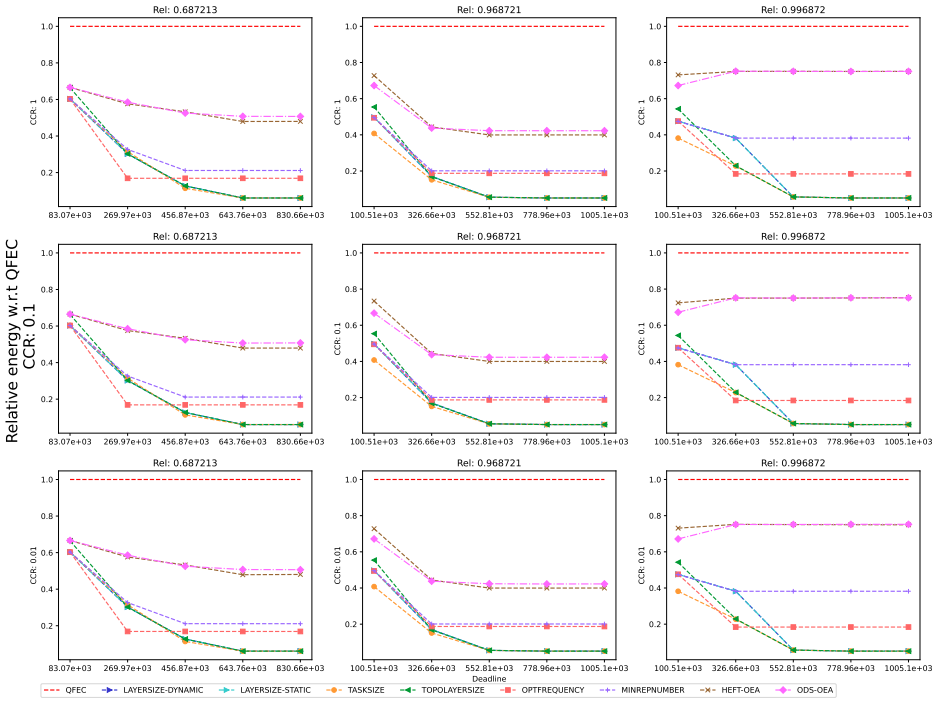

Figure 406: Performance of the different heuristics on the BLAST workflow.

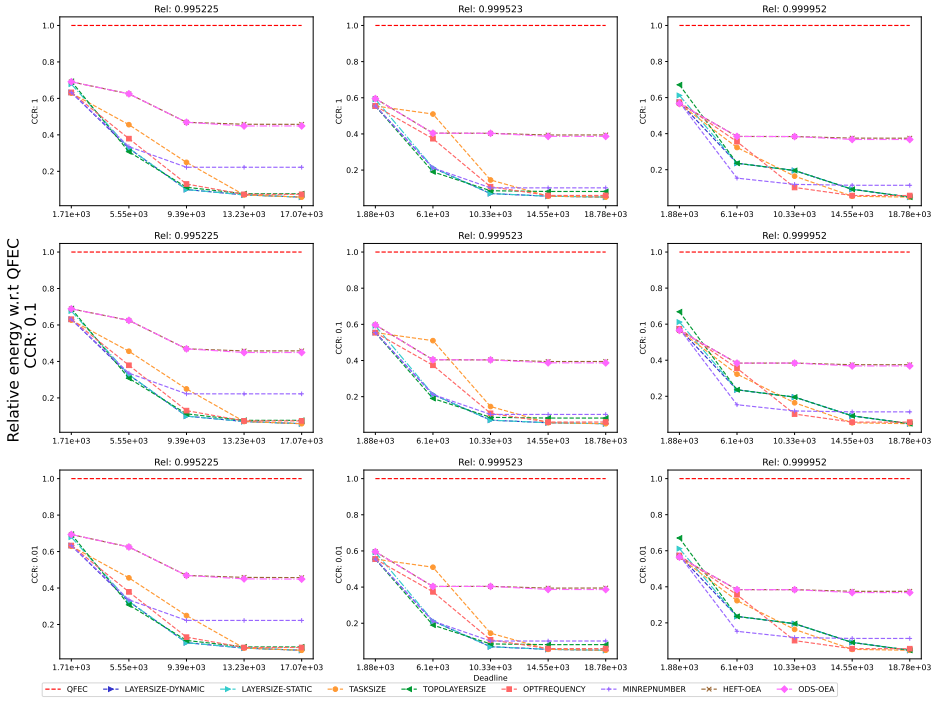

Figure 407: Performance of the different heuristics on the BWA workflow.

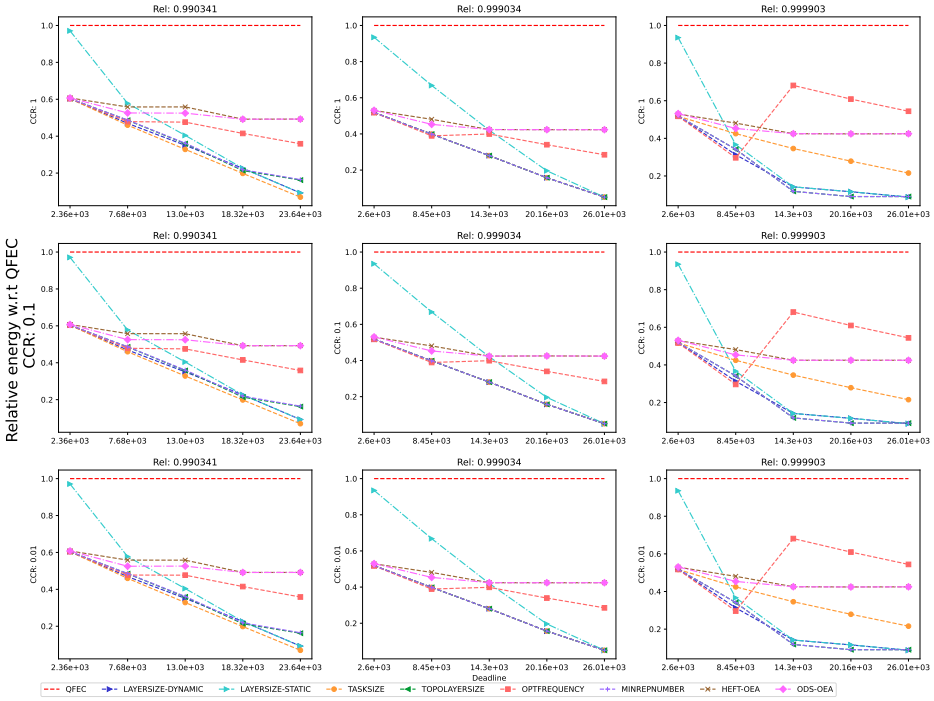

Figure 408: Performance of the different heuristics on the Cholesky workflow.

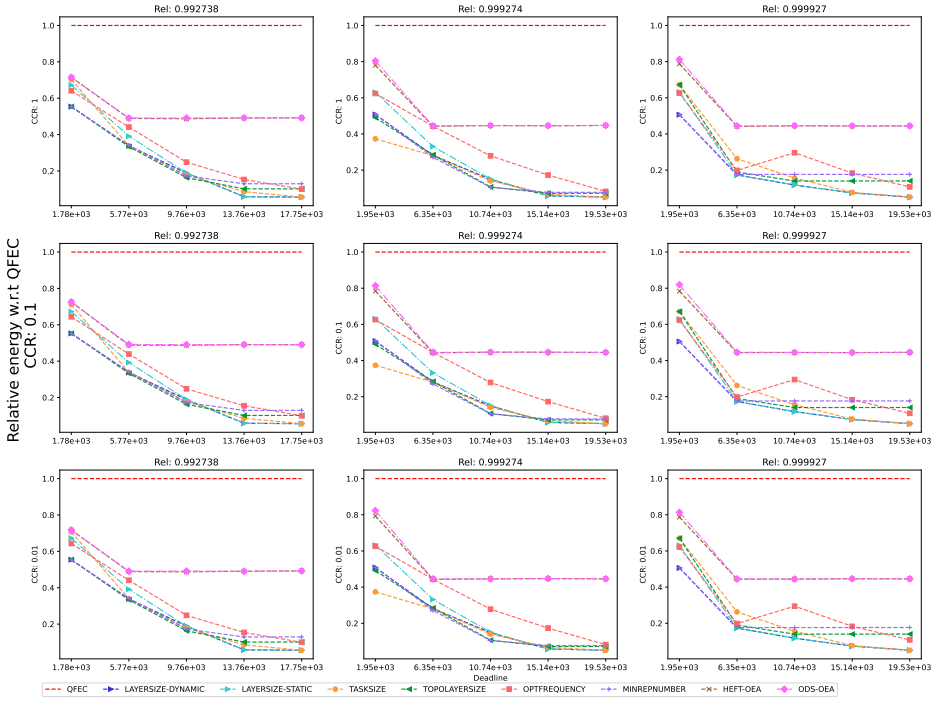

Figure 409: Performance of the different heuristics on the Cycles workflow.

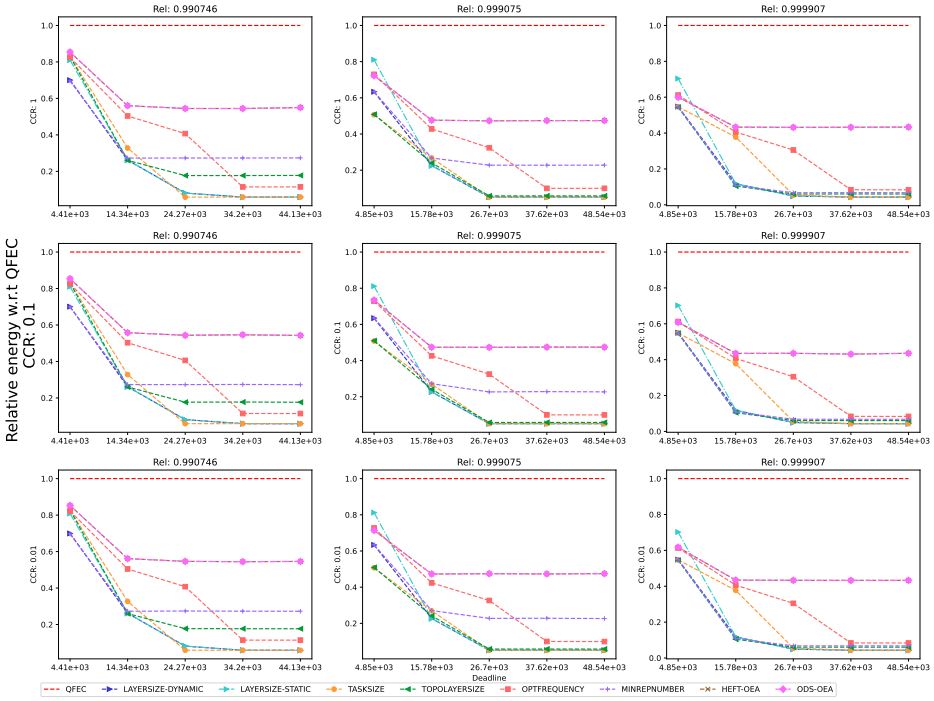

Figure 410: Performance of the different heuristics on the Epigenomics workflow.

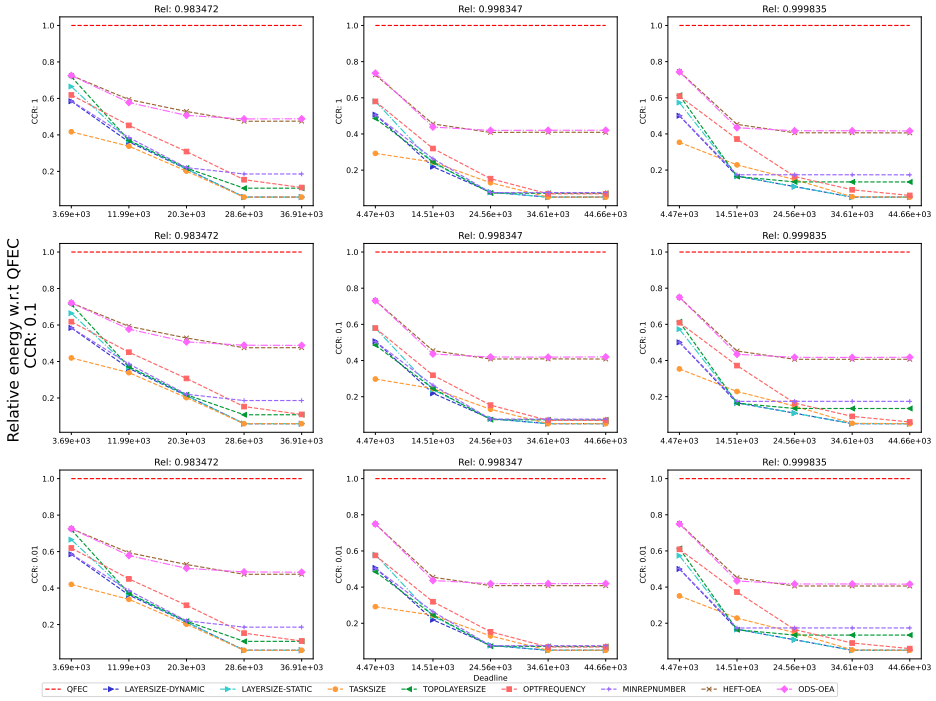

Figure 411: Performance of the different heuristics on the Genome workflow.

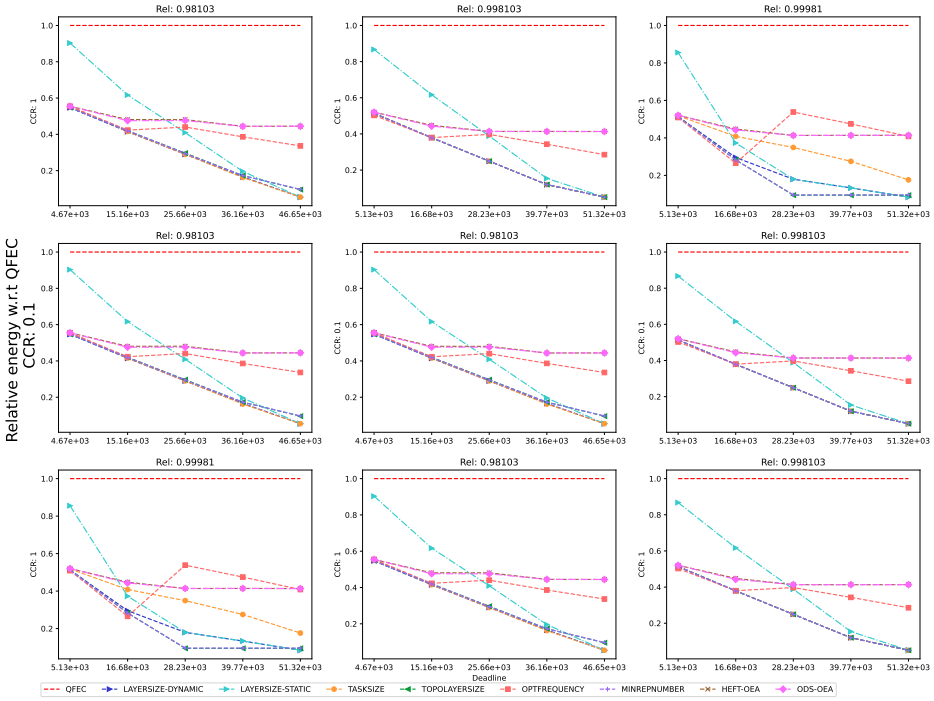

Figure 412: Performance of the different heuristics on the LU workflow.

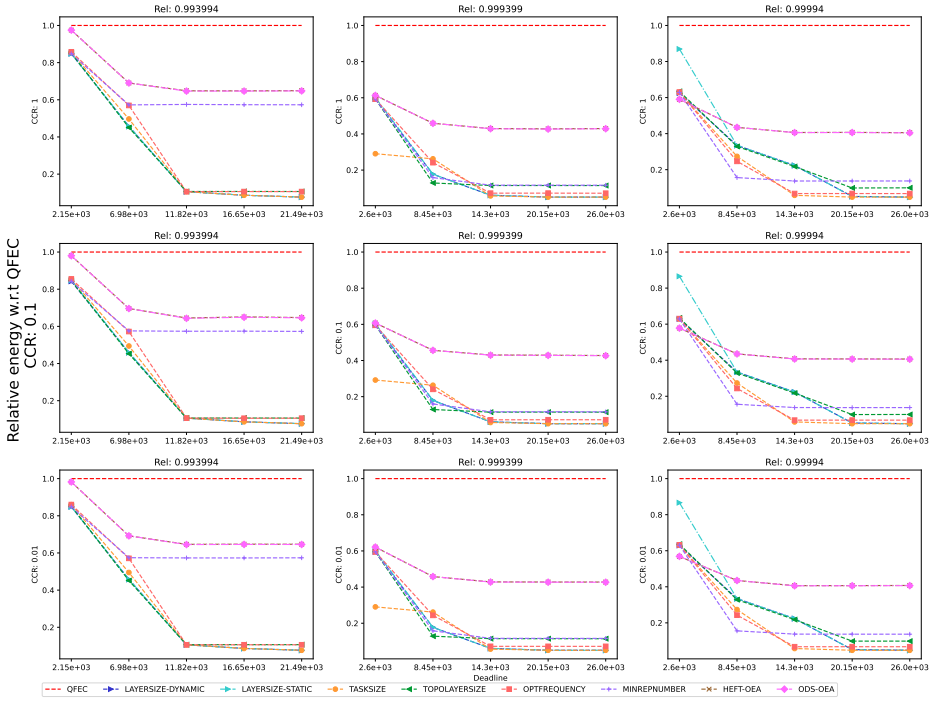

Figure 413: Performance of the different heuristics on the Montage workflow.

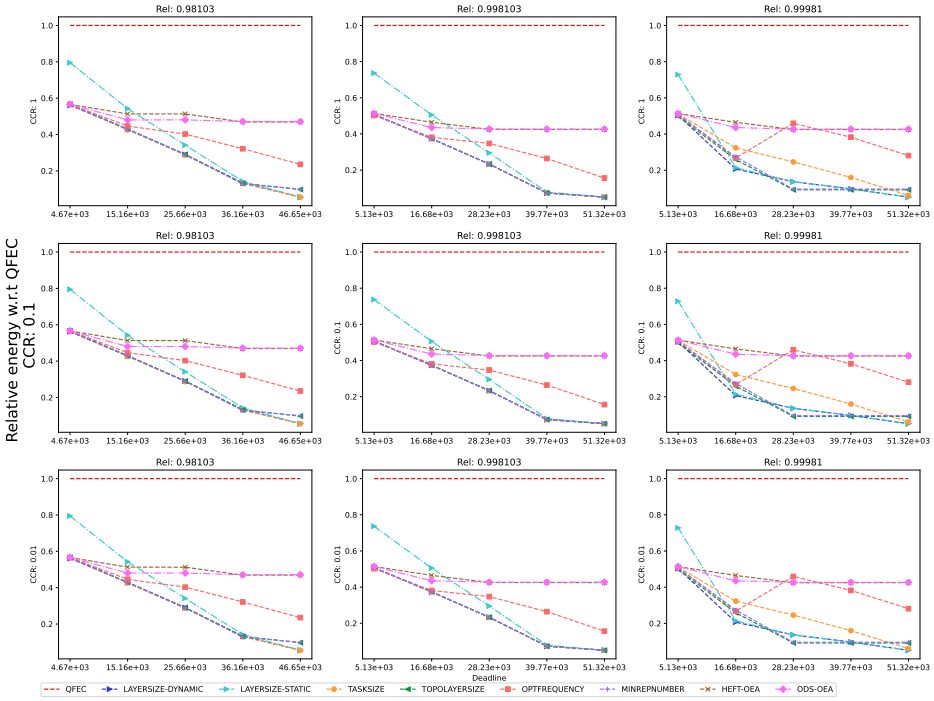

Figure 414: Performance of the different heuristics on the QR workflow.

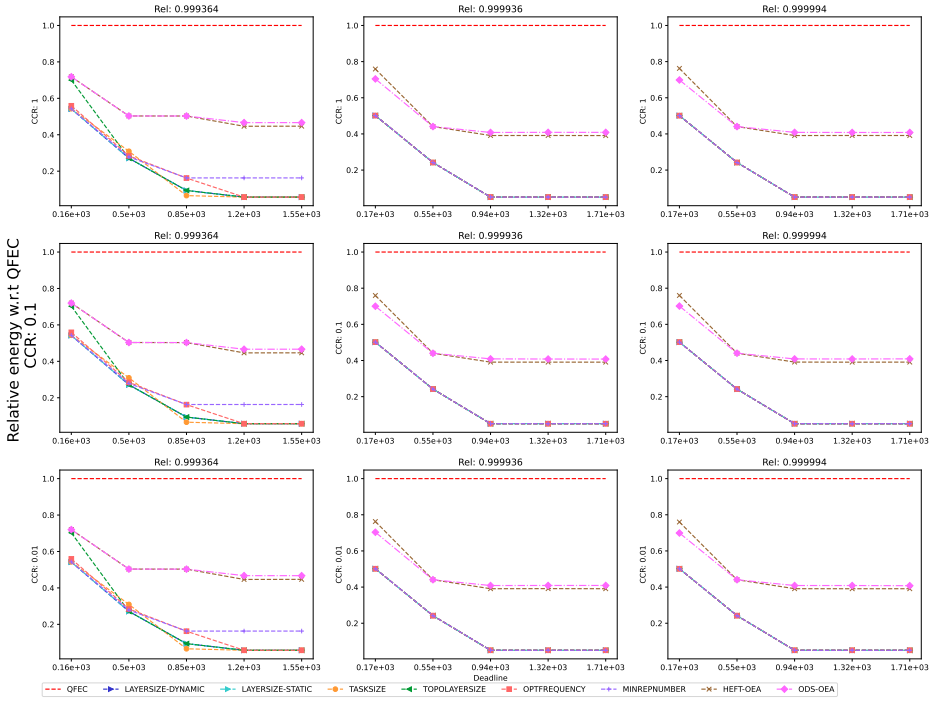

Figure 415: Performance of the different heuristics on the Seismology workflow.

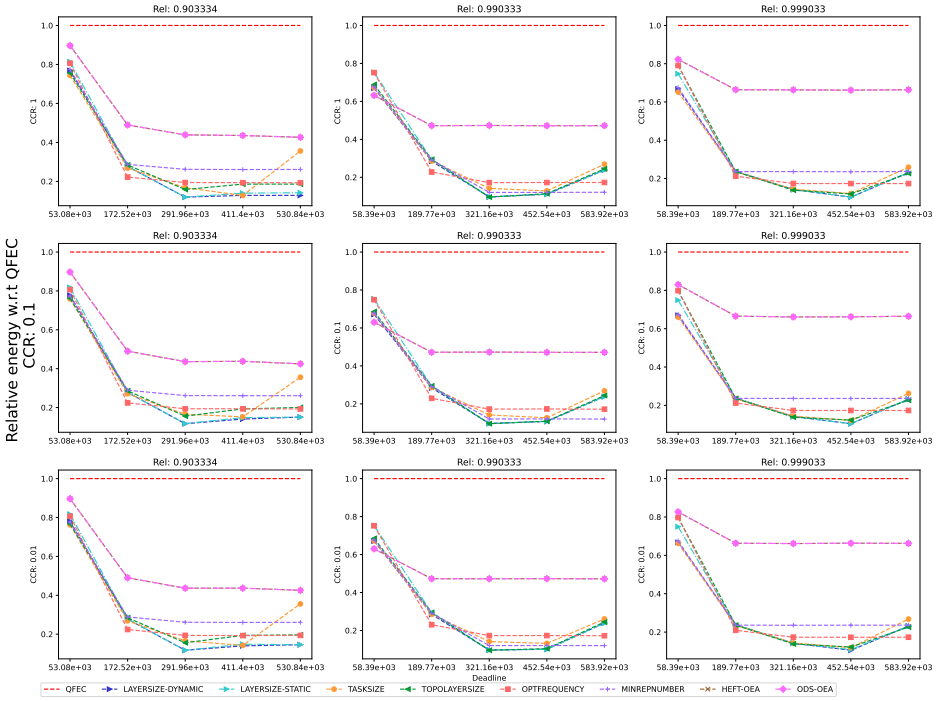

Figure 416: Performance of the different heuristics on the SoyKB workflow.

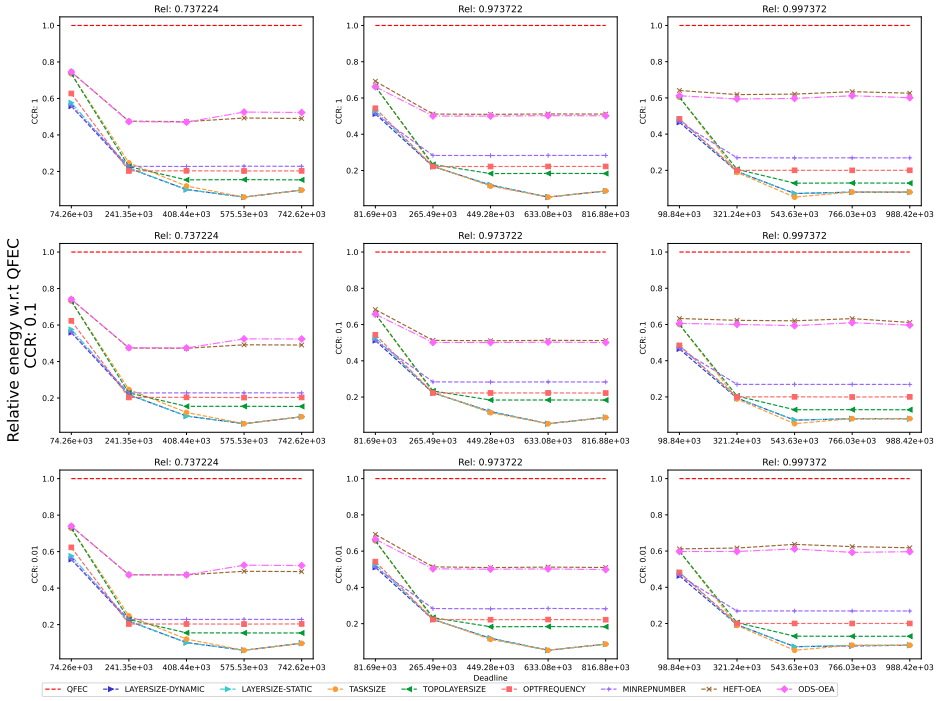

Figure 417: Performance of the different heuristics on the SRASearch.

D.9.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

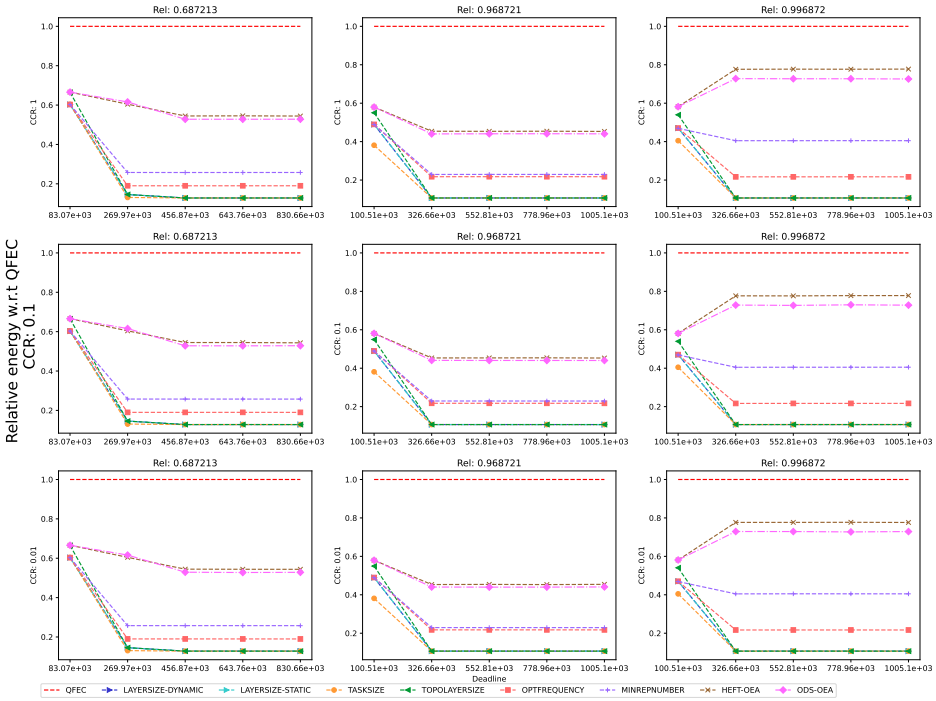

Figure 418: Performance of the different heuristics on the BLAST workflow.

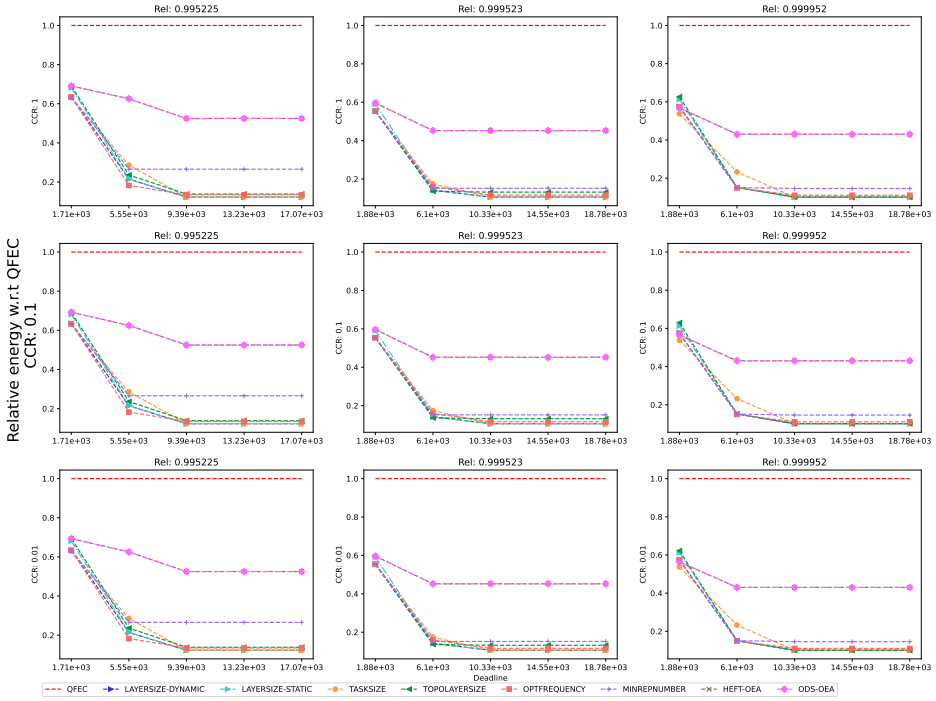

Figure 419: Performance of the different heuristics on the BWA workflow.

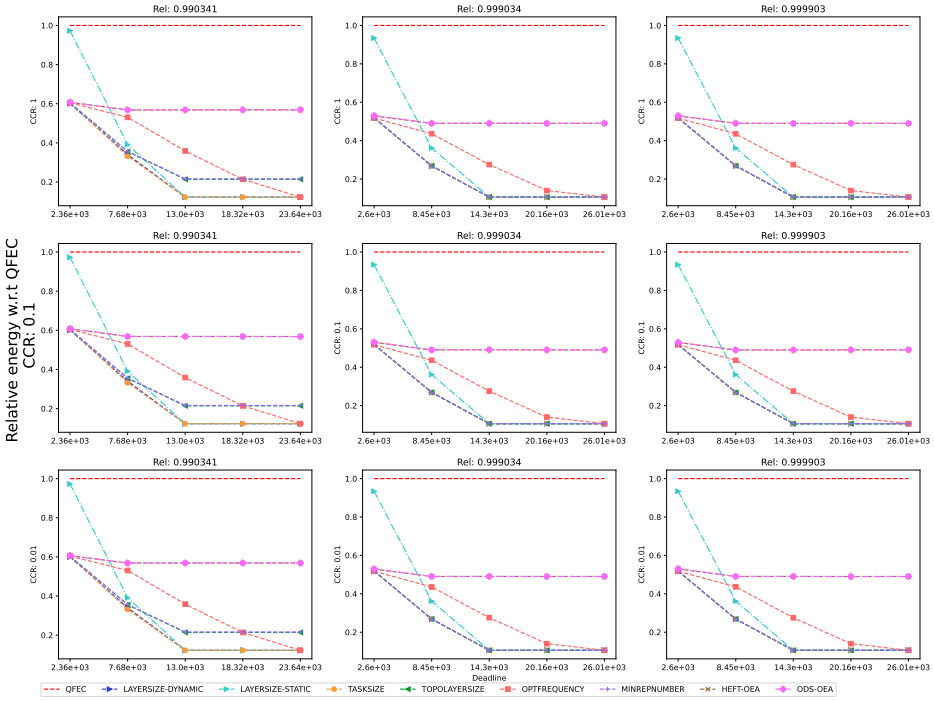

Figure 420: Performance of the different heuristics on the Cholesky workflow.

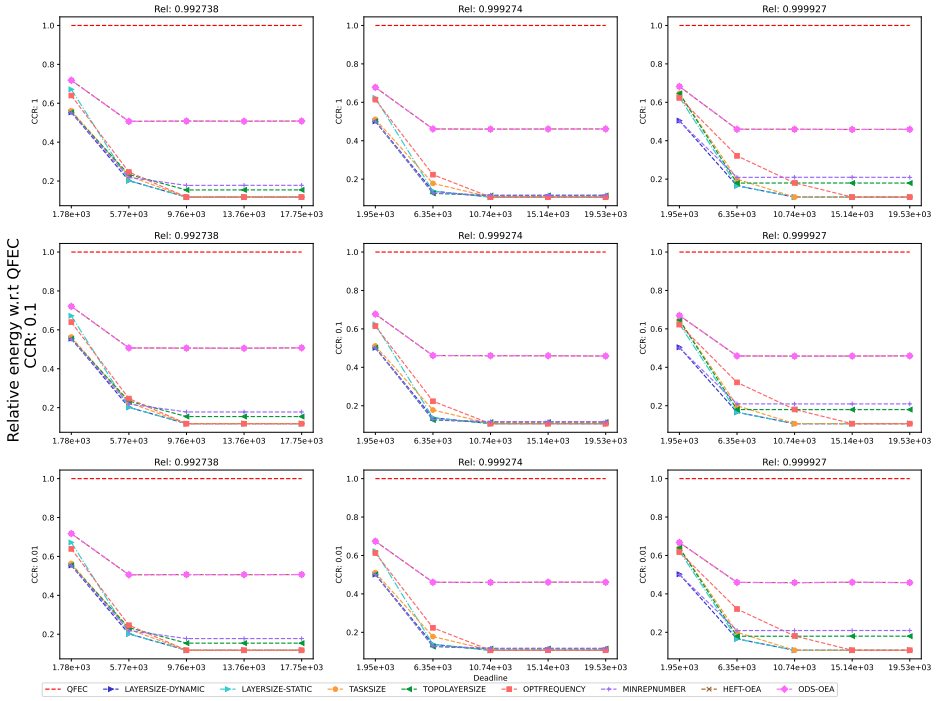

Figure 421: Performance of the different heuristics on the Cycles workflow.

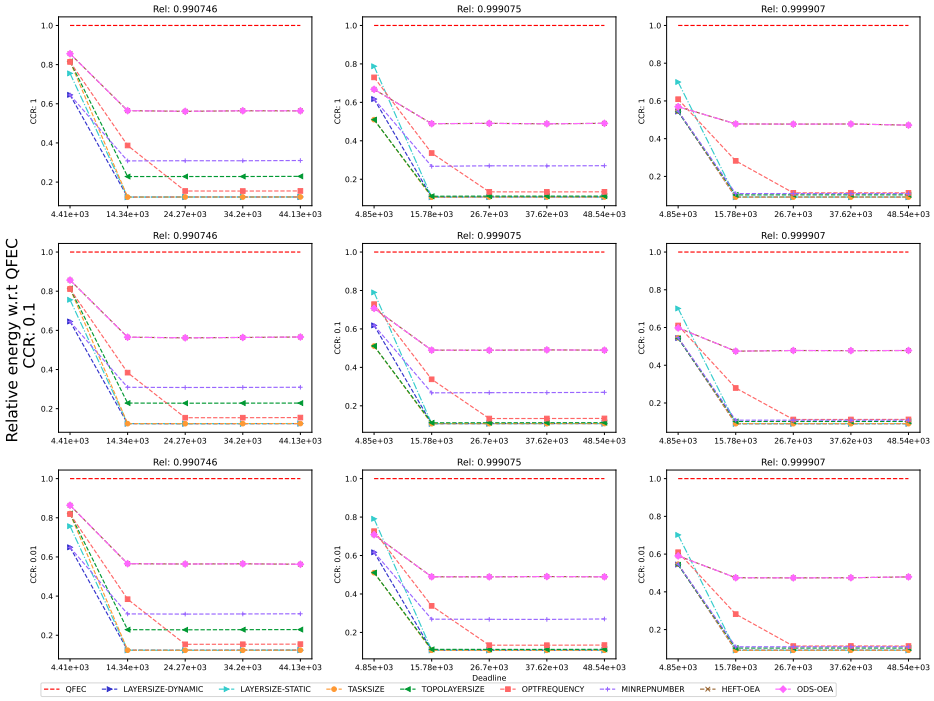

Figure 422: Performance of the different heuristics on the Epigenomics workflow.

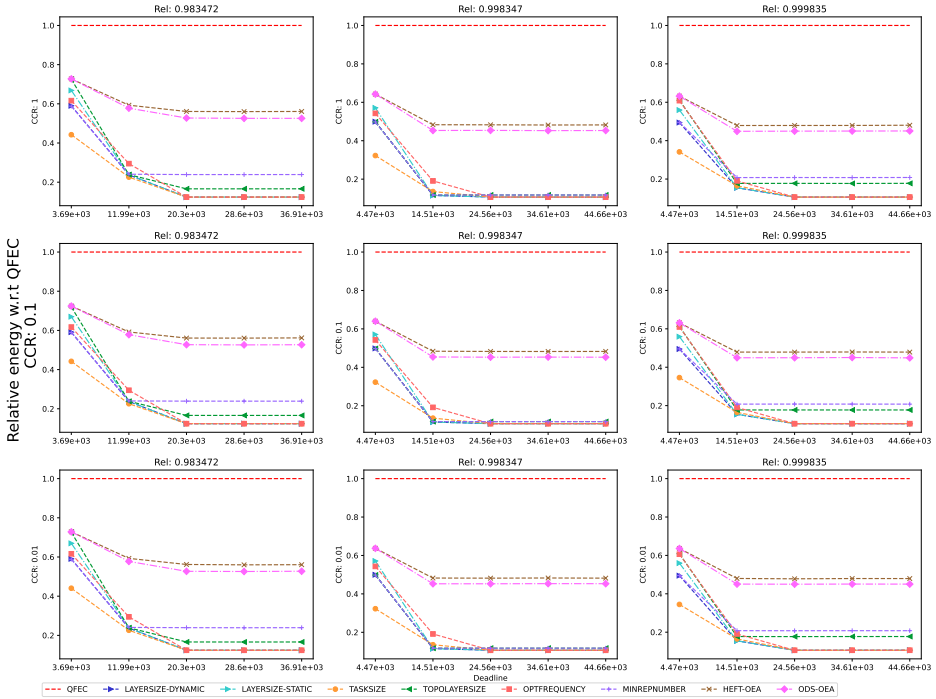

Figure 423: Performance of the different heuristics on the Genome workflow.

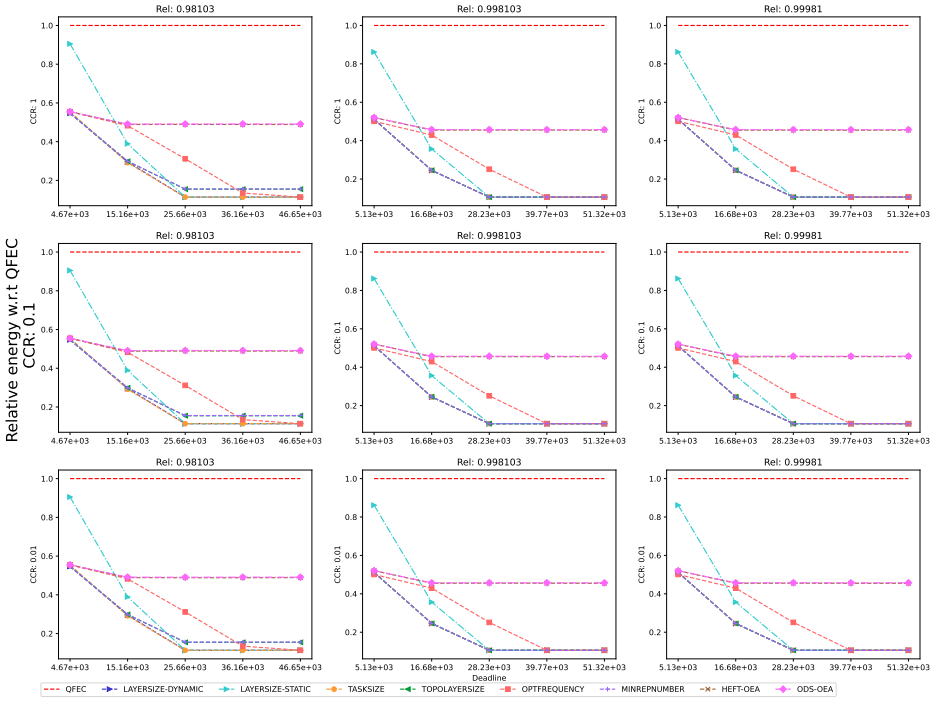

Figure 424: Performance of the different heuristics on the LU workflow.

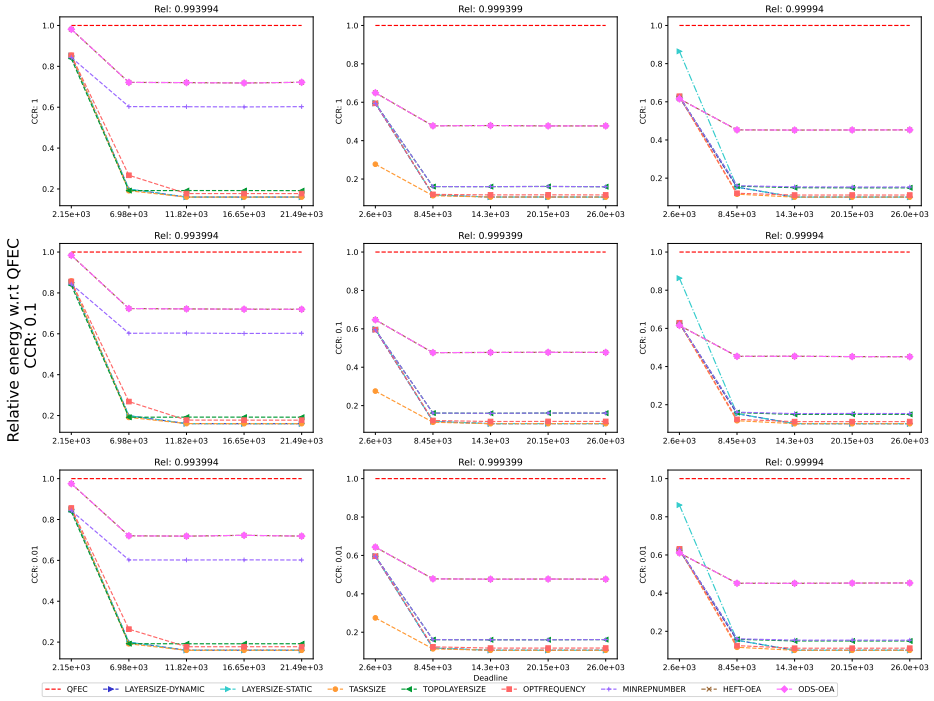

Figure 425: Performance of the different heuristics on the Montage workflow.

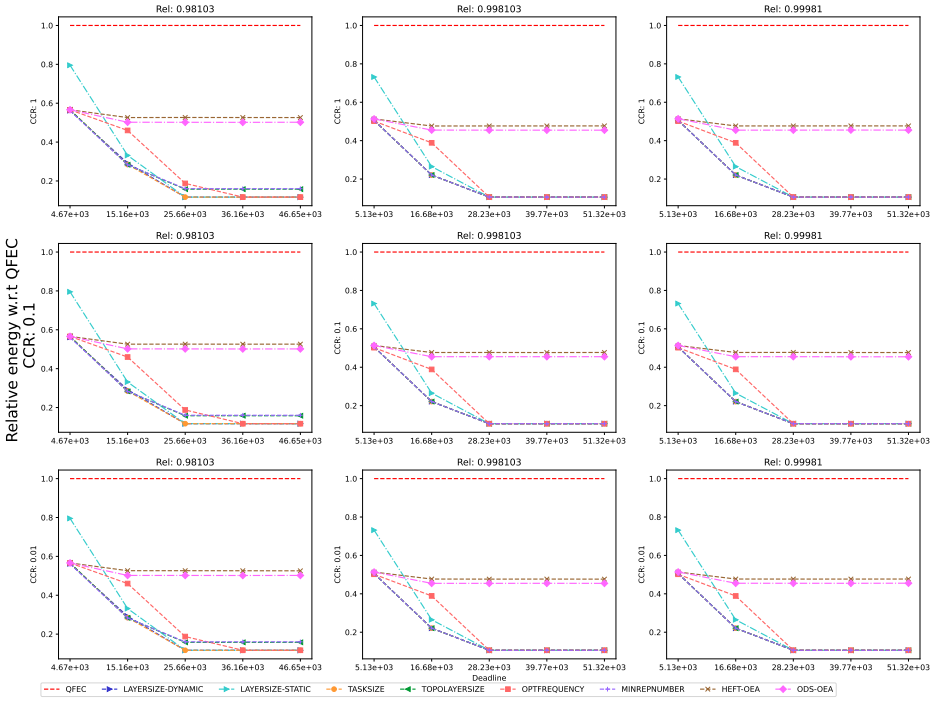

Figure 426: Performance of the different heuristics on the QR workflow.

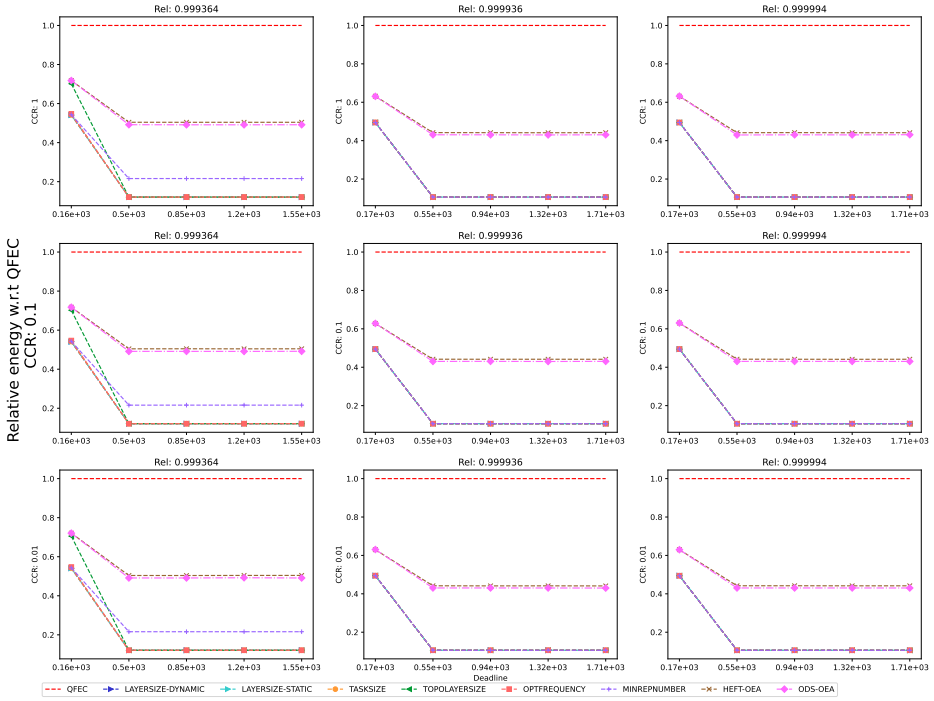

Figure 427: Performance of the different heuristics on the Seismology workflow.

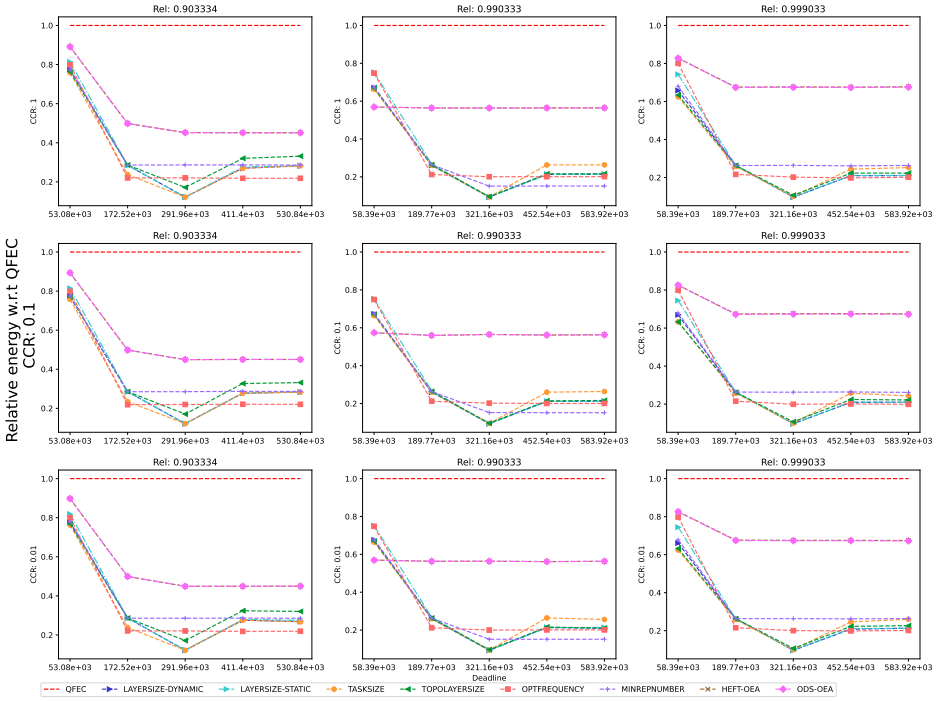

Figure 428: Performance of the different heuristics on the SoyKB workflow.

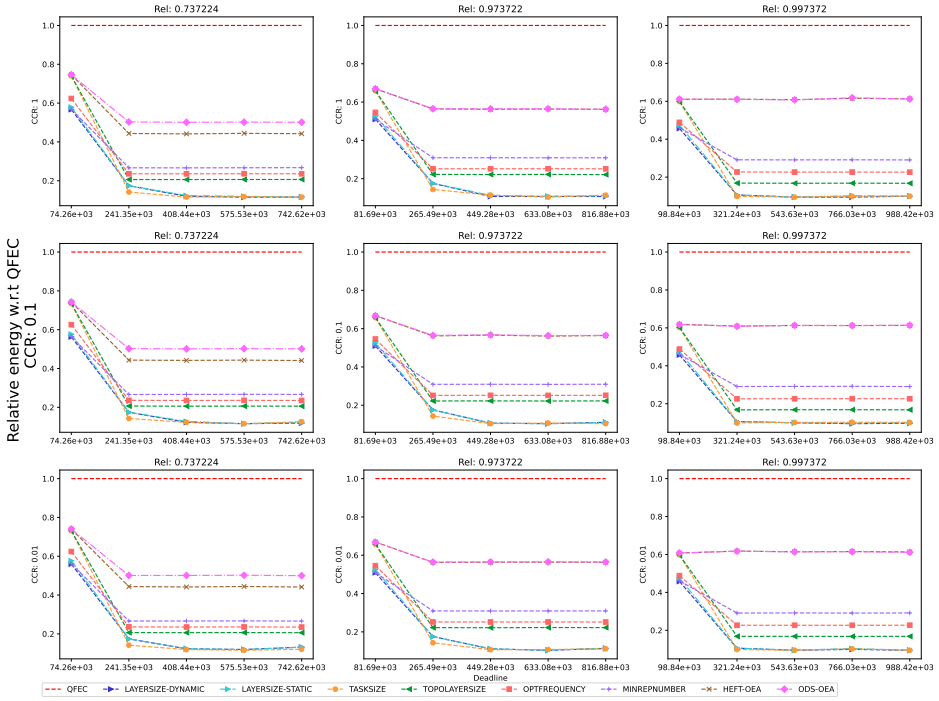

Figure 429: Performance of the different heuristics on the SRASearch.

D.9.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

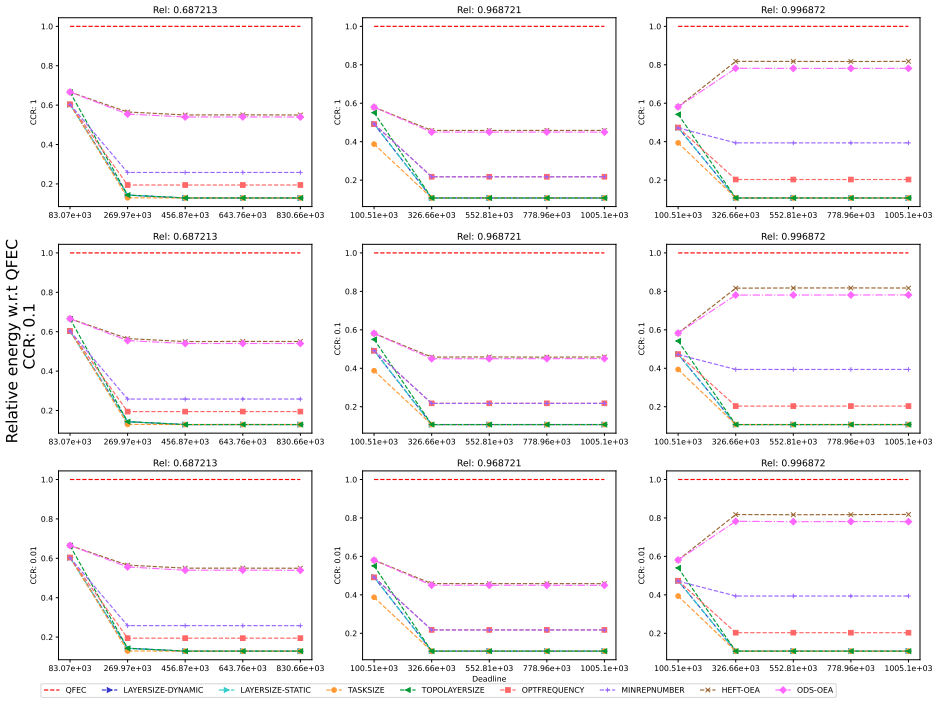

Figure 430: Performance of the different heuristics on the BLAST workflow.

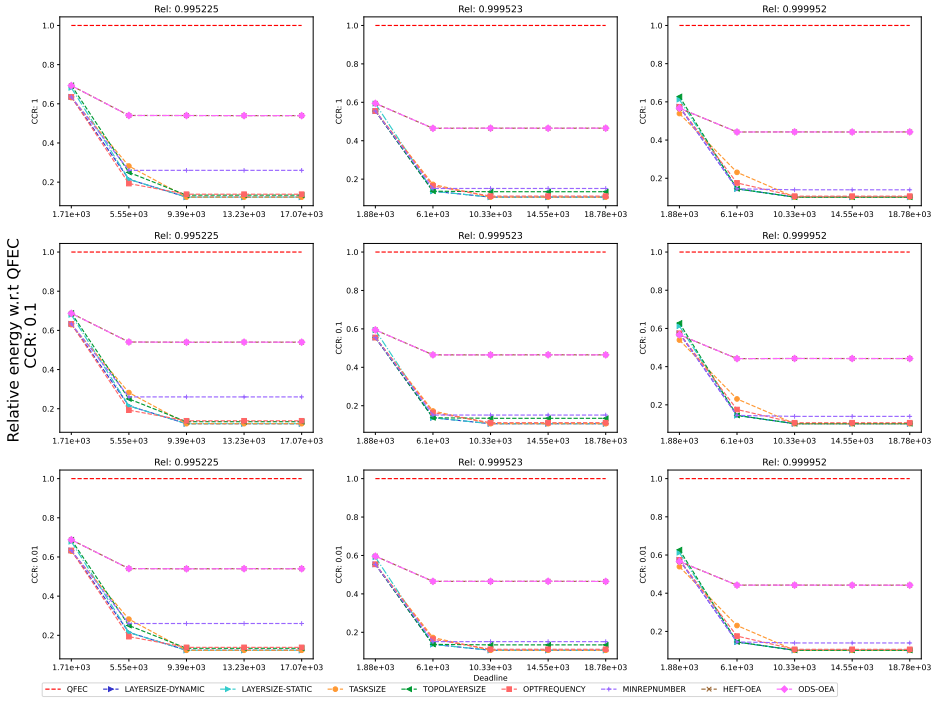

Figure 431: Performance of the different heuristics on the BWA workflow.

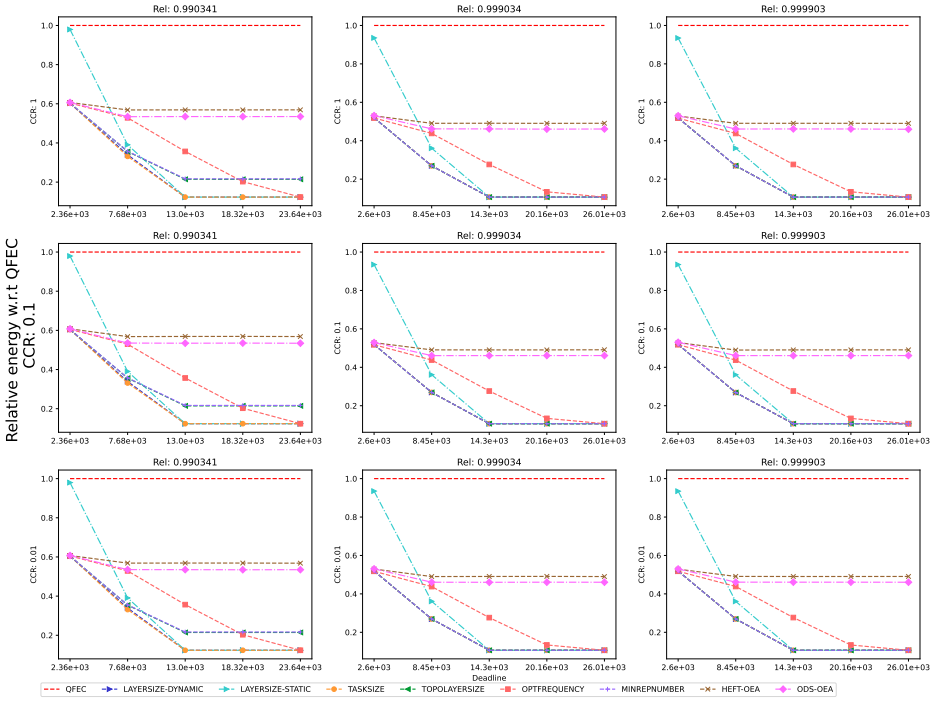

Figure 432: Performance of the different heuristics on the Cholesky workflow.

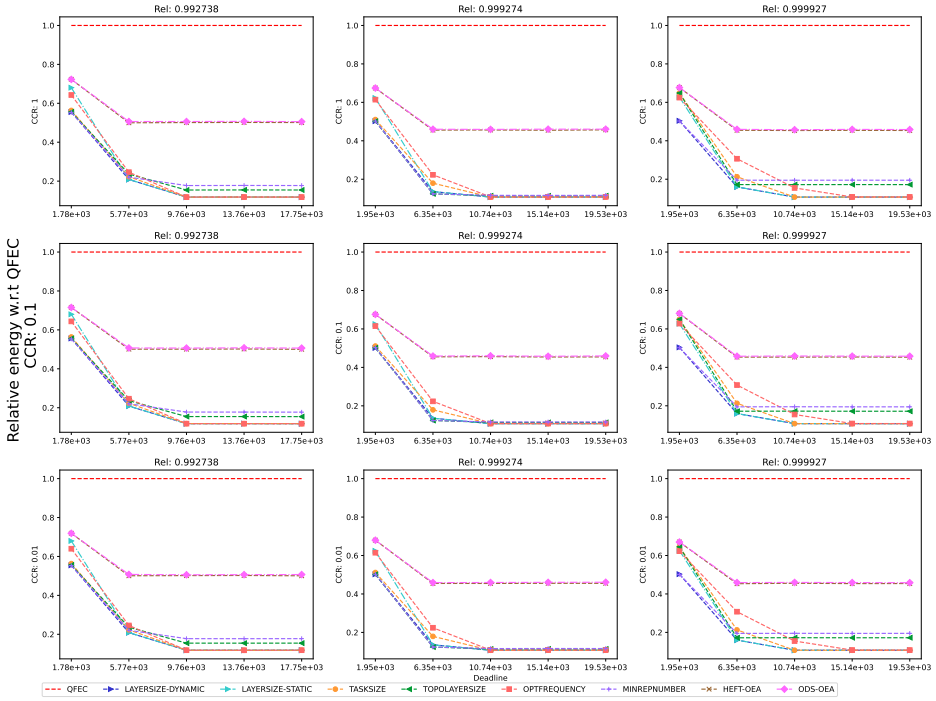

Figure 433: Performance of the different heuristics on the Cycles workflow.

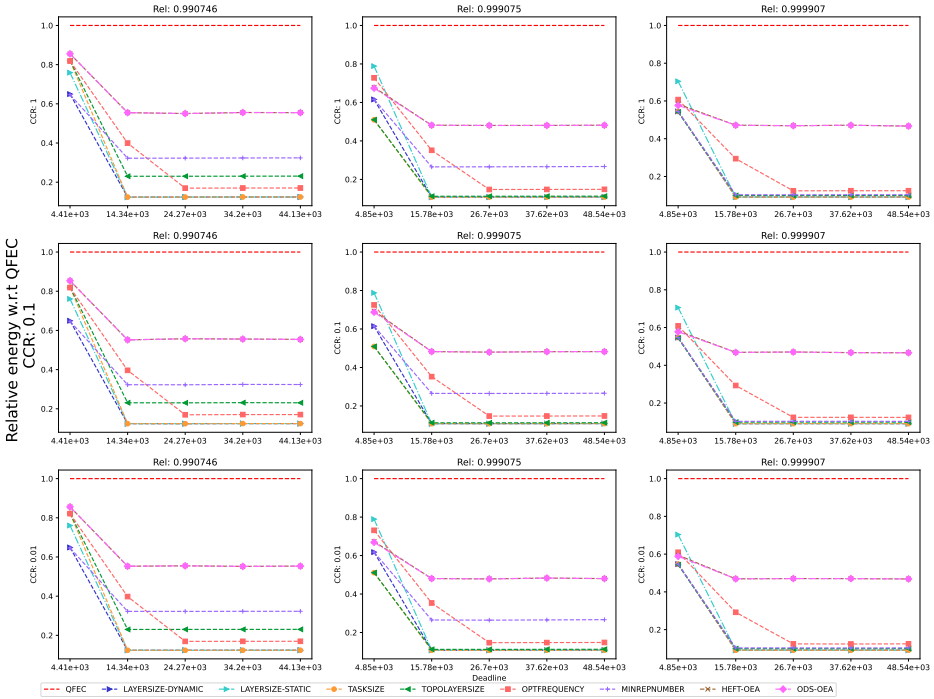

Figure 434: Performance of the different heuristics on the Epigenomics workflow.

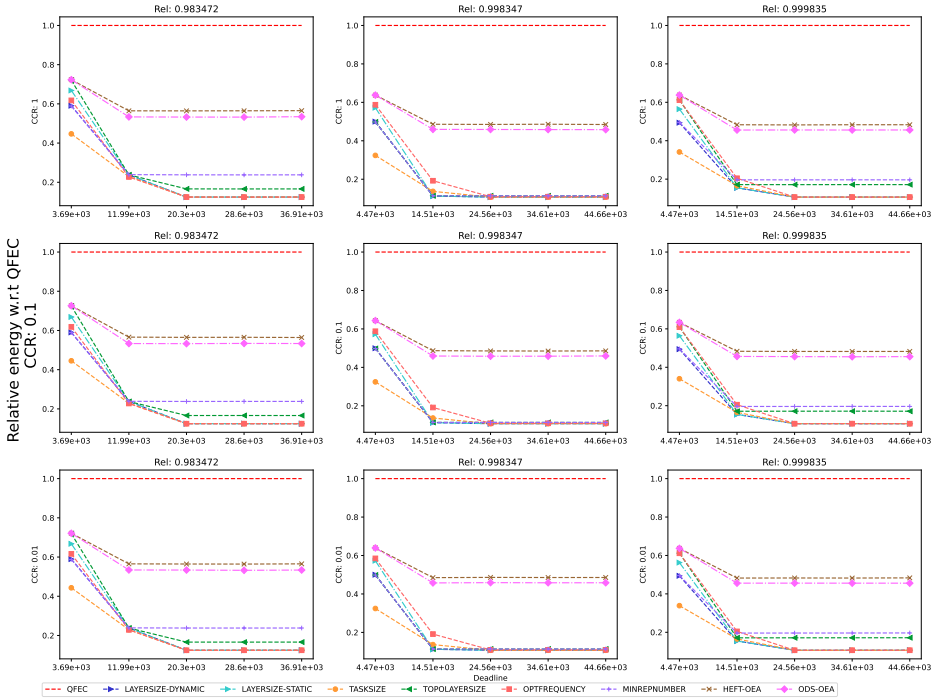

Figure 435: Performance of the different heuristics on the Genome workflow.

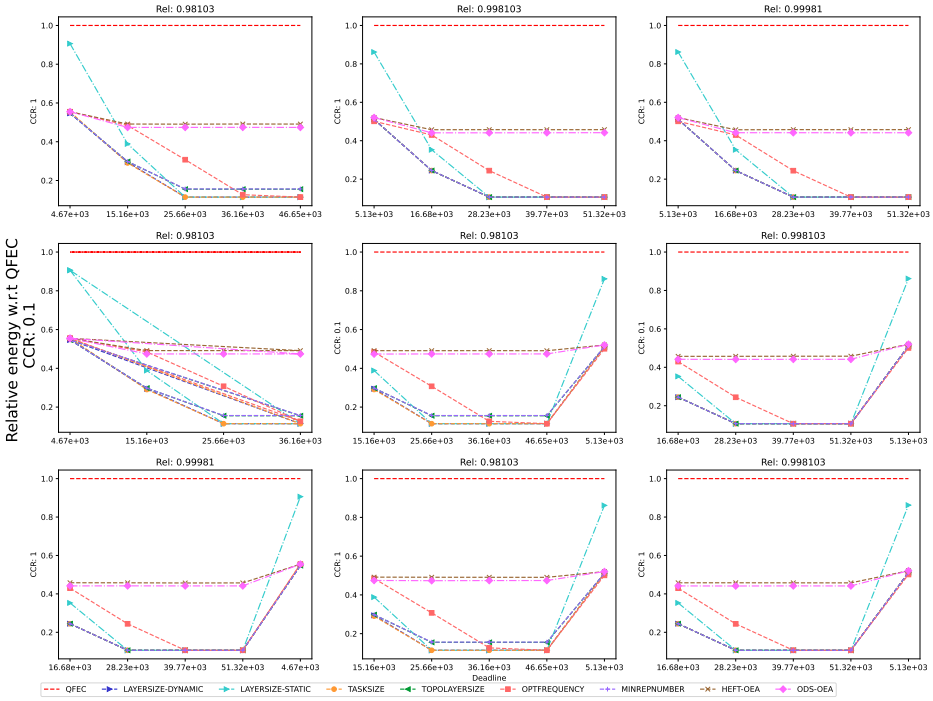

Figure 436: Performance of the different heuristics on the LU workflow.

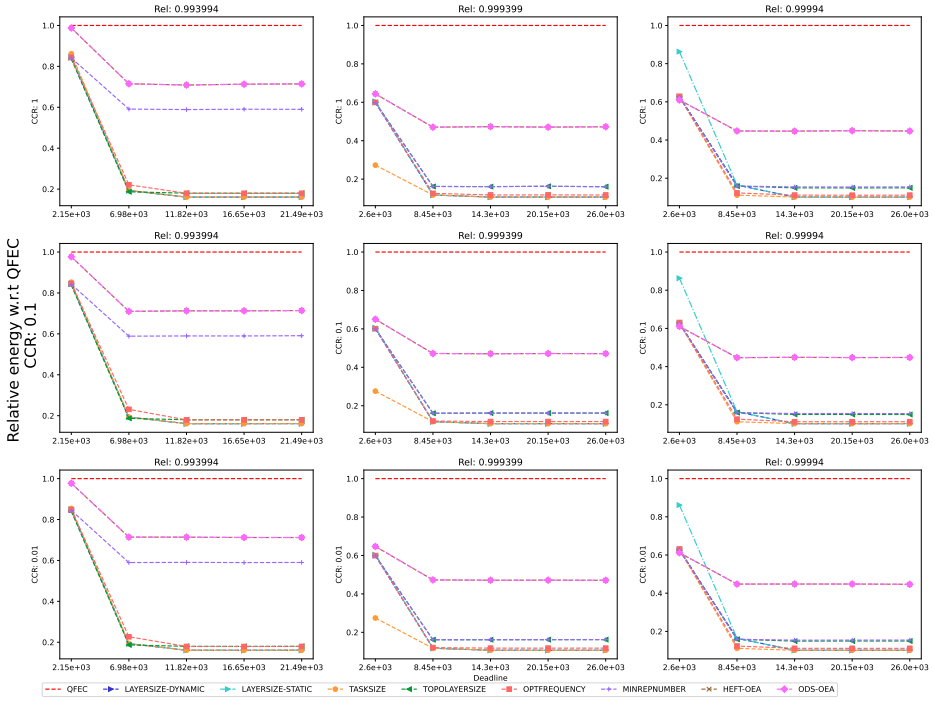

Figure 437: Performance of the different heuristics on the Montage workflow.

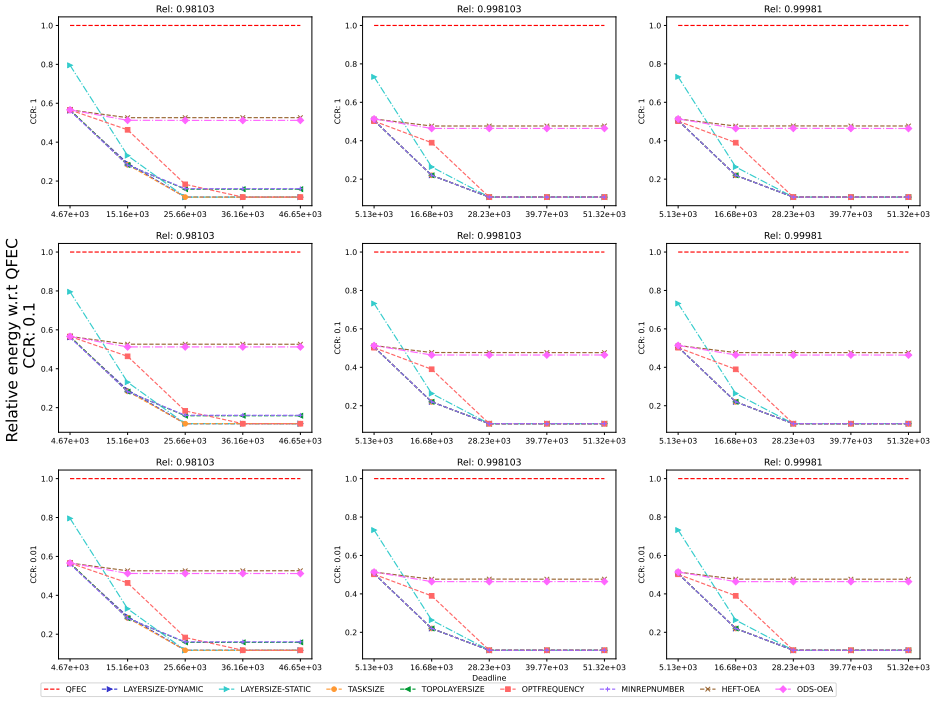

Figure 438: Performance of the different heuristics on the QR workflow.

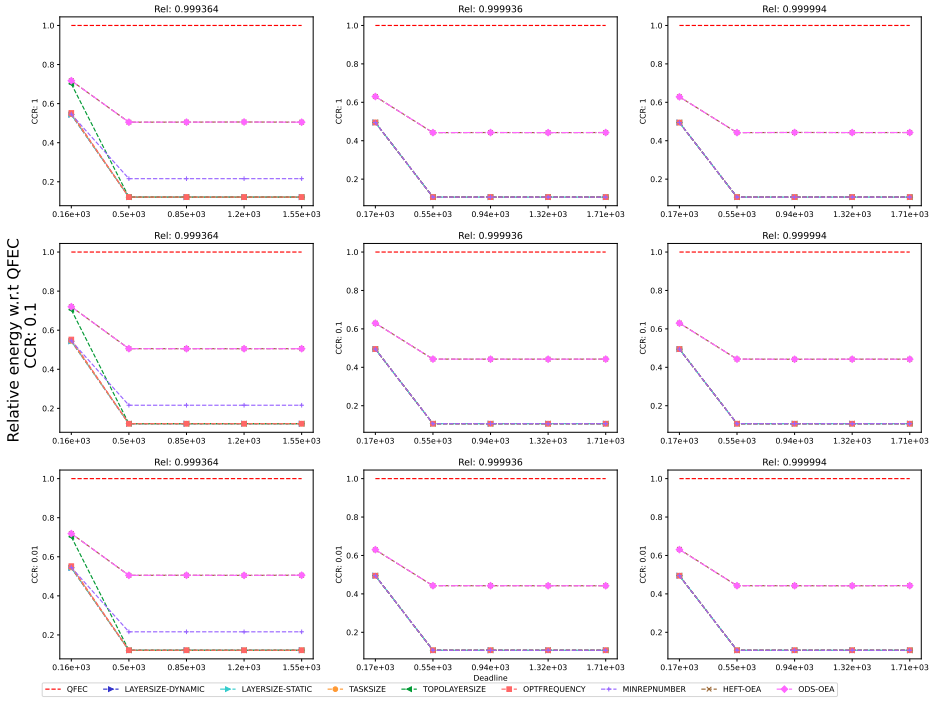

Figure 439: Performance of the different heuristics on the Seismology workflow.

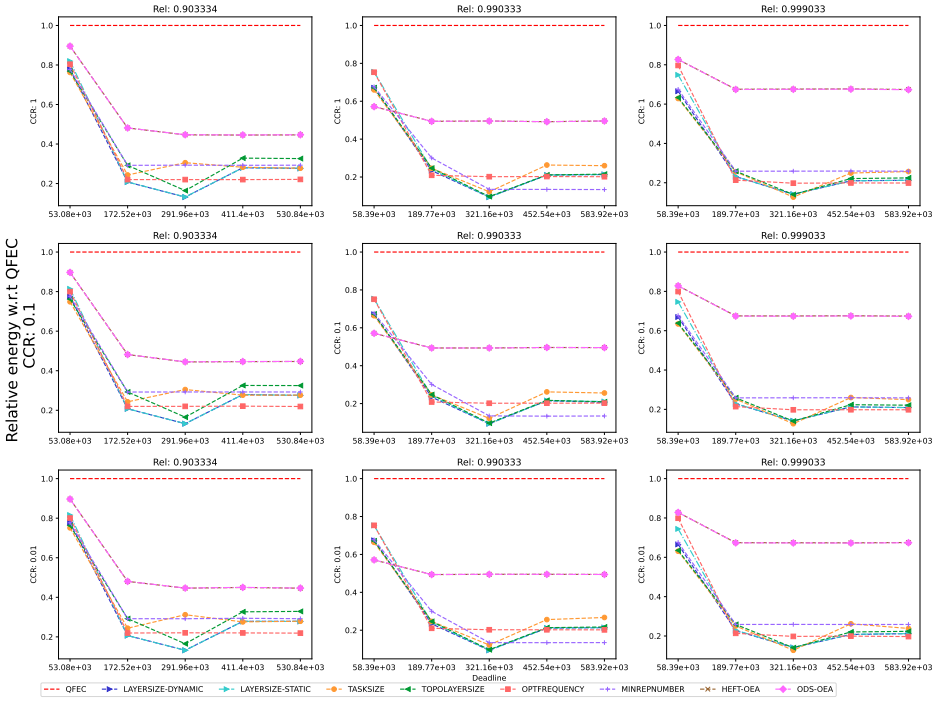

Figure 440: Performance of the different heuristics on the SoyKB workflow.

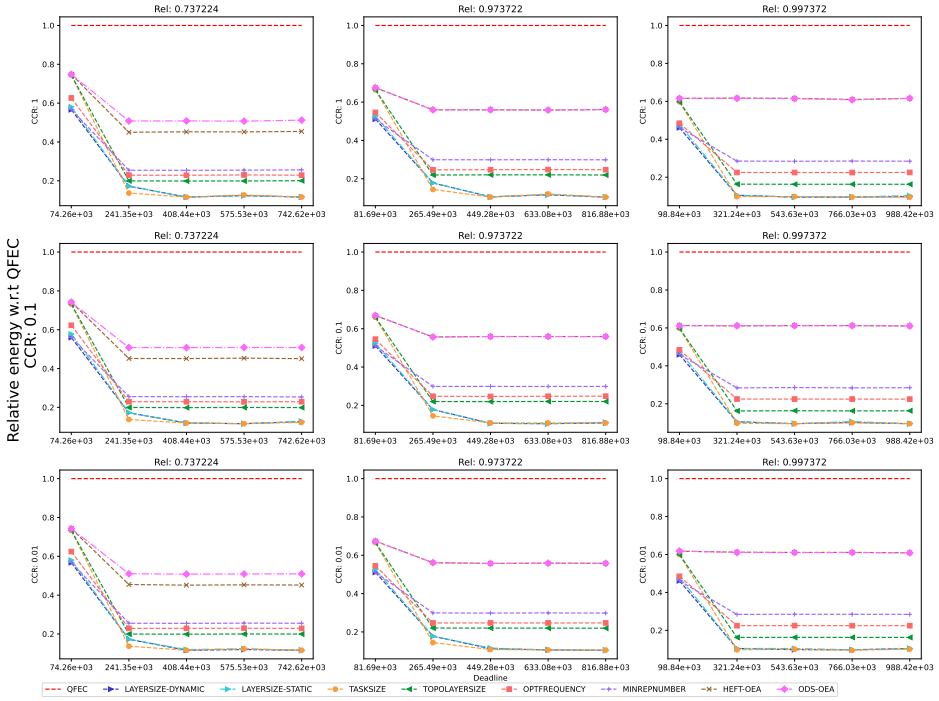

Figure 441: Performance of the different heuristics on the SRASearch.

E Actual execution times drawn from a uniform distribution

E.1 $BC/WC = 0.1$

E.1.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

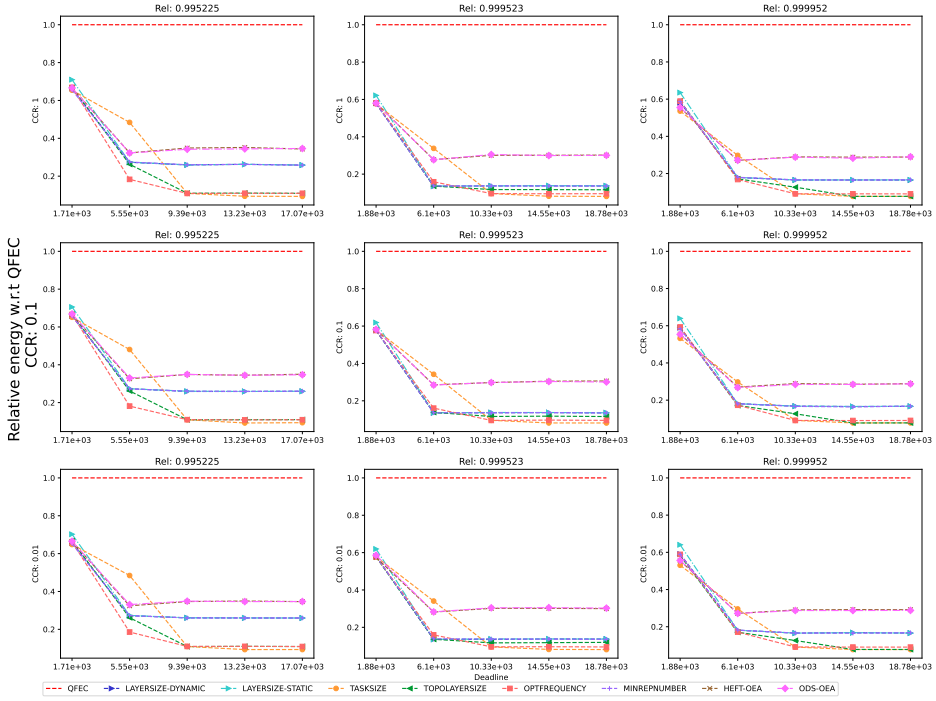

Figure 442: Performance of the different heuristics on the BWA workflow.

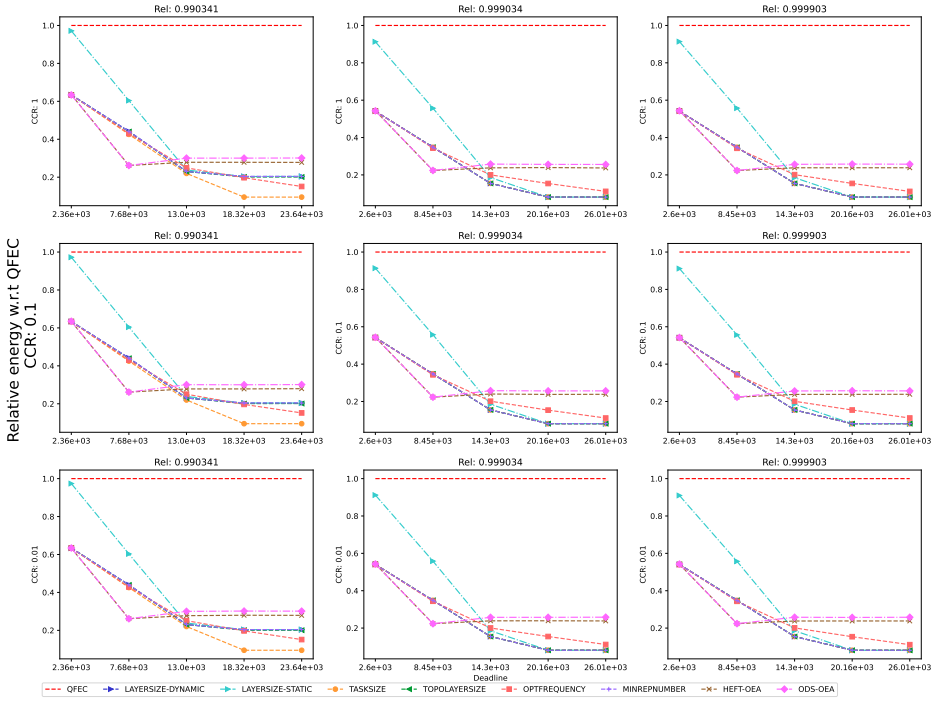

Figure 443: Performance of the different heuristics on the Cholesky workflow.

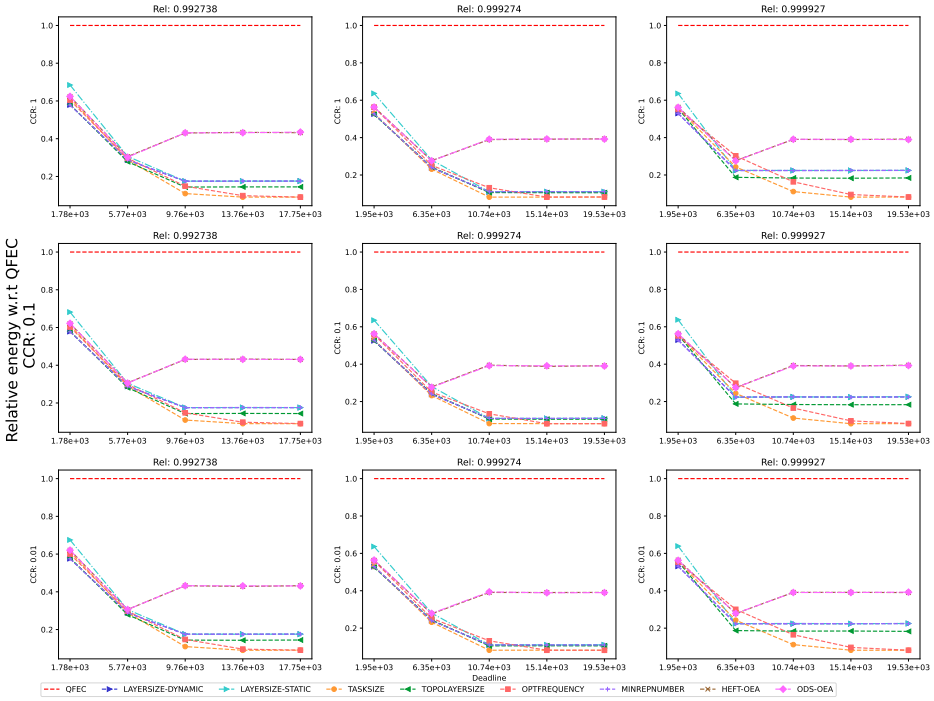

Figure 444: Performance of the different heuristics on the Cycles workflow.

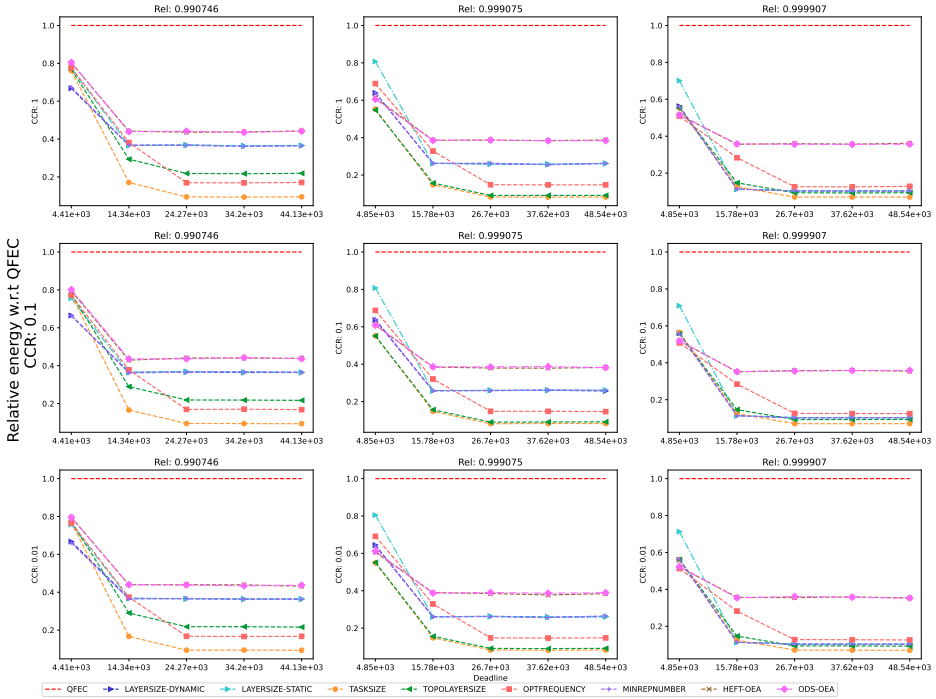

Figure 445: Performance of the different heuristics on the Epigenomics workflow.

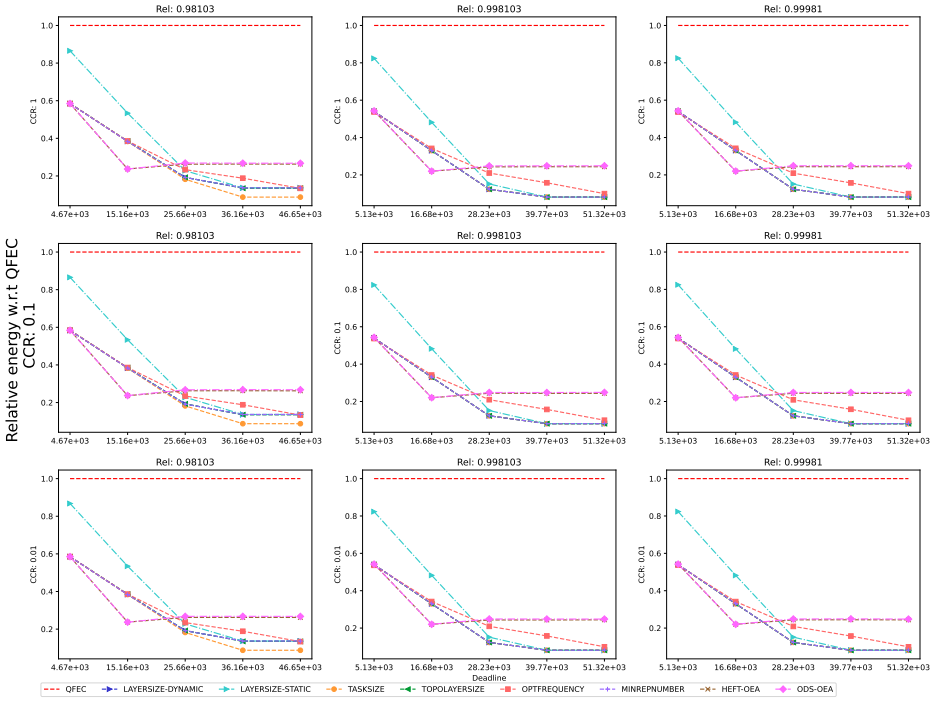

Figure 446: Performance of the different heuristics on the LU workflow.

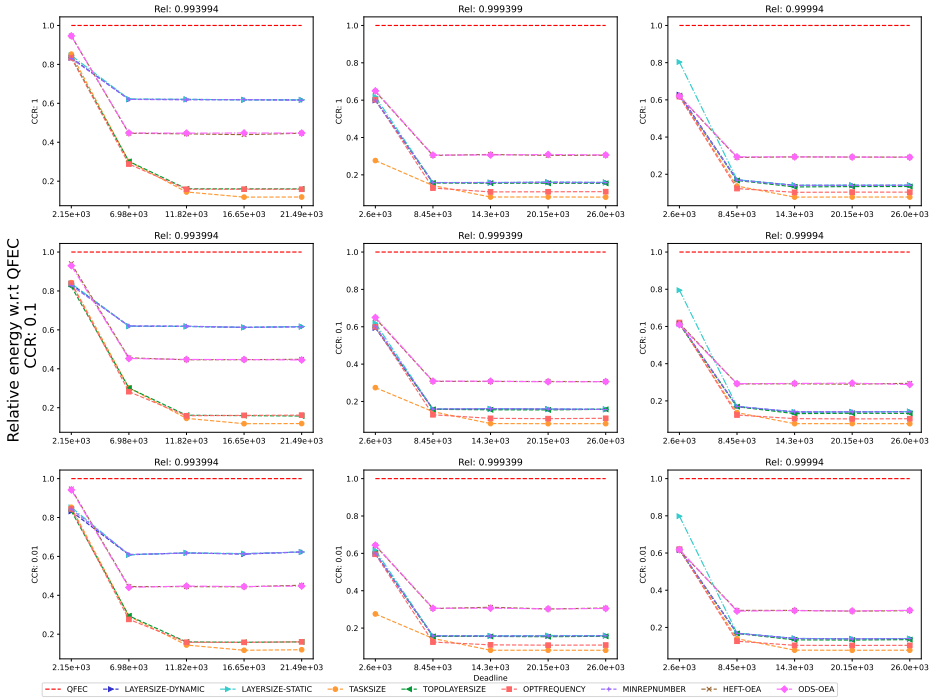

Figure 447: Performance of the different heuristics on the Montage workflow.

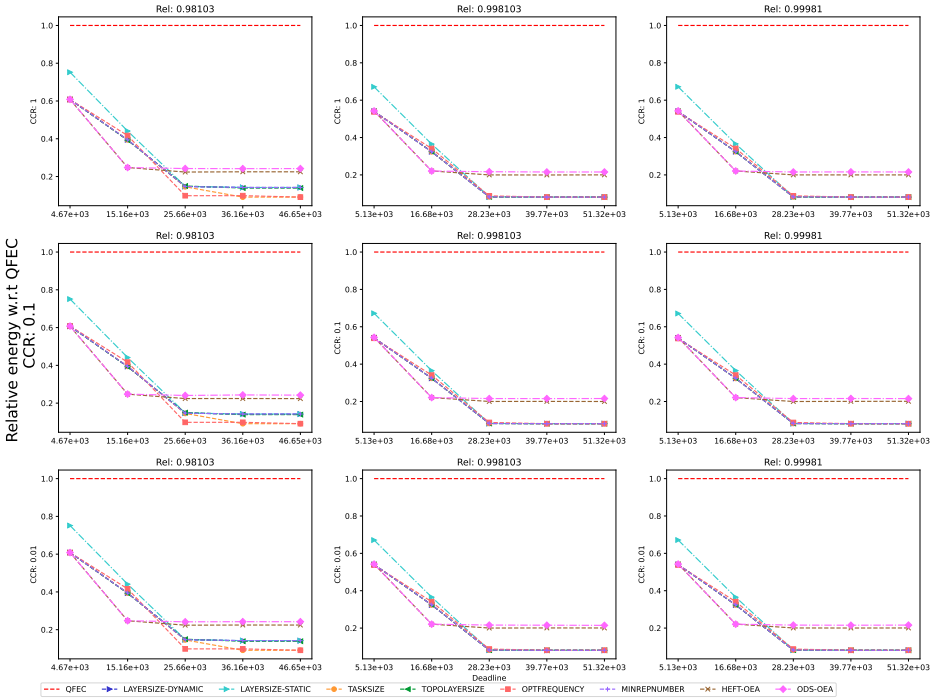

Figure 448: Performance of the different heuristics on the QR workflow.

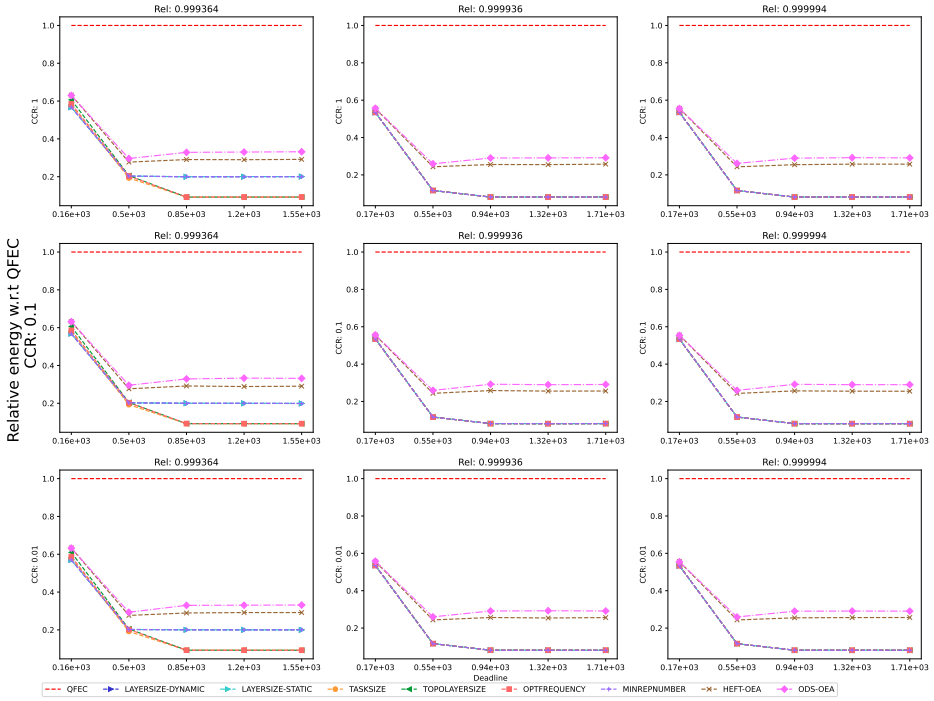

Figure 449: Performance of the different heuristics on the Seismology workflow.

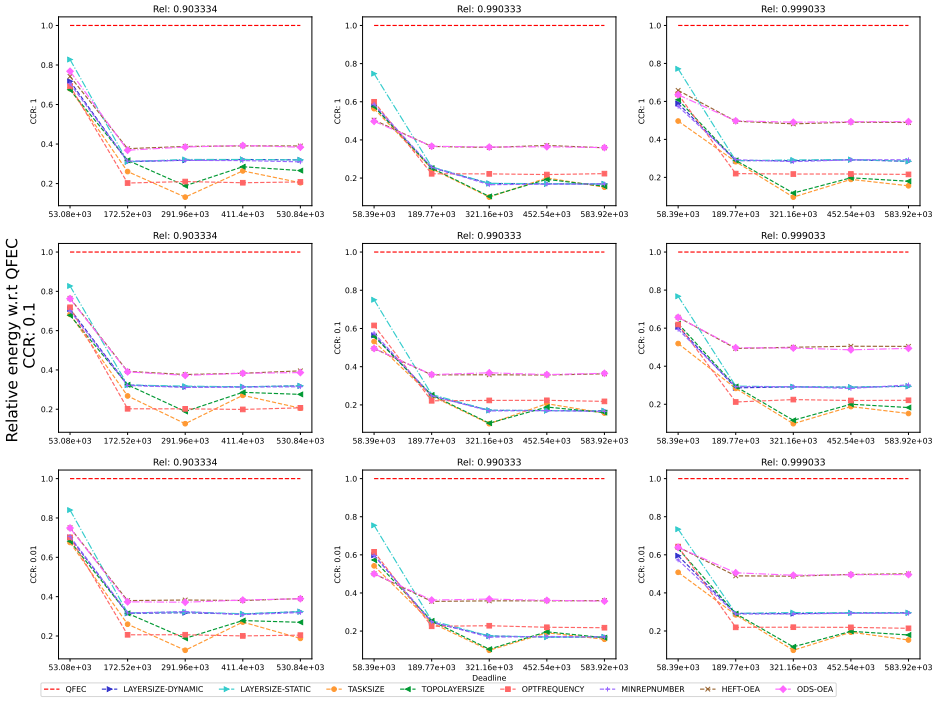

Figure 450: Performance of the different heuristics on the SoyKB workflow.

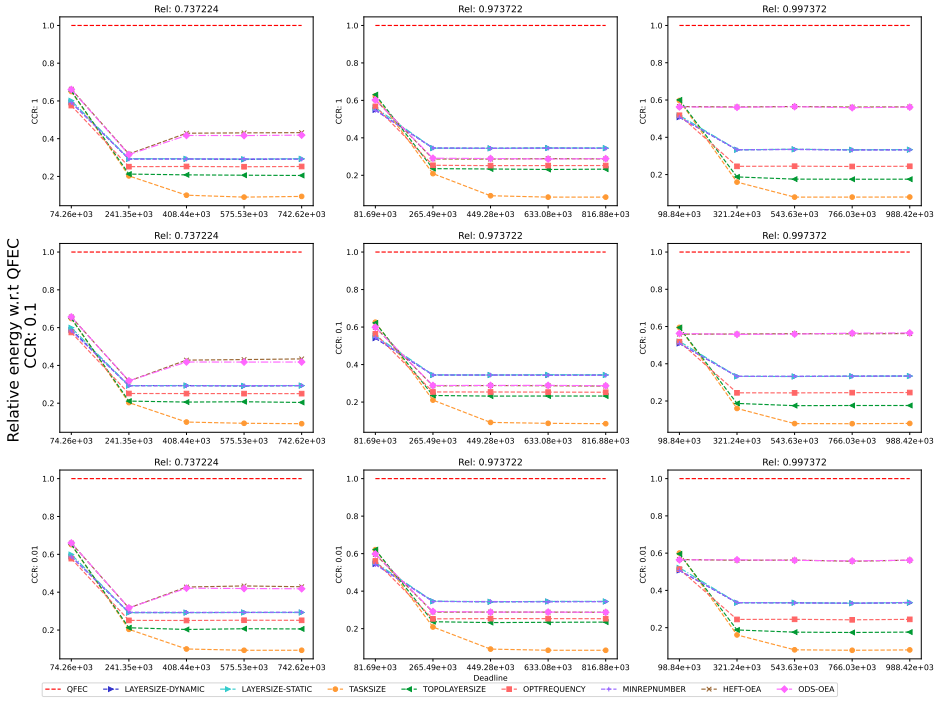

Figure 451: Performance of the different heuristics on the SRASearch.

E.1.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

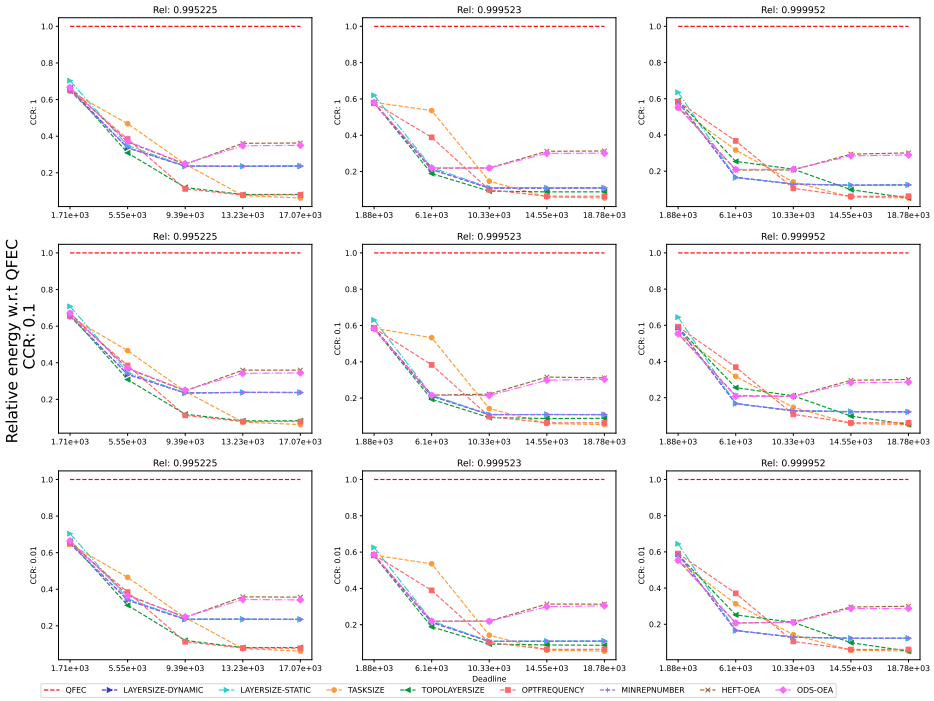

Figure 452: Performance of the different heuristics on the BWA workflow.

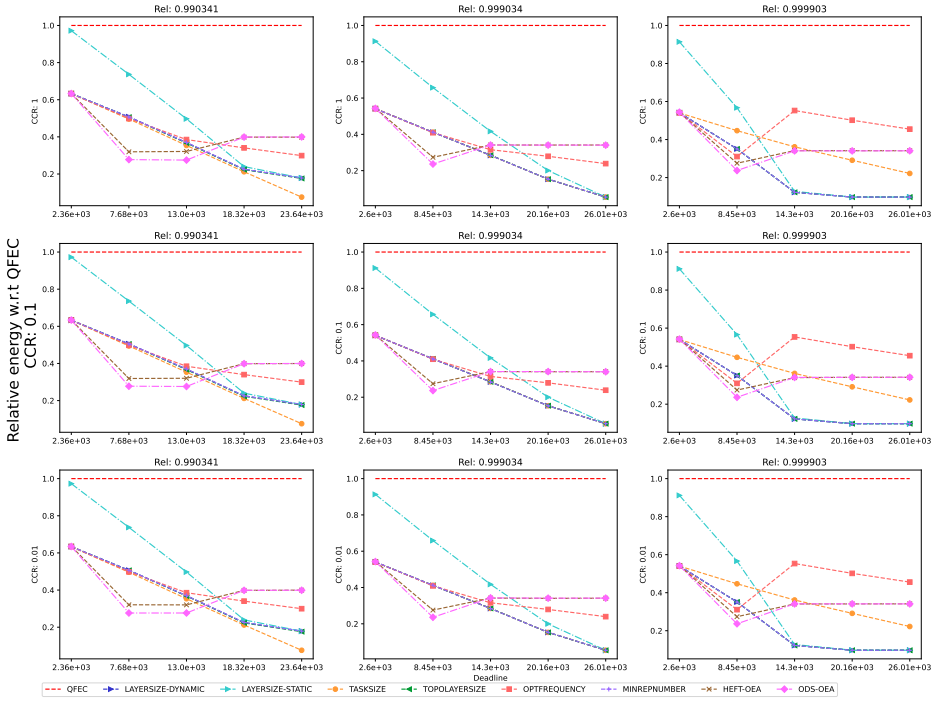

Figure 453: Performance of the different heuristics on the Cholesky workflow.

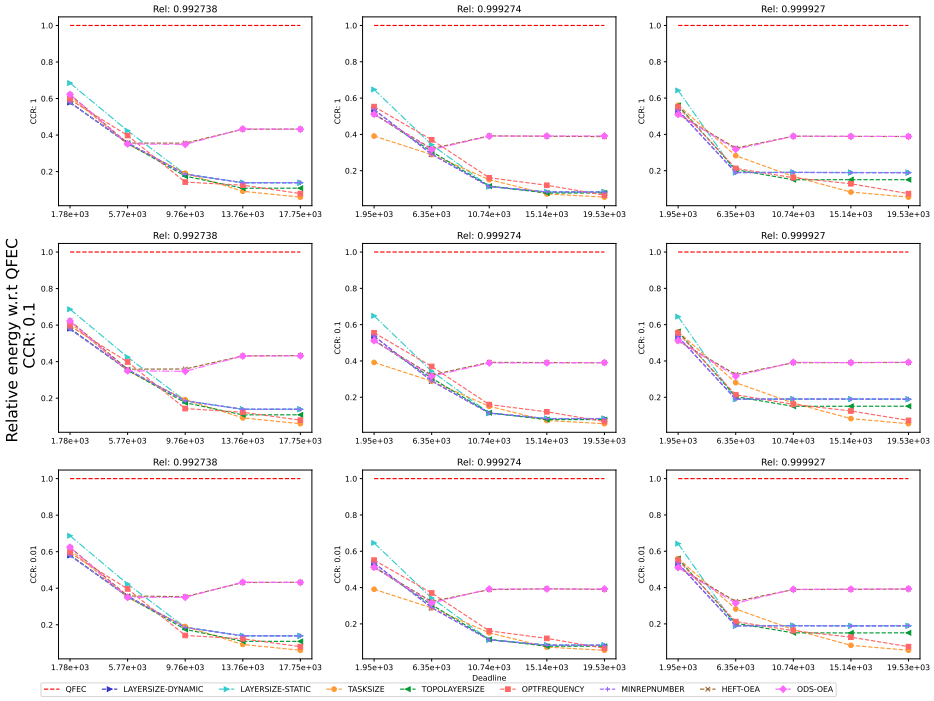

Figure 454: Performance of the different heuristics on the Cycles workflow.

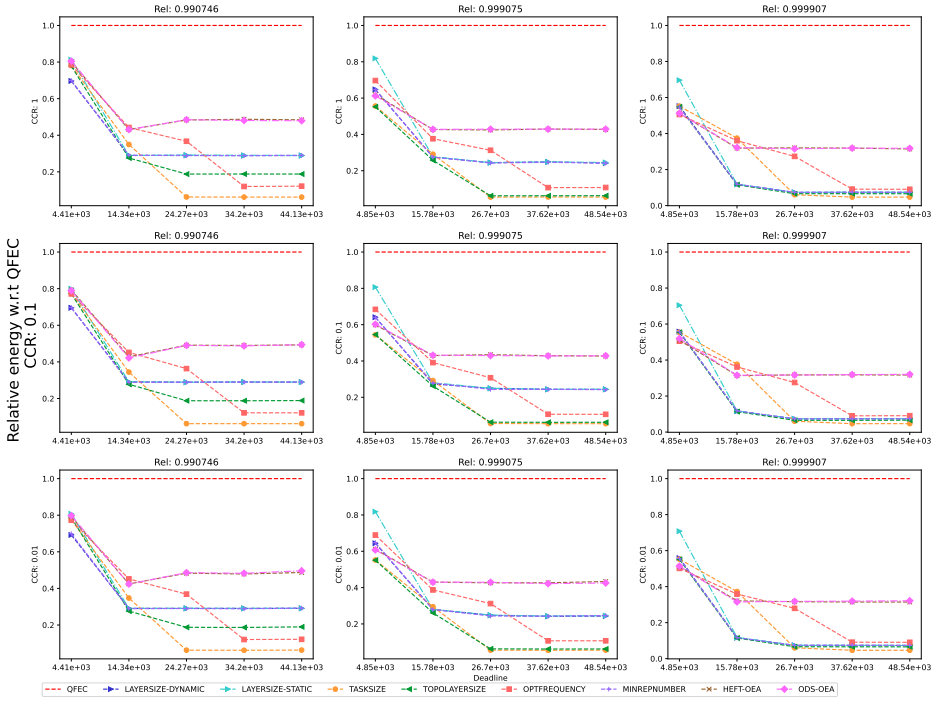

Figure 455: Performance of the different heuristics on the Epigenomics workflow.

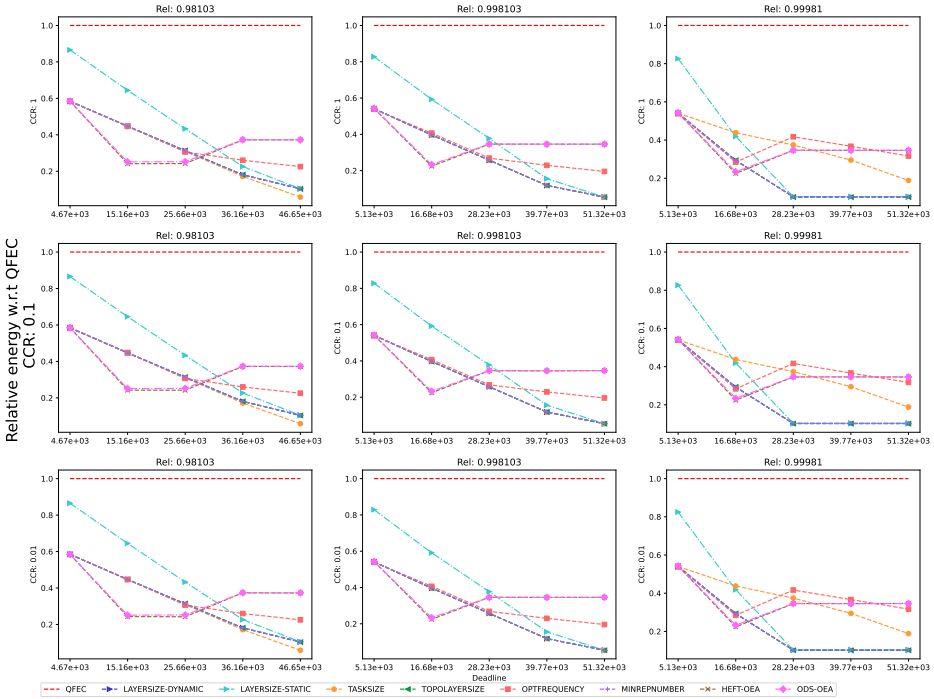

Figure 456: Performance of the different heuristics on the LU workflow.

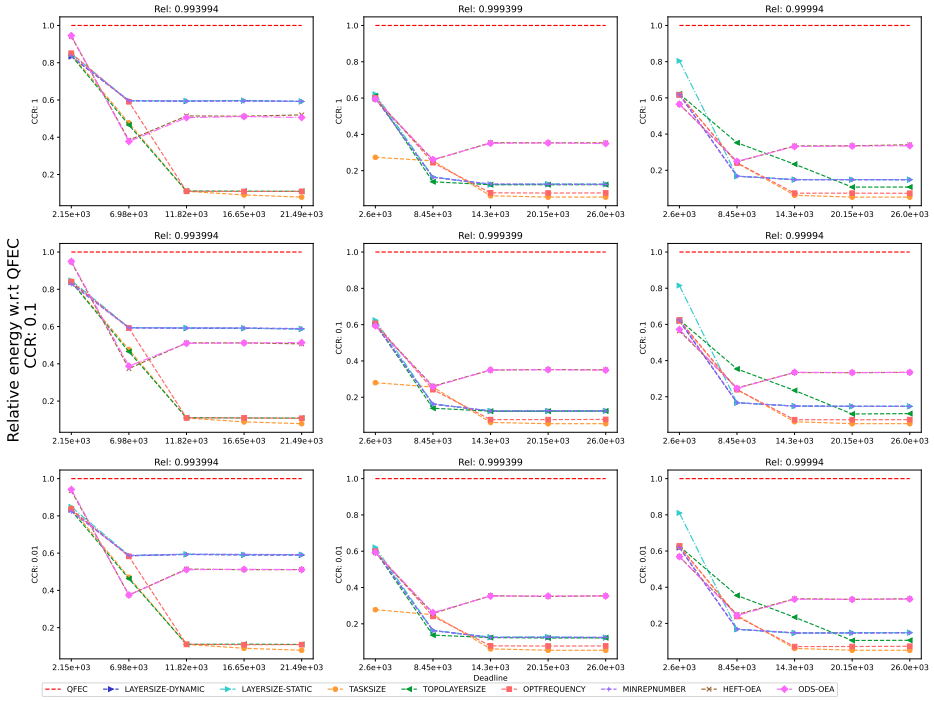

Figure 457: Performance of the different heuristics on the Montage workflow.

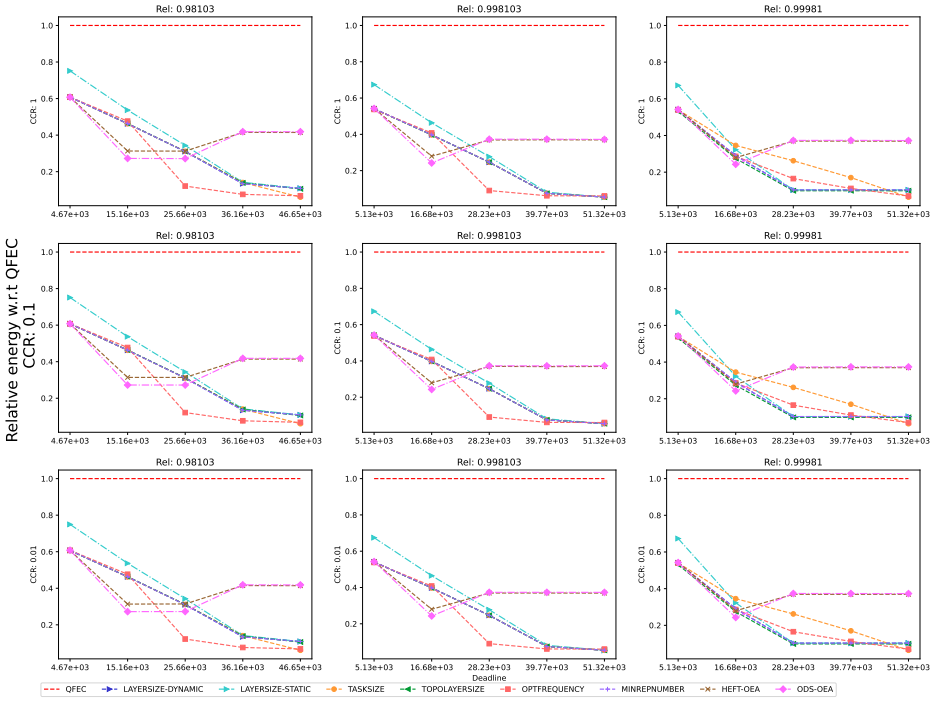

Figure 458: Performance of the different heuristics on the QR workflow.

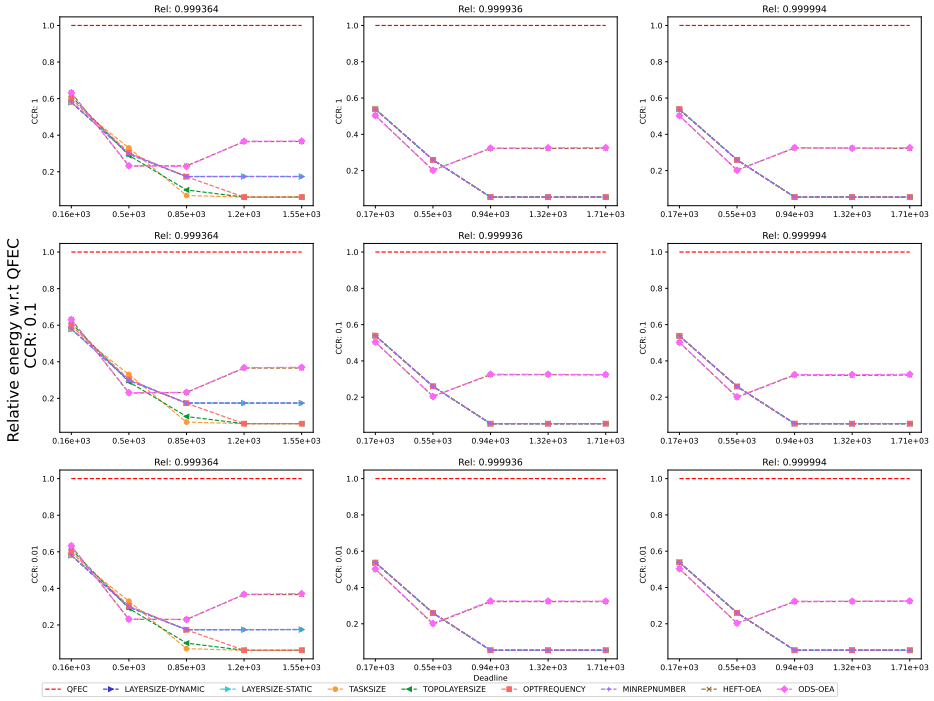

Figure 459: Performance of the different heuristics on the Seismology workflow.

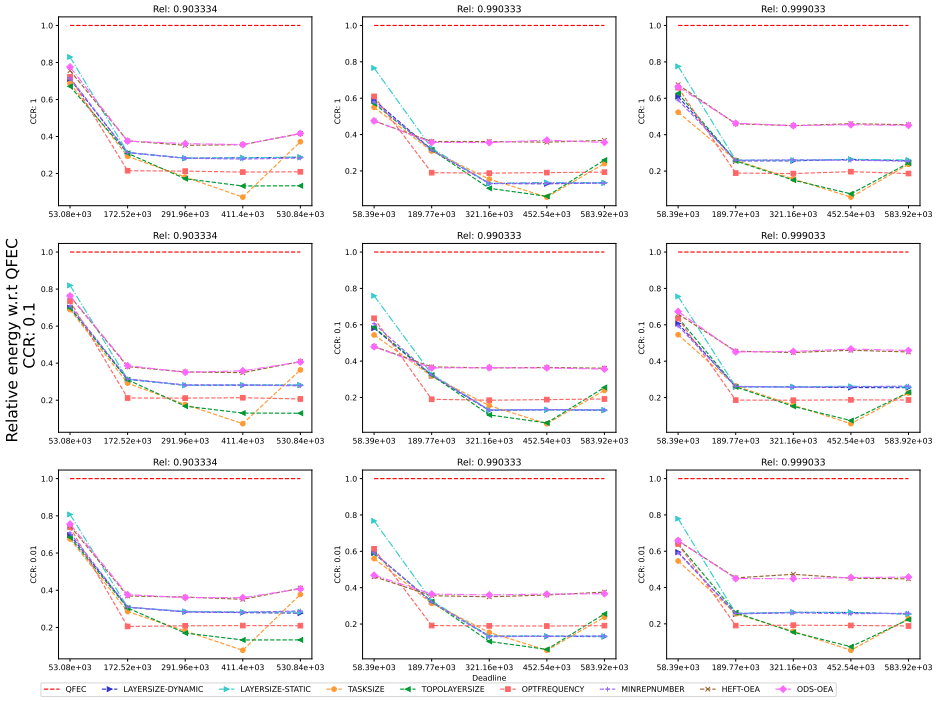

Figure 460: Performance of the different heuristics on the SoyKB workflow.

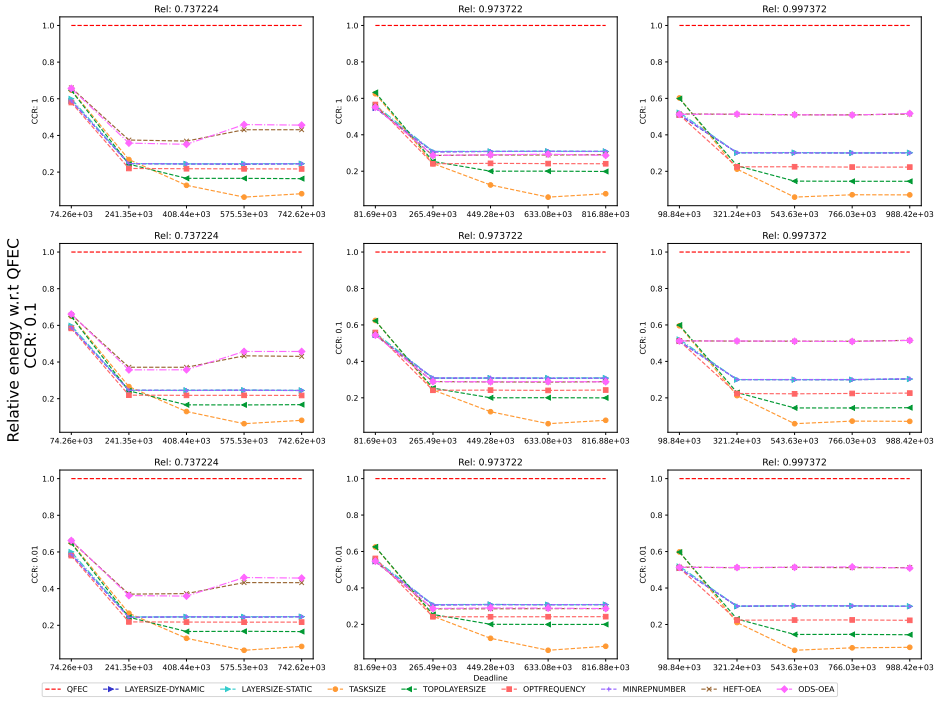

Figure 461: Performance of the different heuristics on the SRASearch.

E.1.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

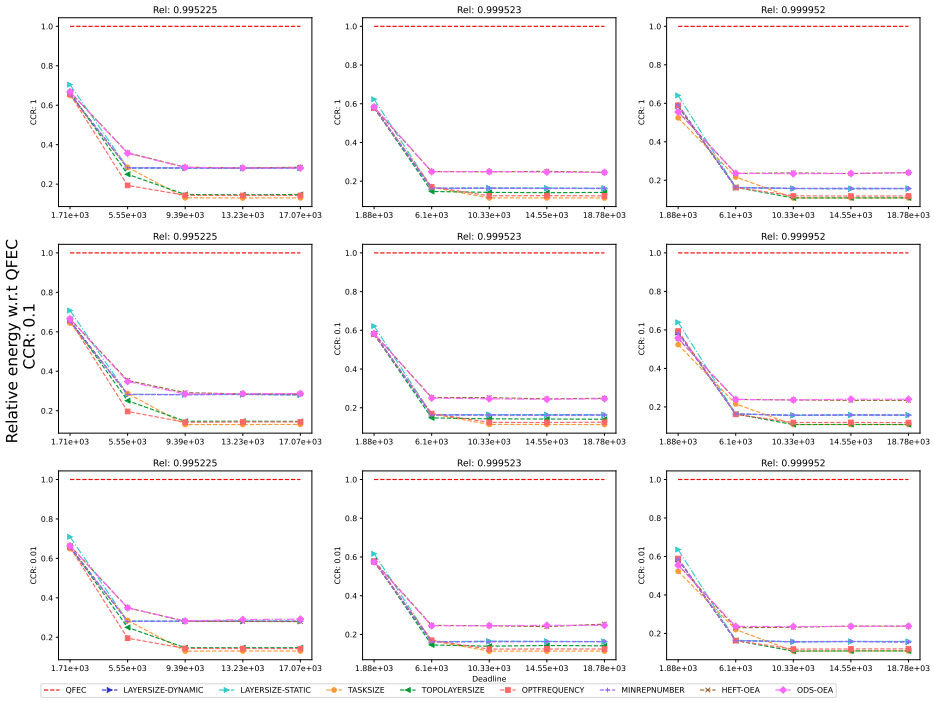

Figure 462: Performance of the different heuristics on the BWA workflow.

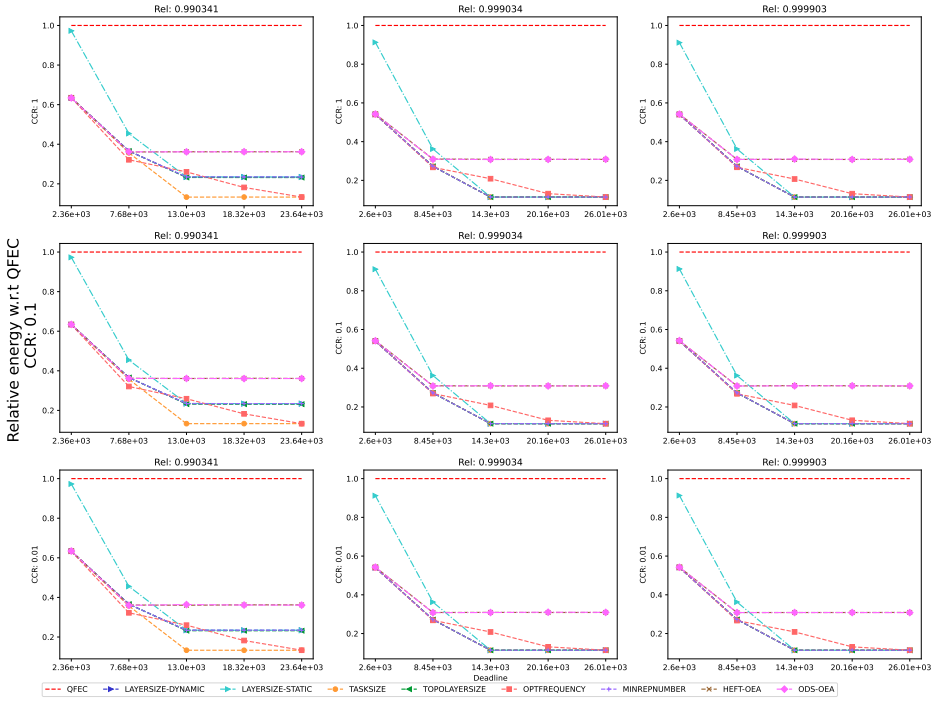

Figure 463: Performance of the different heuristics on the Cholesky workflow.

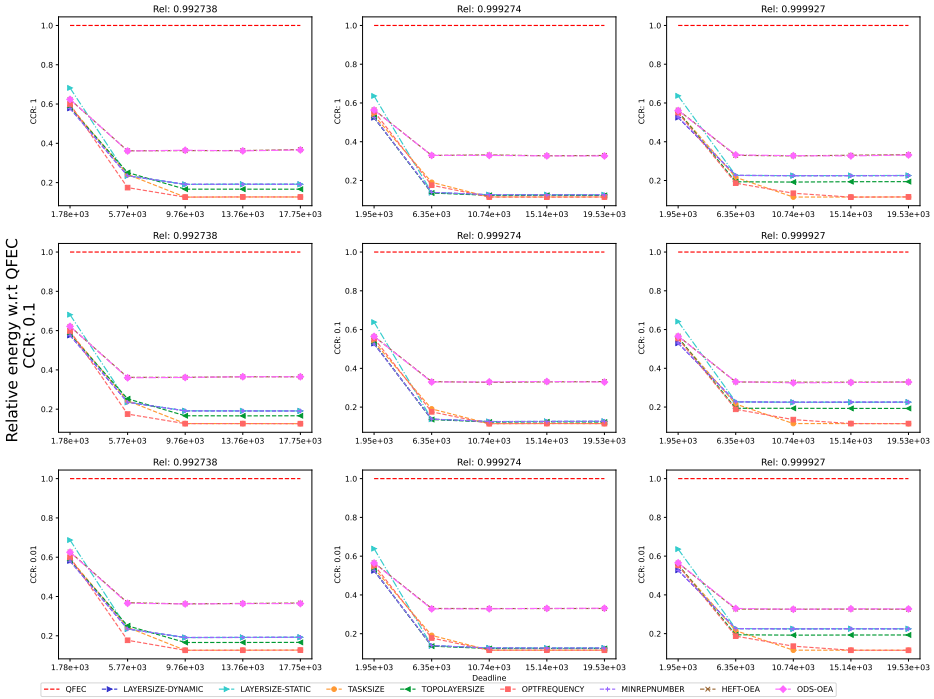

Figure 464: Performance of the different heuristics on the Cycles workflow.

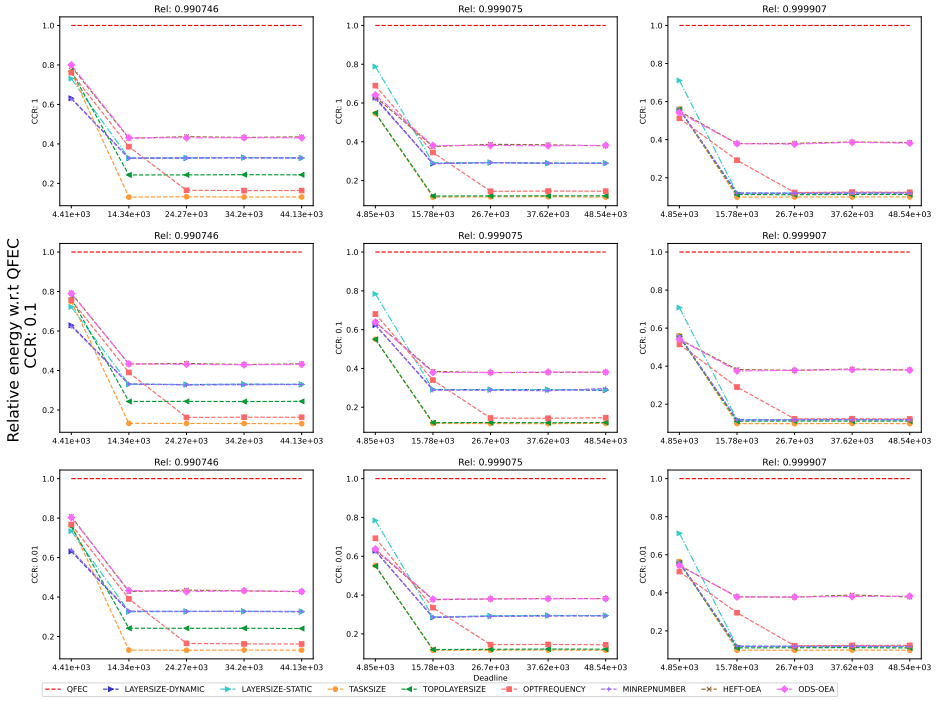

Figure 465: Performance of the different heuristics on the Epigenomics workflow.

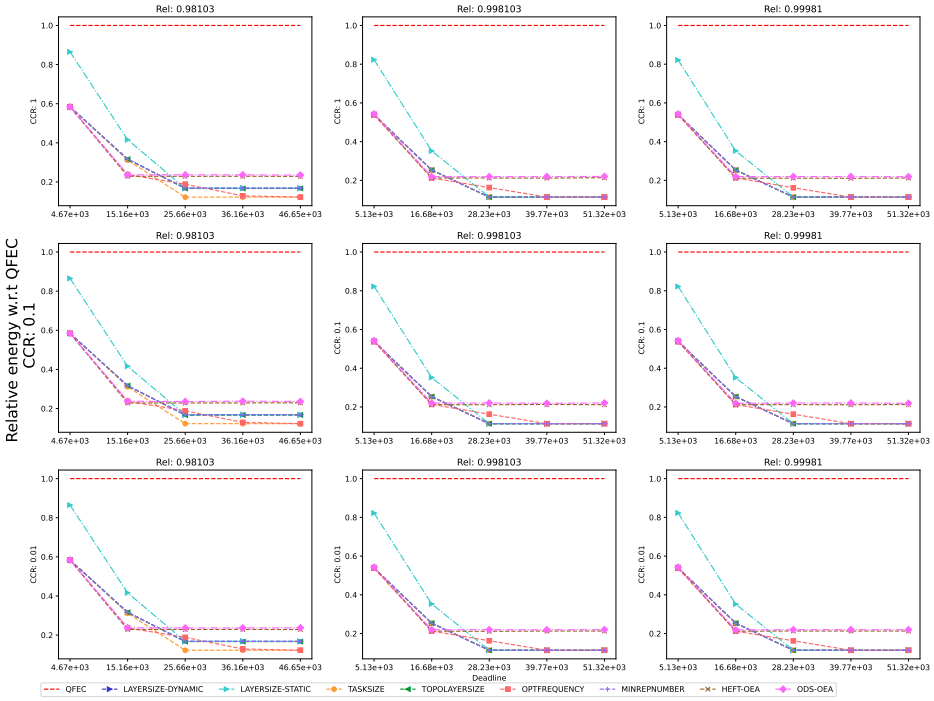

Figure 466: Performance of the different heuristics on the LU workflow.

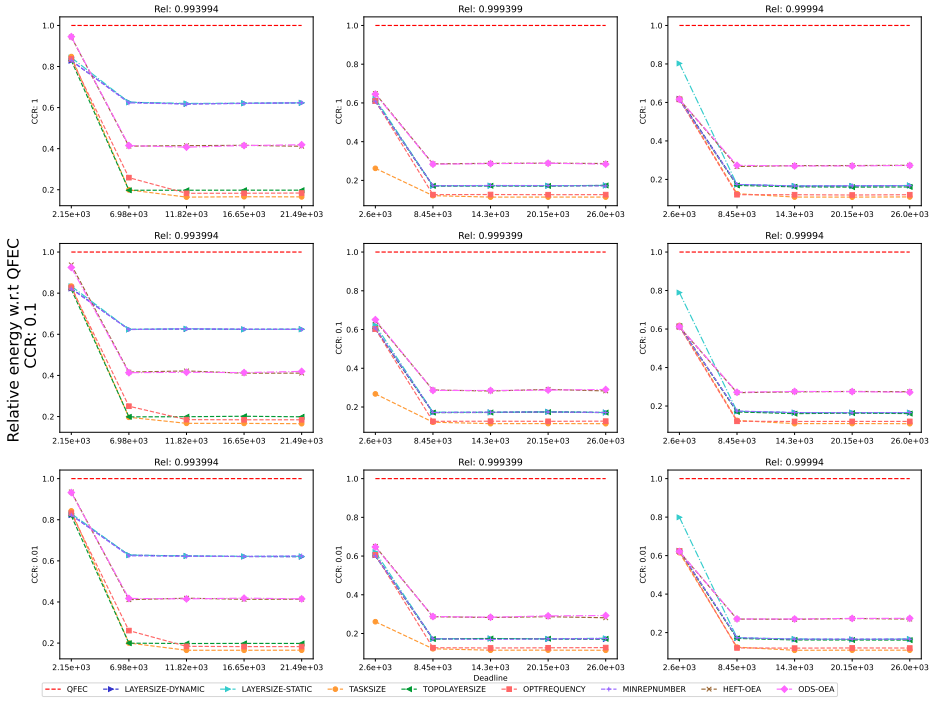

Figure 467: Performance of the different heuristics on the Montage workflow.

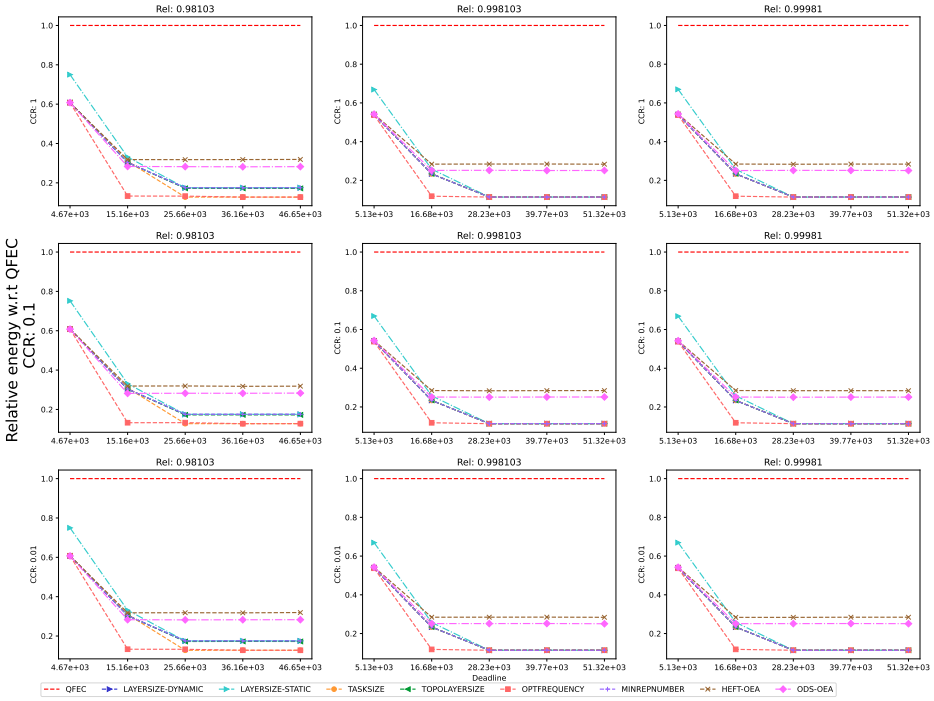

Figure 468: Performance of the different heuristics on the QR workflow.

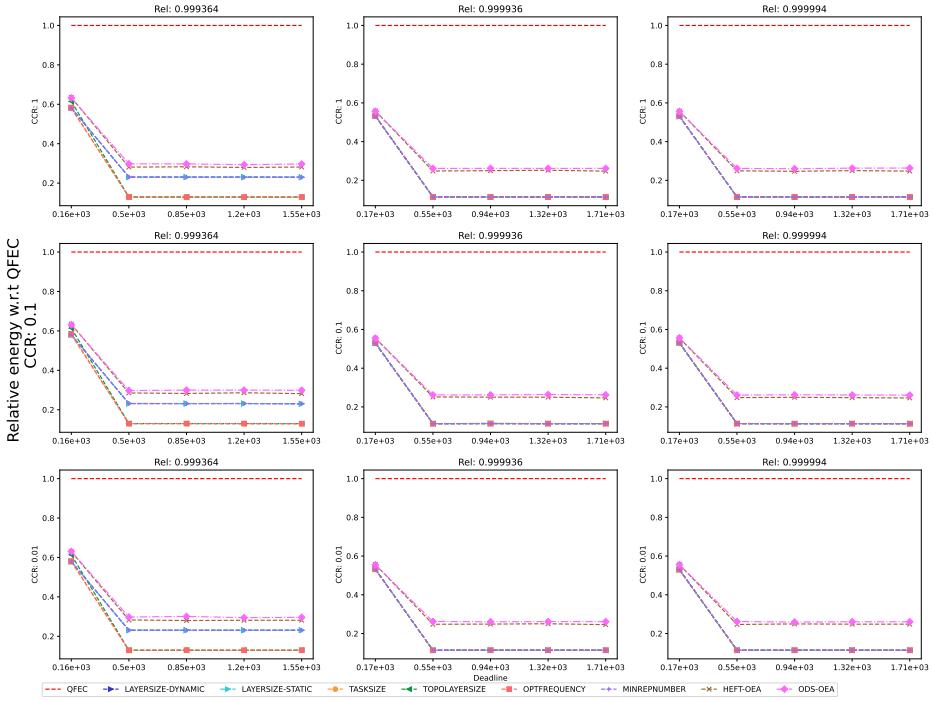

Figure 469: Performance of the different heuristics on the Seismology workflow.

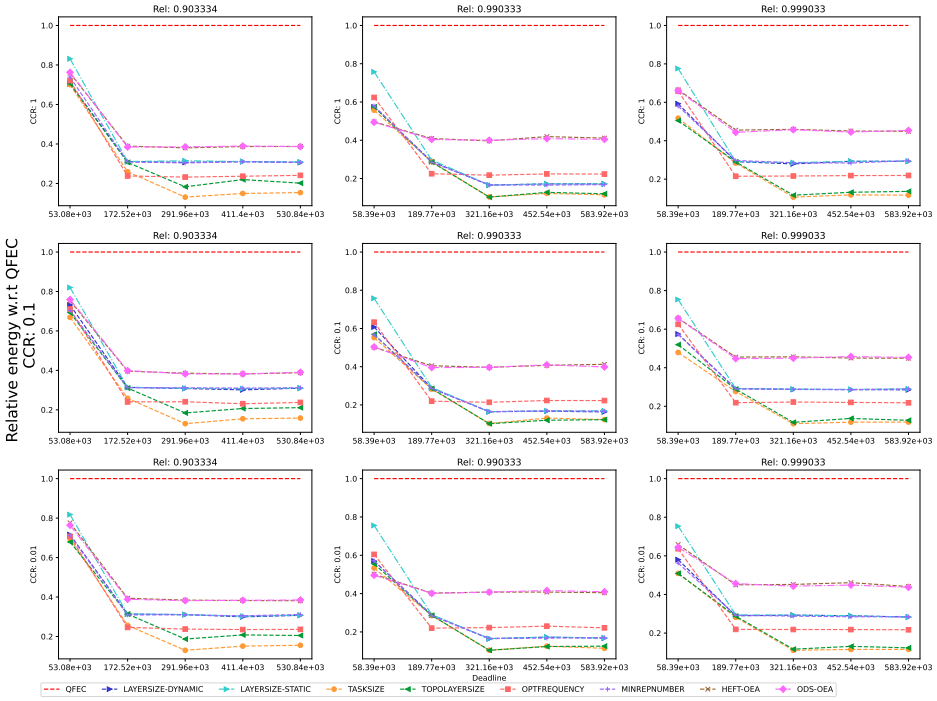

Figure 470: Performance of the different heuristics on the SoyKB workflow.

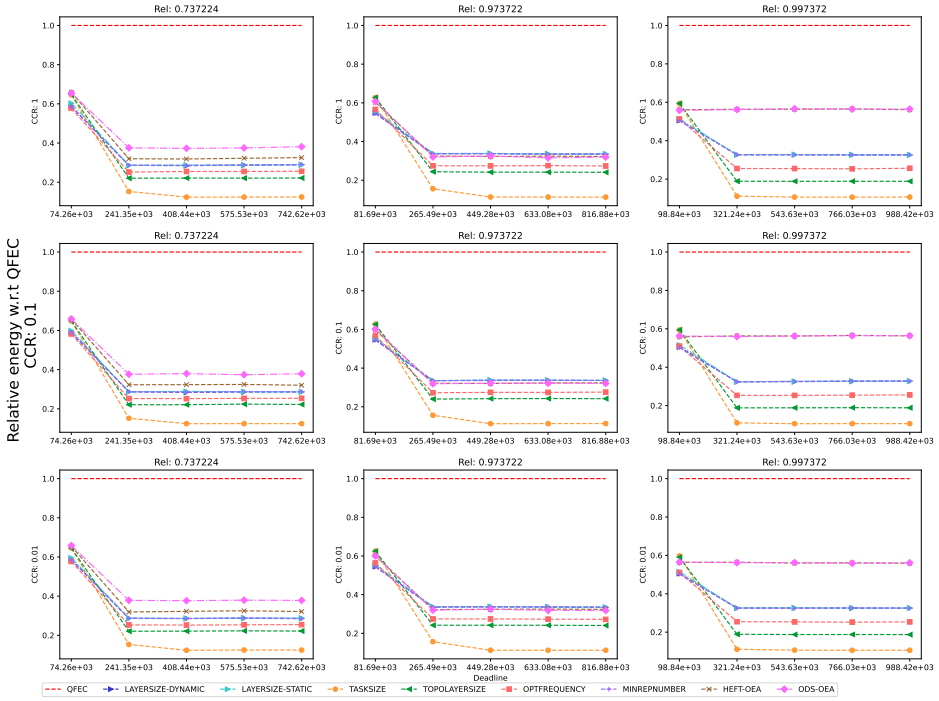

Figure 471: Performance of the different heuristics on the SRASearch.

E.1.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

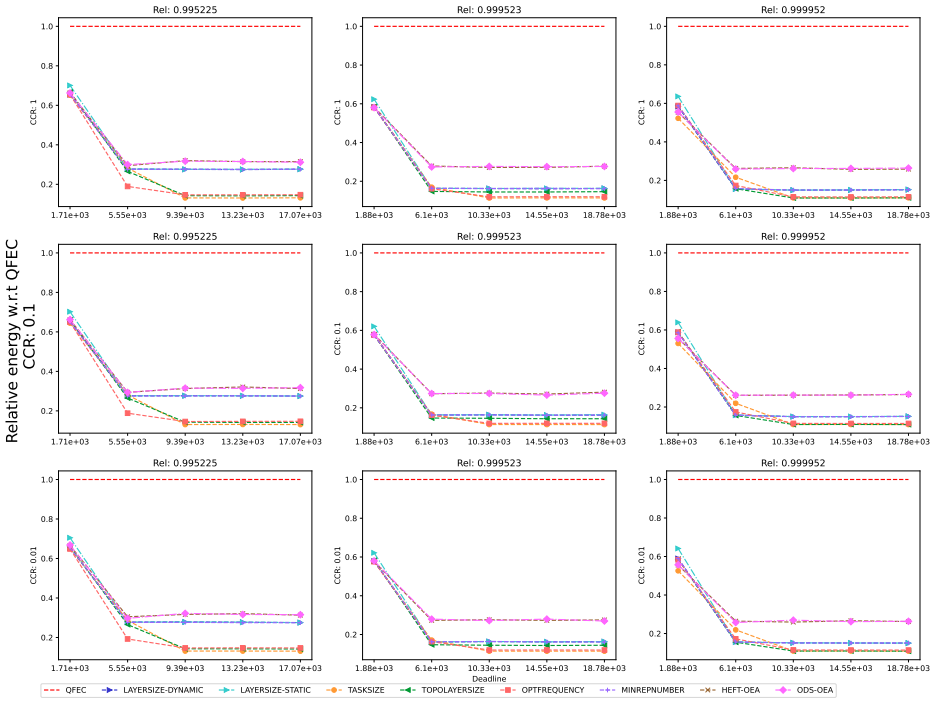

Figure 472: Performance of the different heuristics on the BWA workflow.

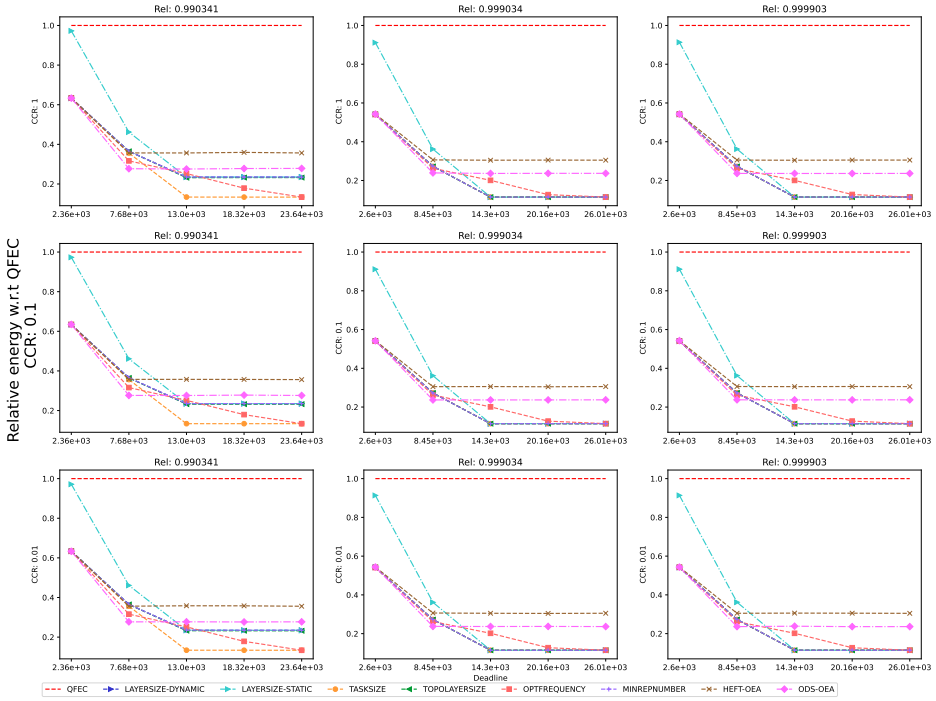

Figure 473: Performance of the different heuristics on the Cholesky workflow.

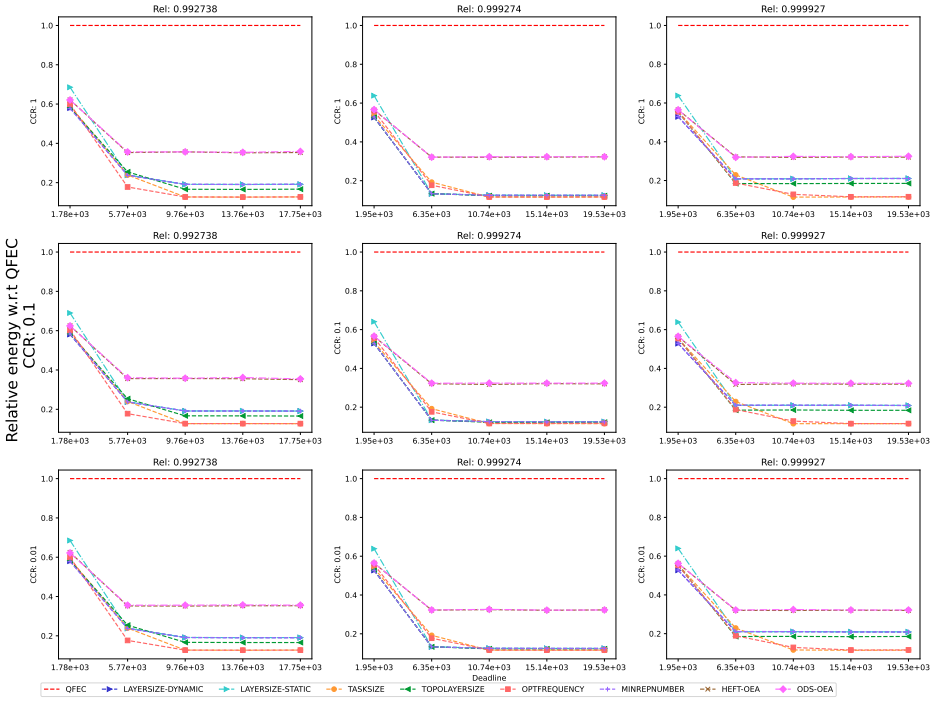

Figure 474: Performance of the different heuristics on the Cycles workflow.

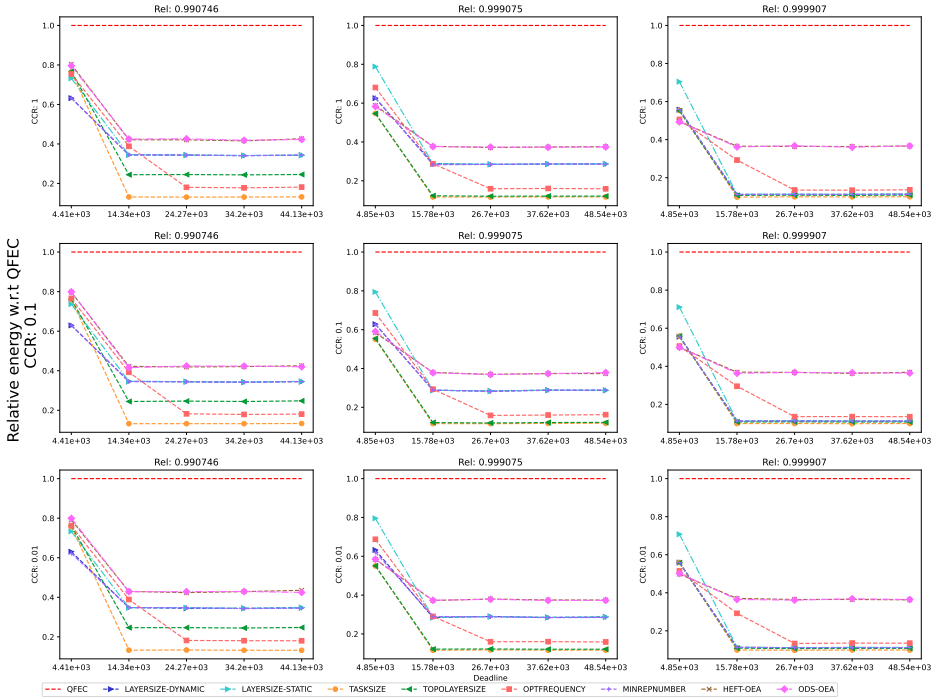

Figure 475: Performance of the different heuristics on the Epigenomics workflow.

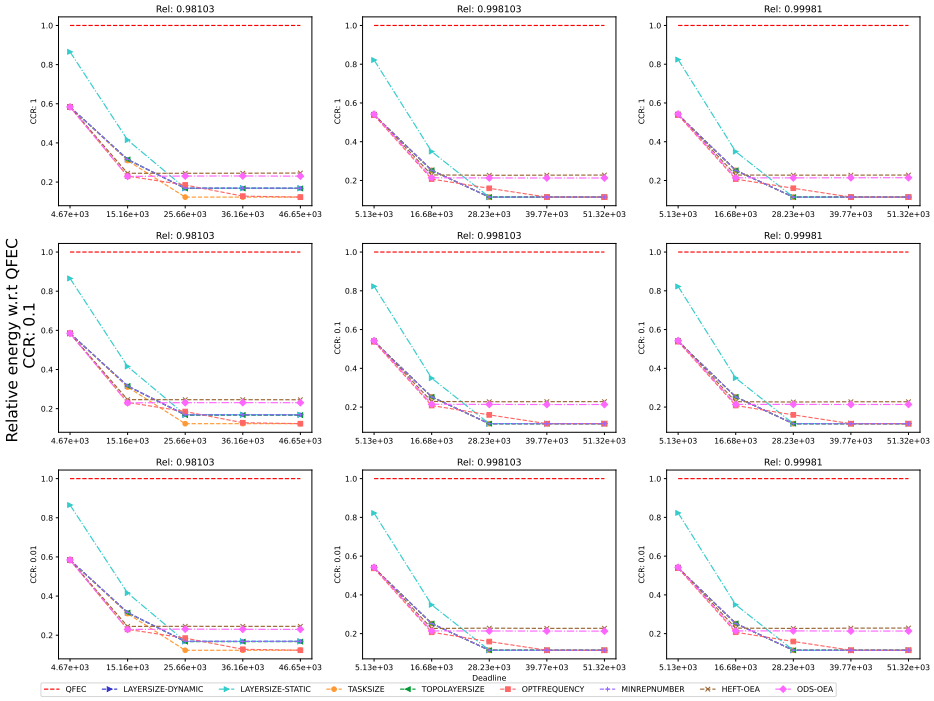

Figure 476: Performance of the different heuristics on the LU workflow.

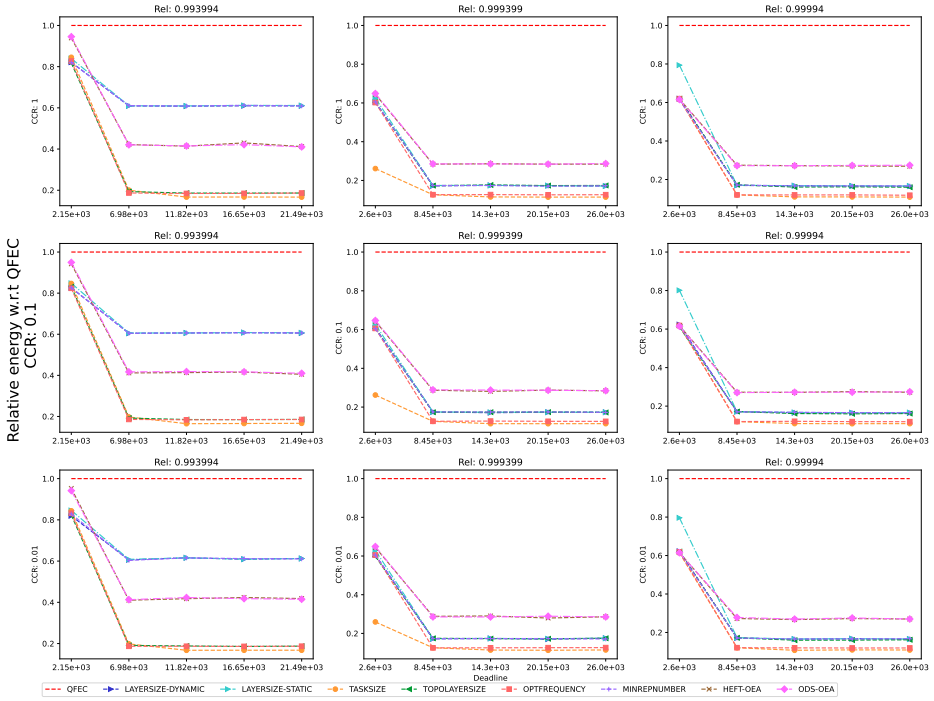

Figure 477: Performance of the different heuristics on the Montage workflow.

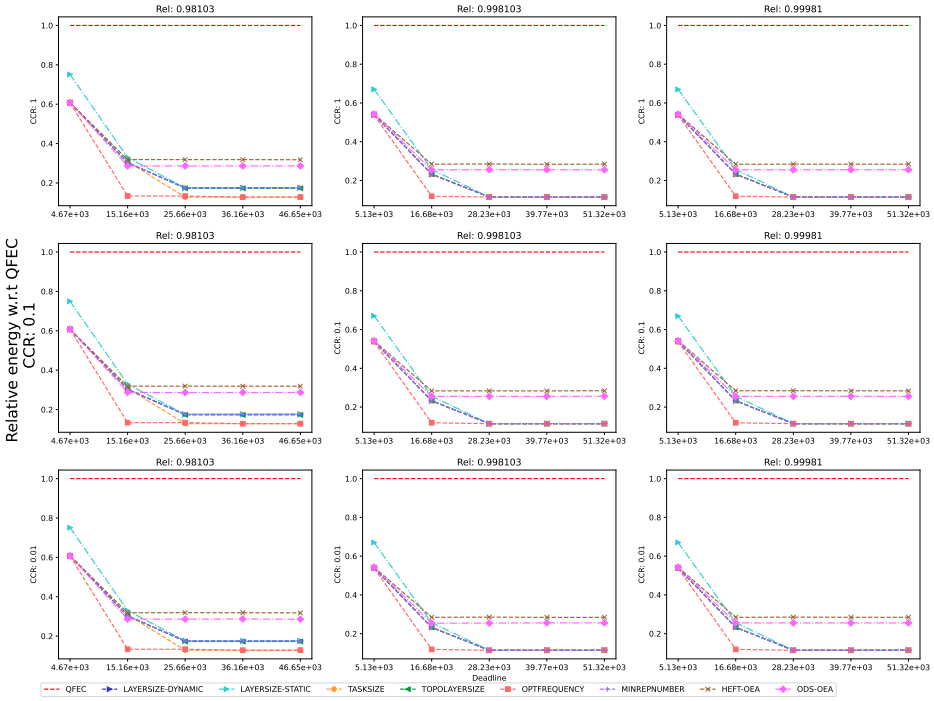

Figure 478: Performance of the different heuristics on the QR workflow.

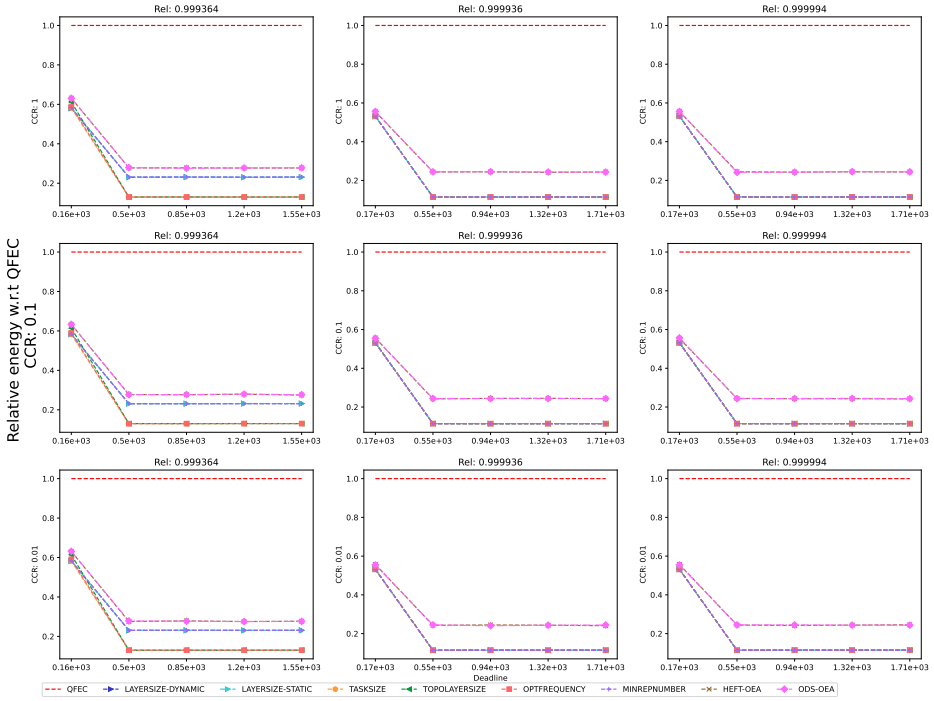

Figure 479: Performance of the different heuristics on the Seismology workflow.

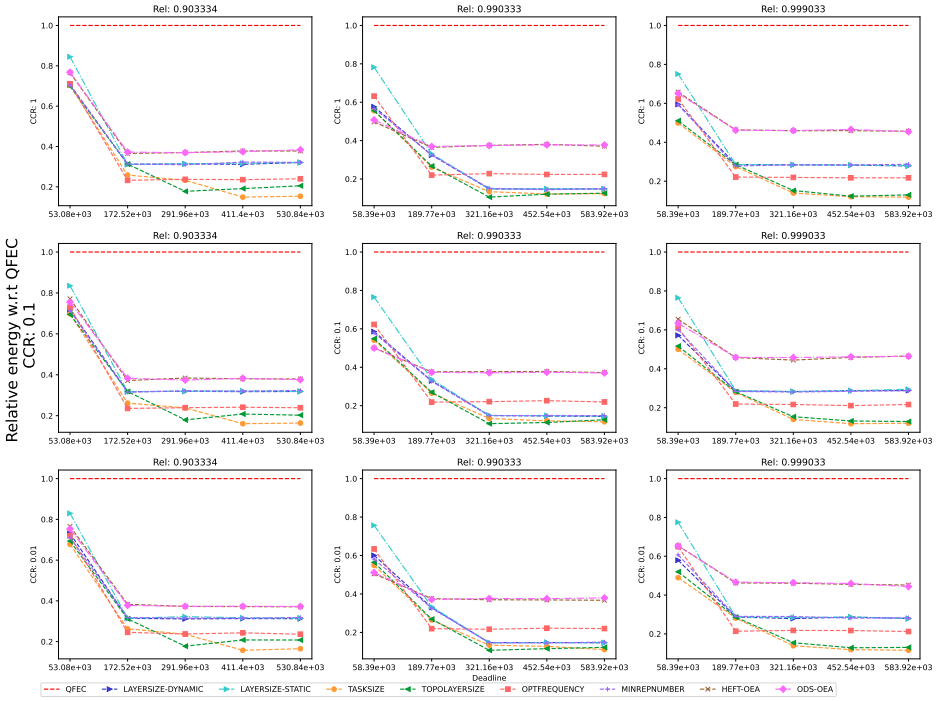

Figure 480: Performance of the different heuristics on the SoyKB workflow.

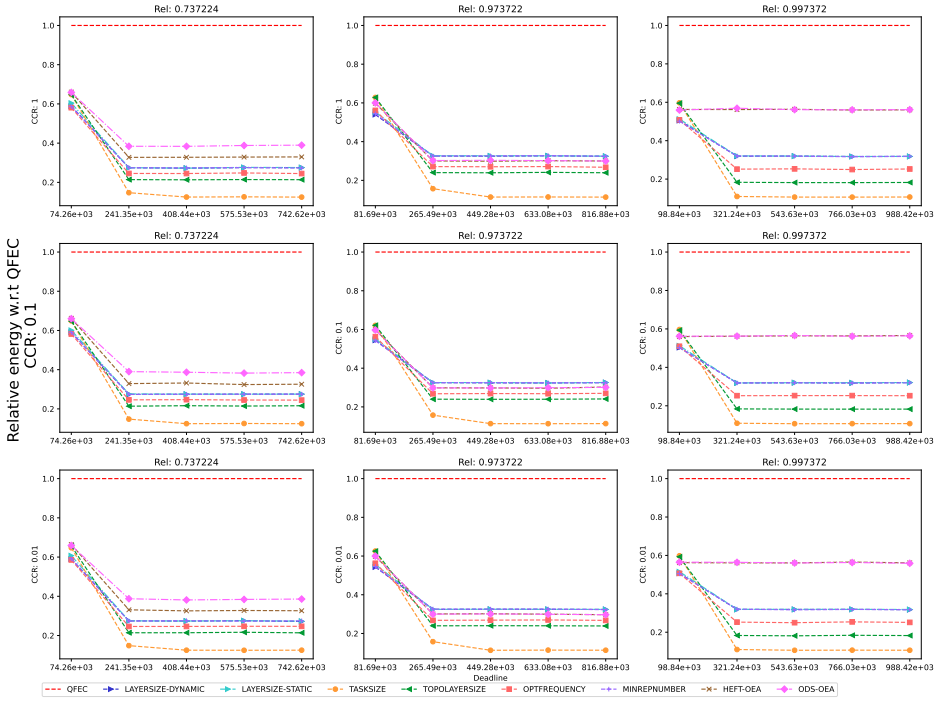

Figure 481: Performance of the different heuristics on the SRASearch.

E.2 $BC/WC = 0.2$

E.2.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

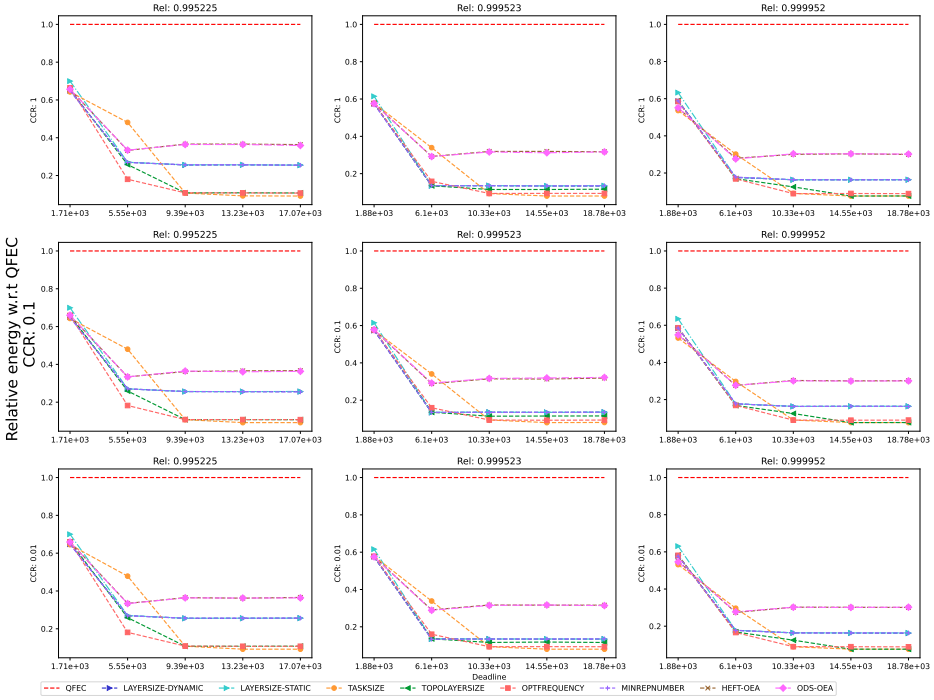

Figure 482: Performance of the different heuristics on the BWA workflow.

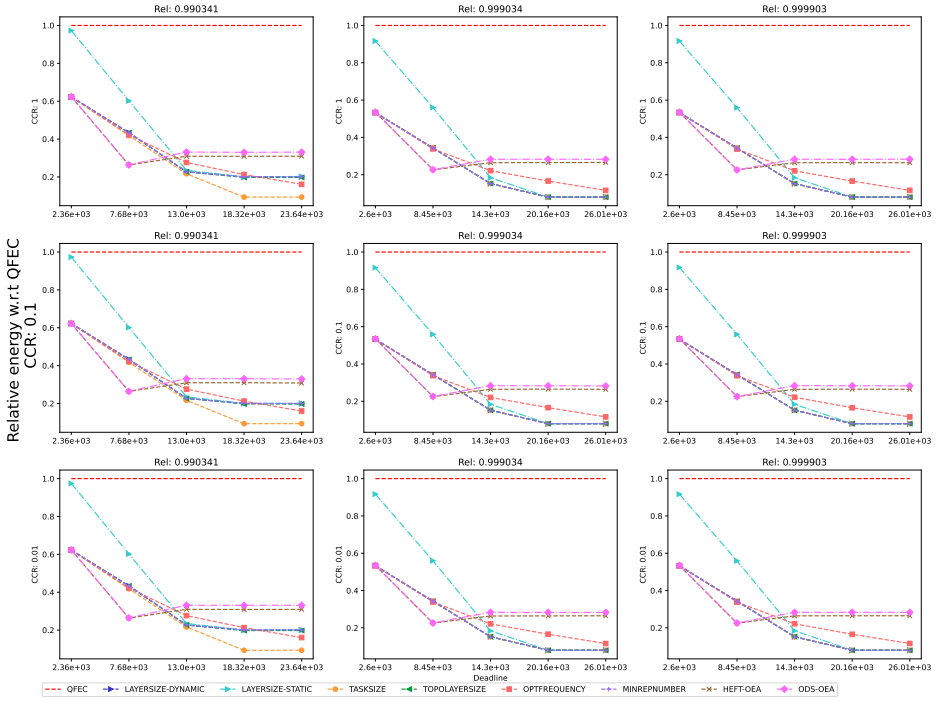

Figure 483: Performance of the different heuristics on the Cholesky workflow.

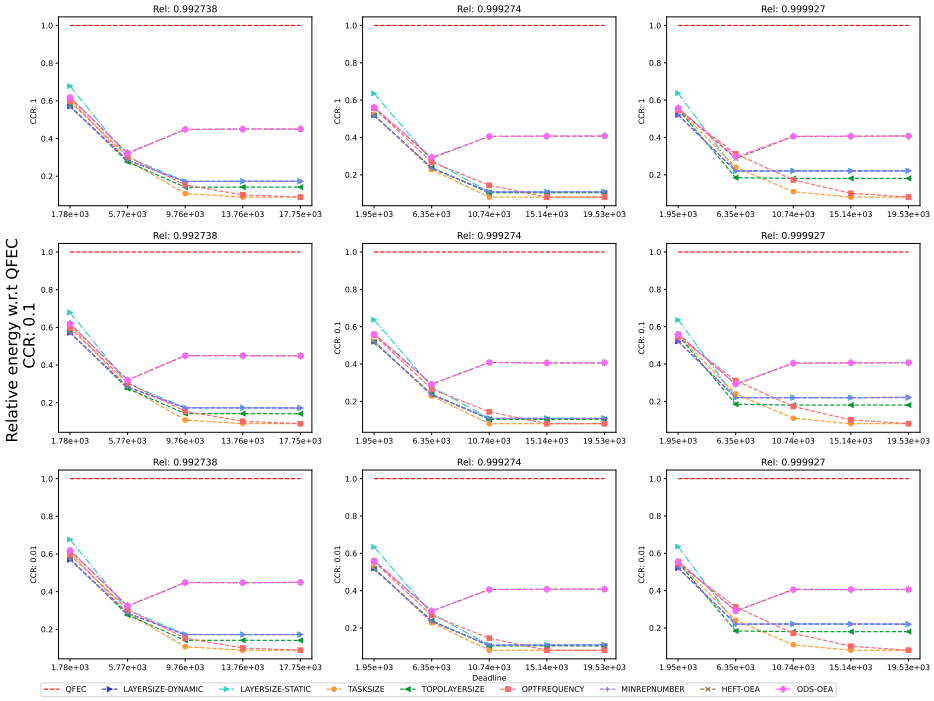

Figure 484: Performance of the different heuristics on the Cycles workflow.

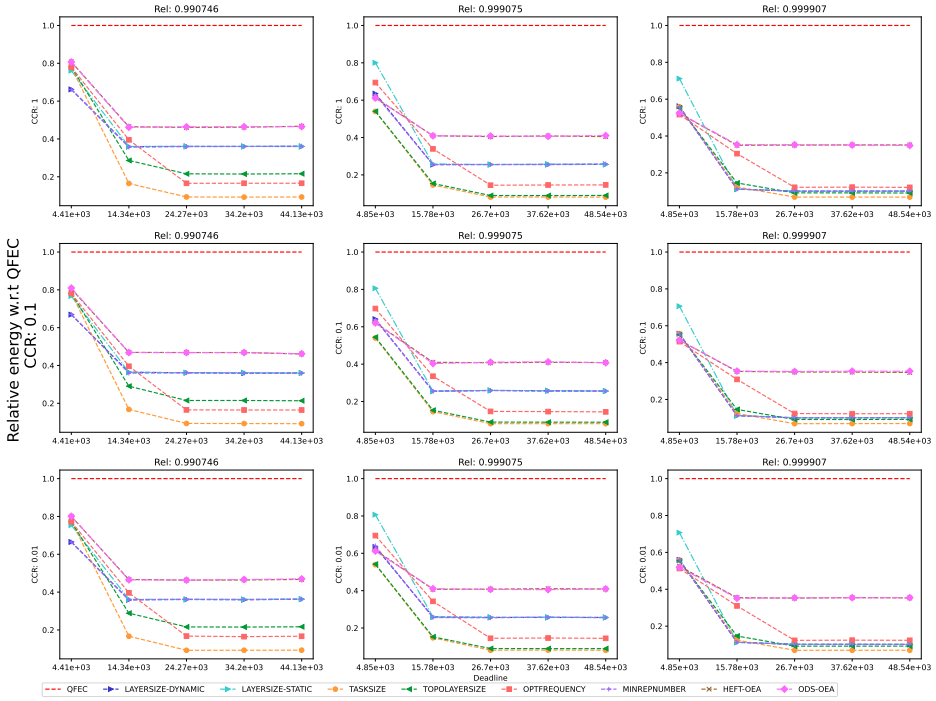

Figure 485: Performance of the different heuristics on the Epigenomics workflow.

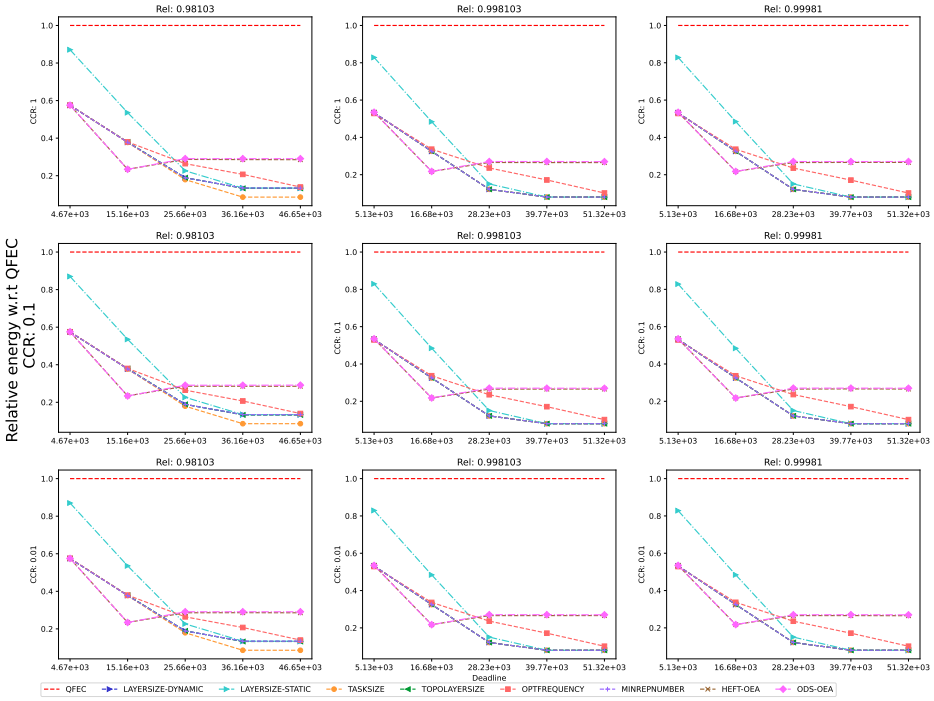

Figure 486: Performance of the different heuristics on the LU workflow.

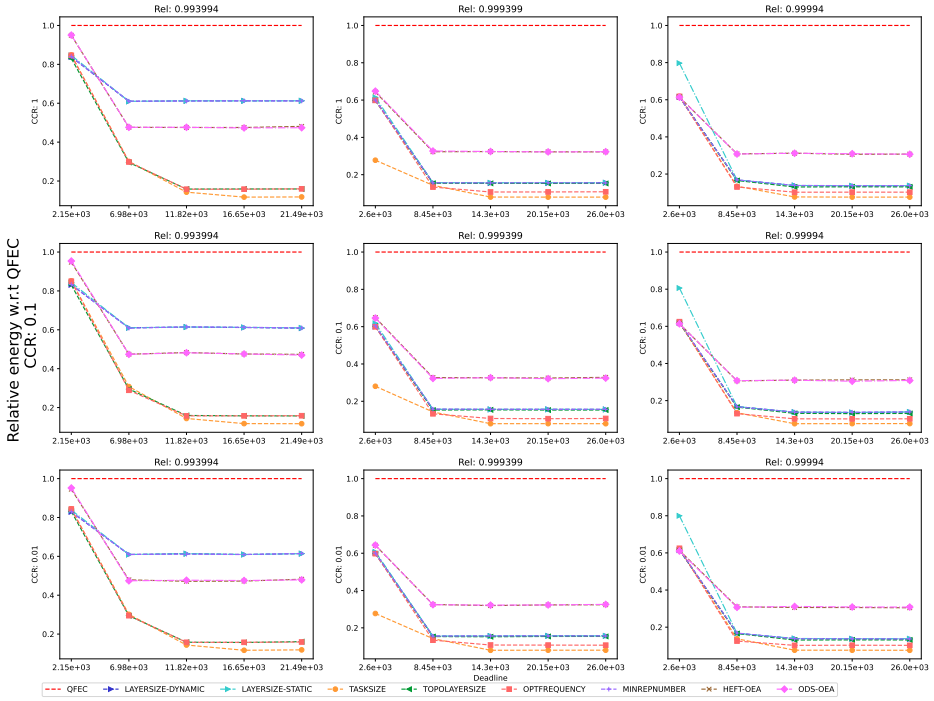

Figure 487: Performance of the different heuristics on the Montage workflow.

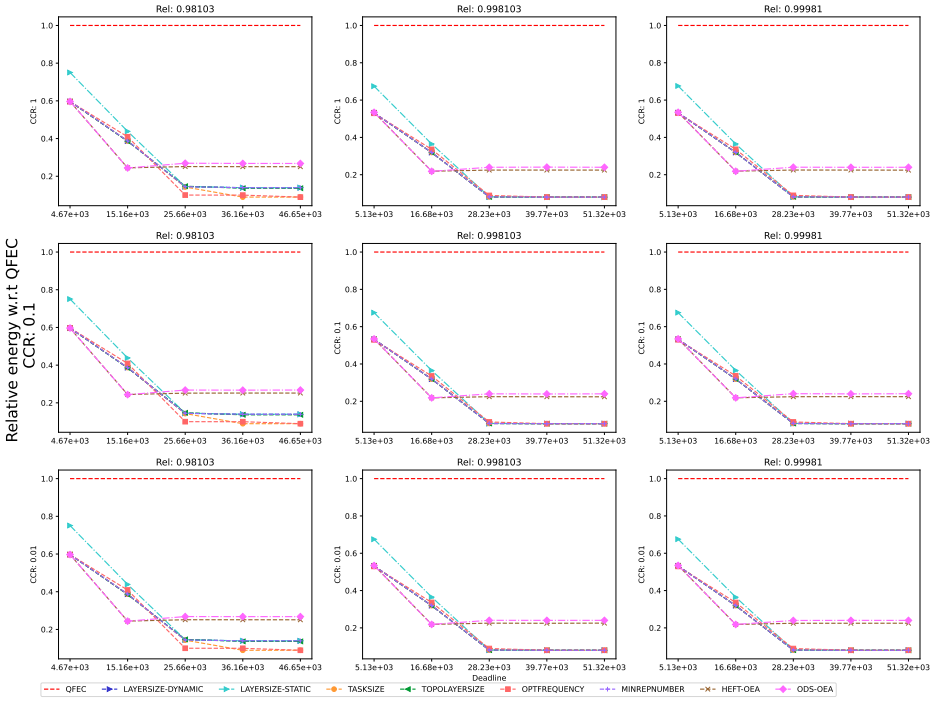

Figure 488: Performance of the different heuristics on the QR workflow.

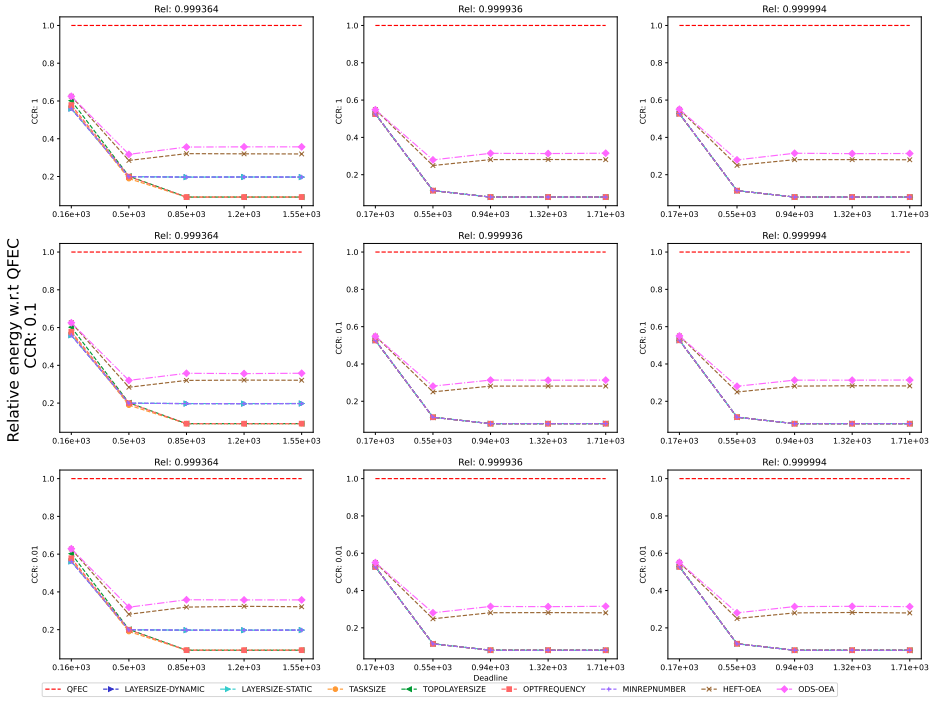

Figure 489: Performance of the different heuristics on the Seismology workflow.

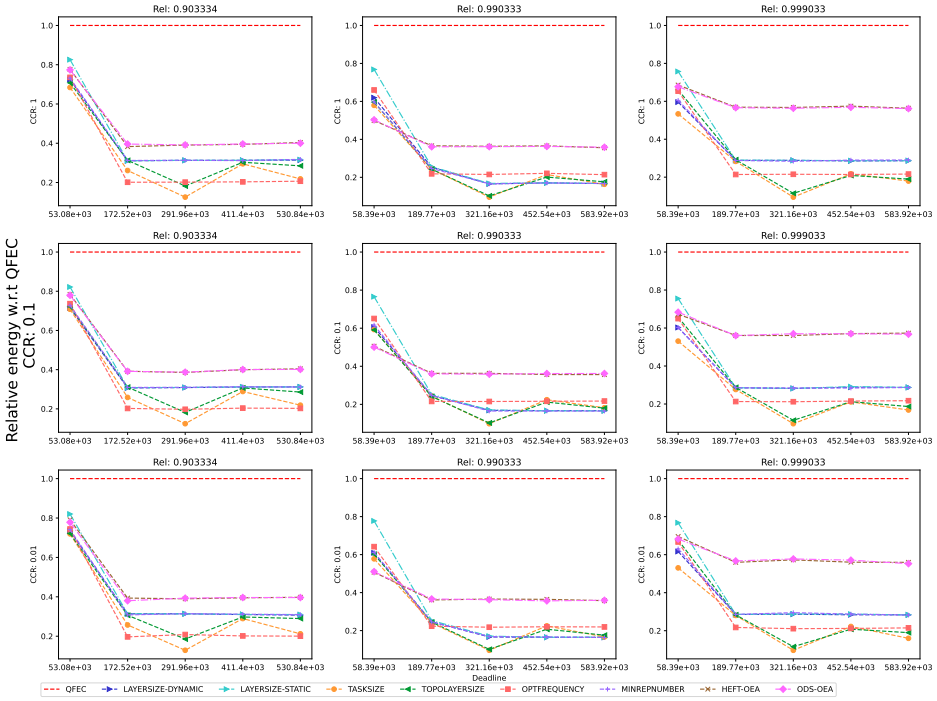

Figure 490: Performance of the different heuristics on the SoyKB workflow.

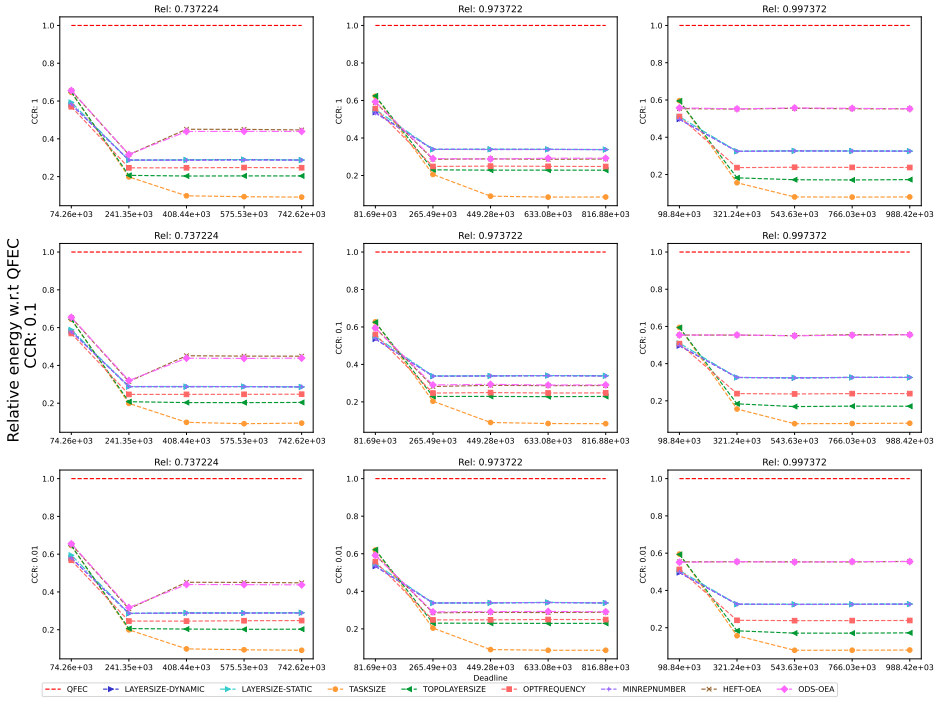

Figure 491: Performance of the different heuristics on the SRASearch.

E.2.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

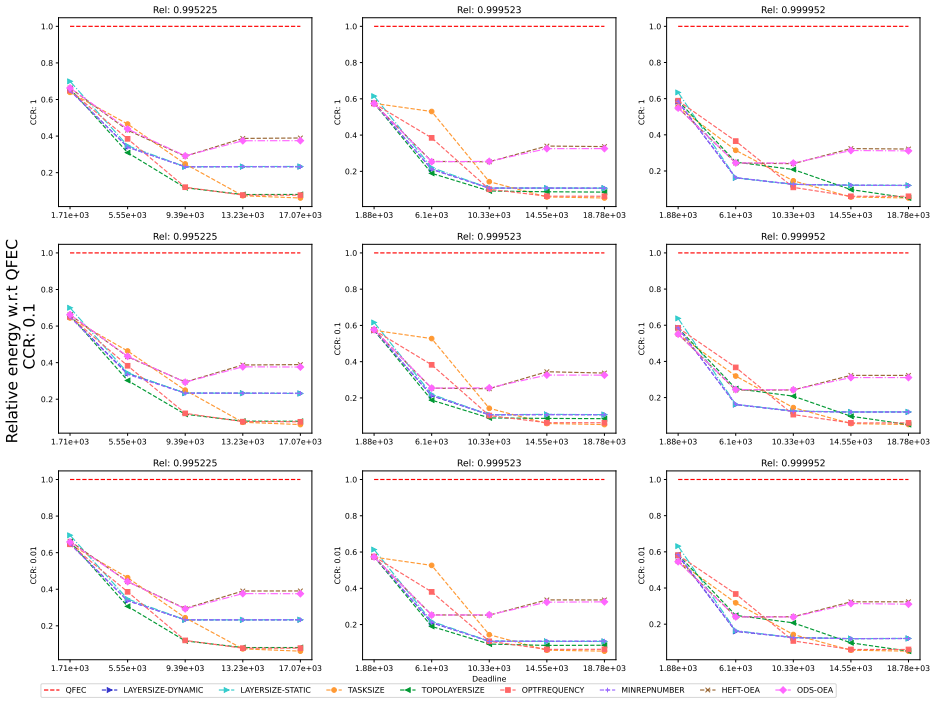

Figure 492: Performance of the different heuristics on the BWA workflow.

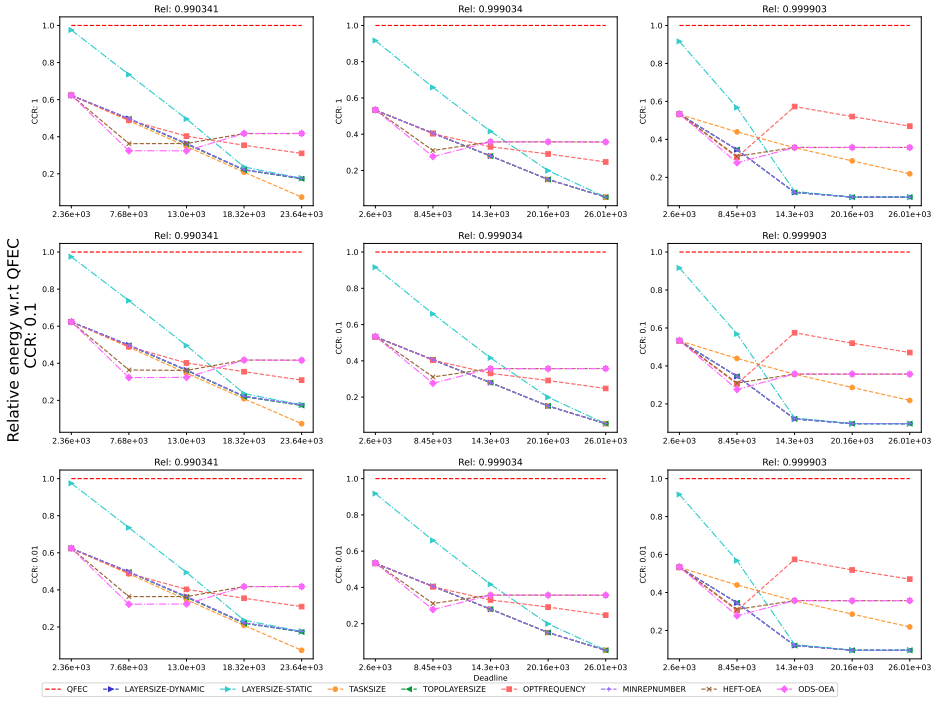

Figure 493: Performance of the different heuristics on the Cholesky workflow.

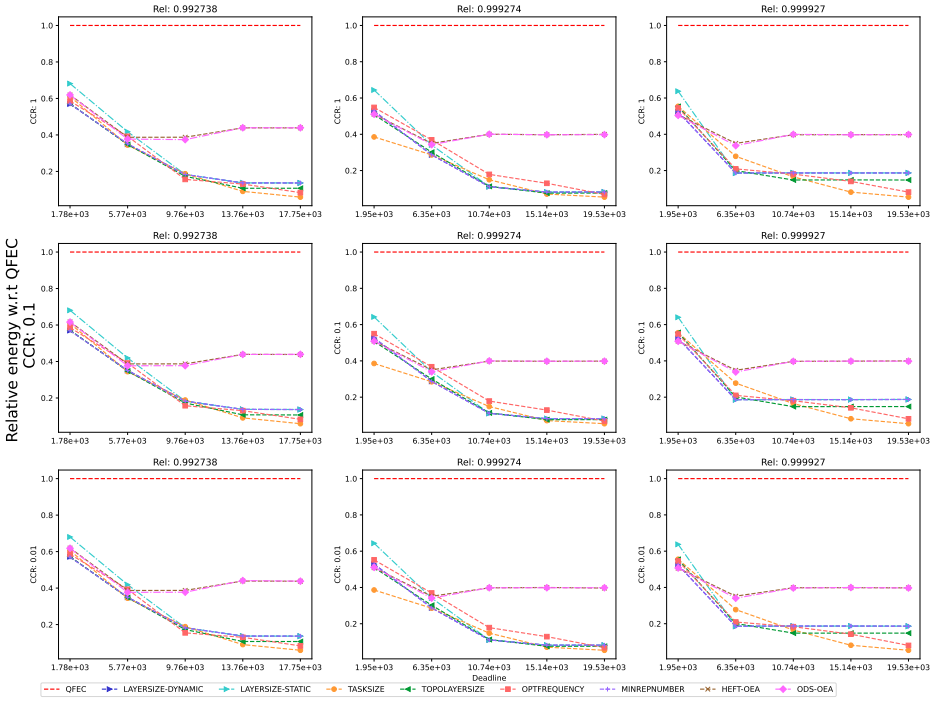

Figure 494: Performance of the different heuristics on the Cycles workflow.

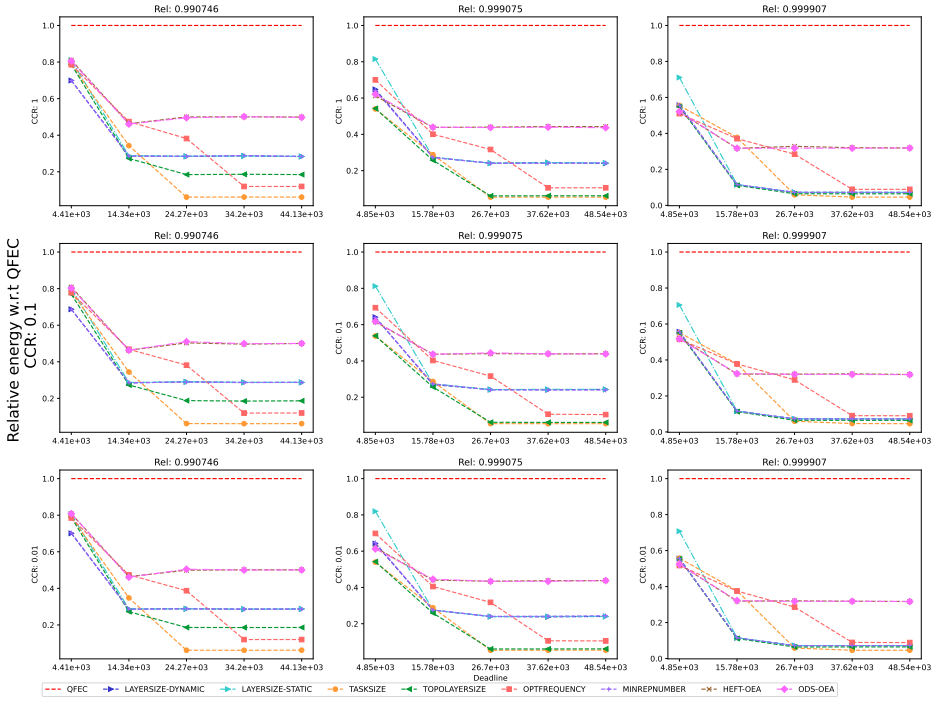

Figure 495: Performance of the different heuristics on the Epigenomics workflow.

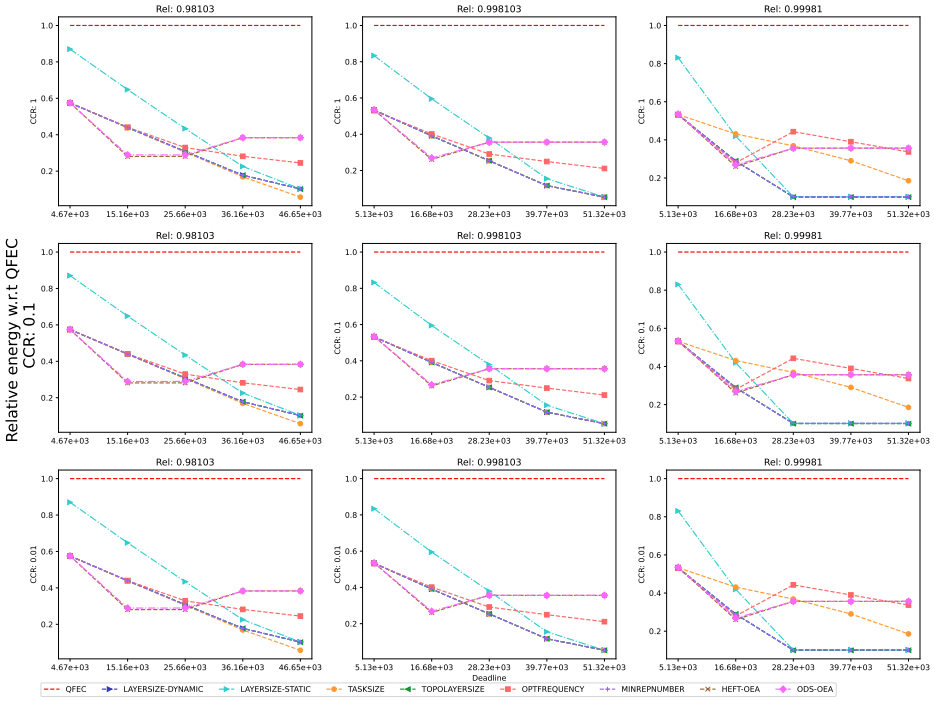

Figure 496: Performance of the different heuristics on the LU workflow.

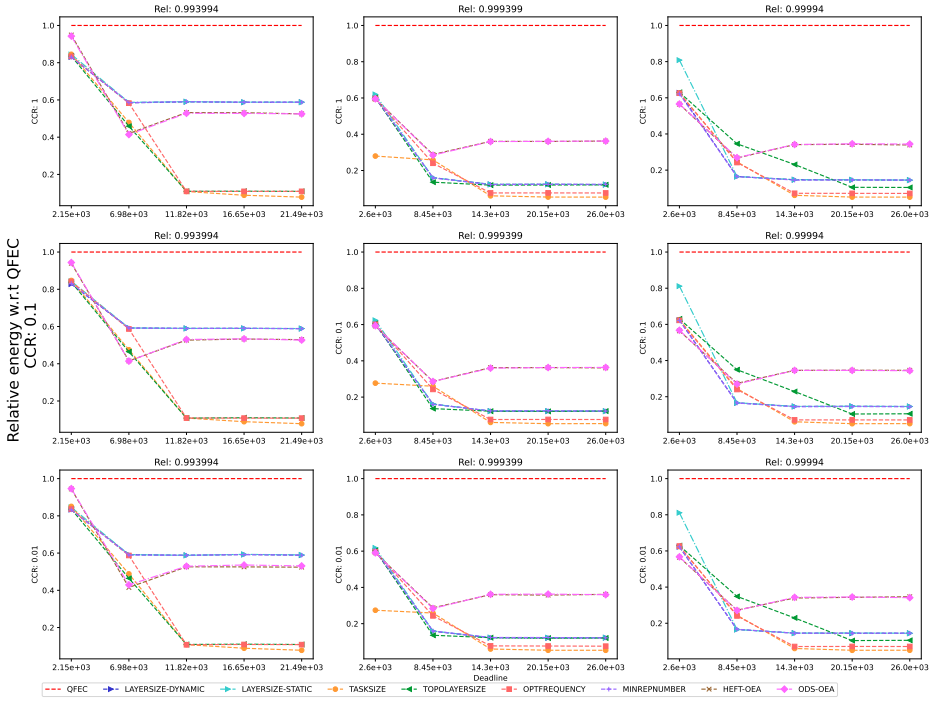

Figure 497: Performance of the different heuristics on the Montage workflow.

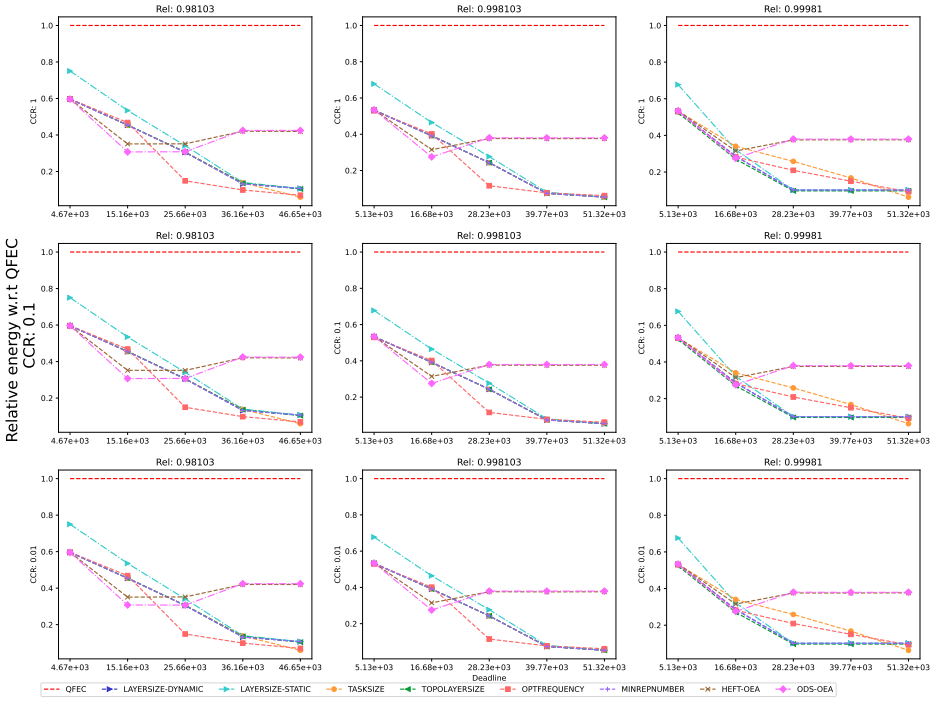

Figure 498: Performance of the different heuristics on the QR workflow.

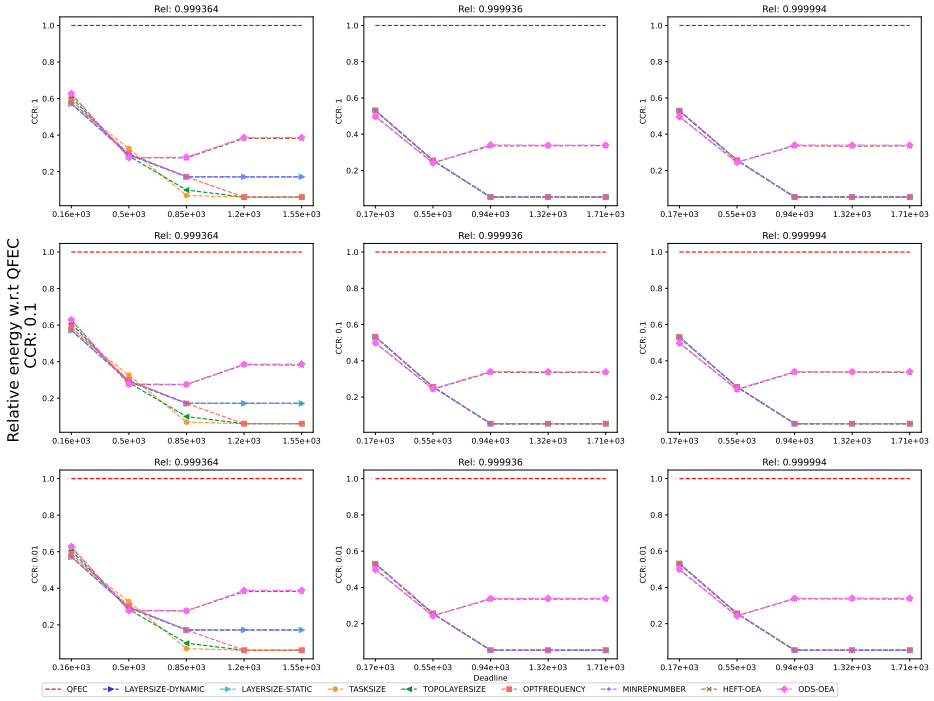

Figure 499: Performance of the different heuristics on the Seismology workflow.

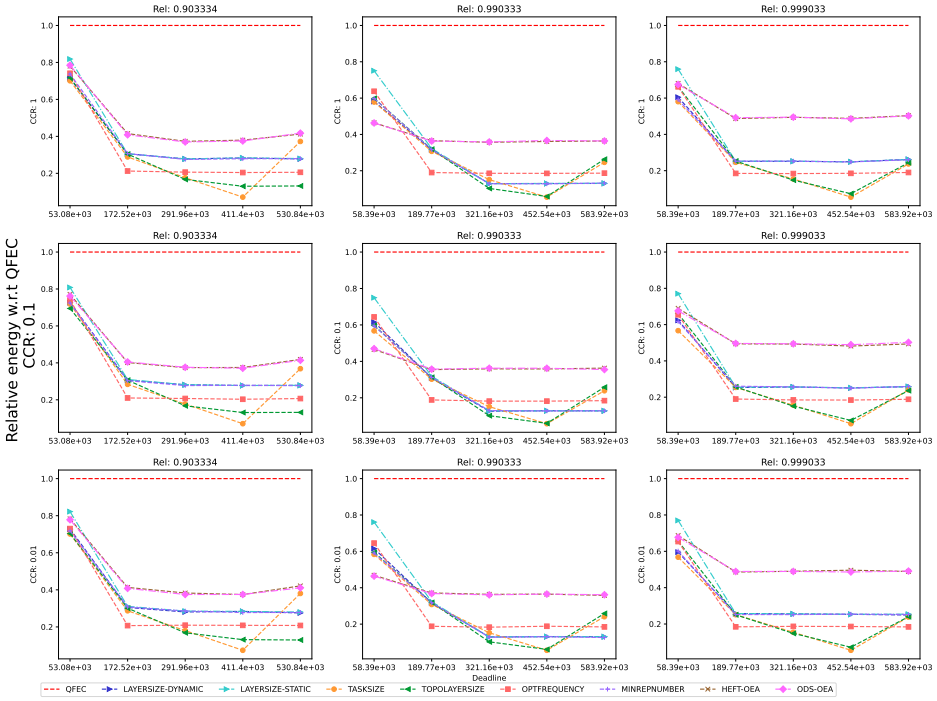

Figure 500: Performance of the different heuristics on the SoyKB workflow.

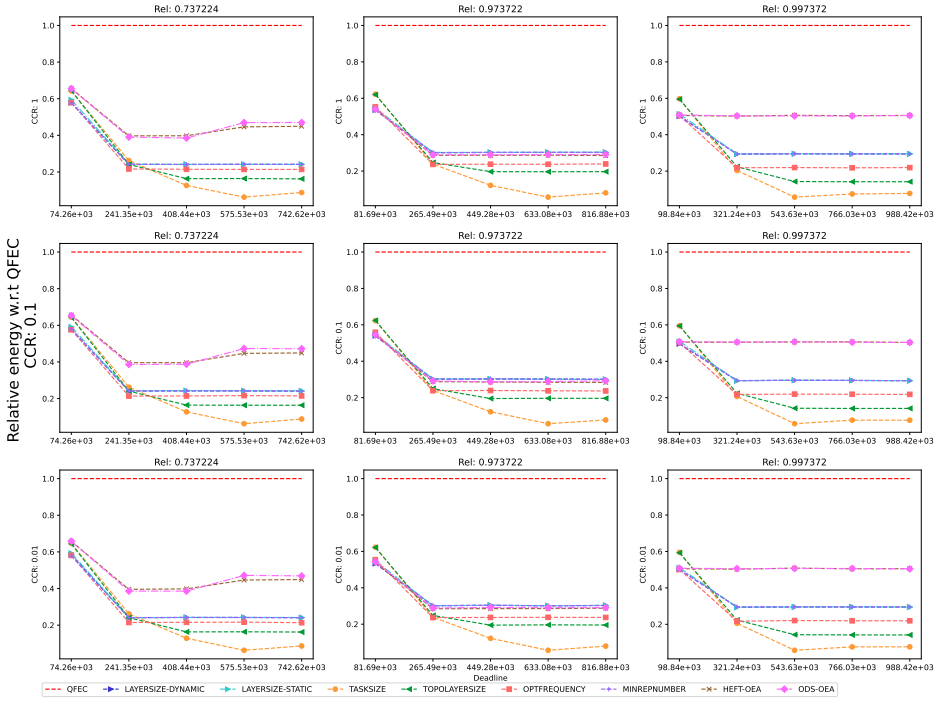

Figure 501: Performance of the different heuristics on the SRASearch.

E.2.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

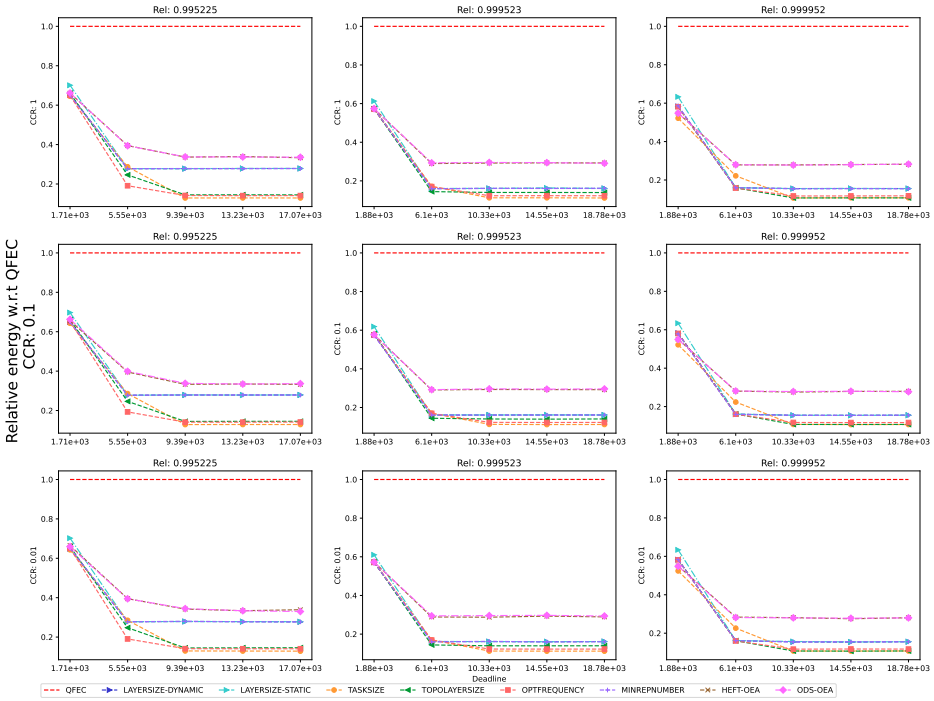

Figure 502: Performance of the different heuristics on the BWA workflow.

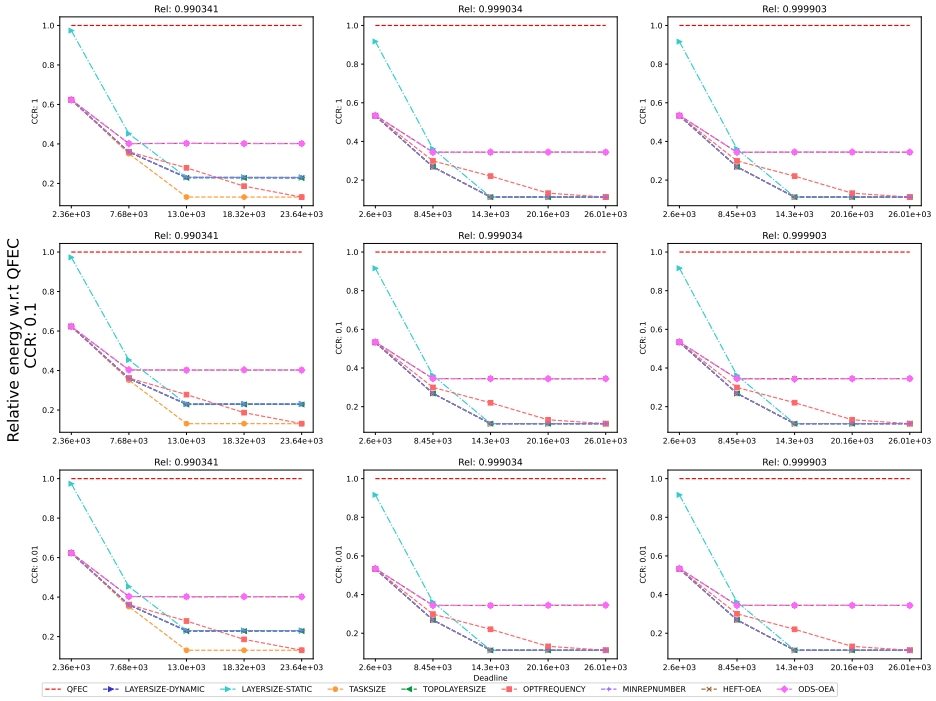

Figure 503: Performance of the different heuristics on the Cholesky workflow.

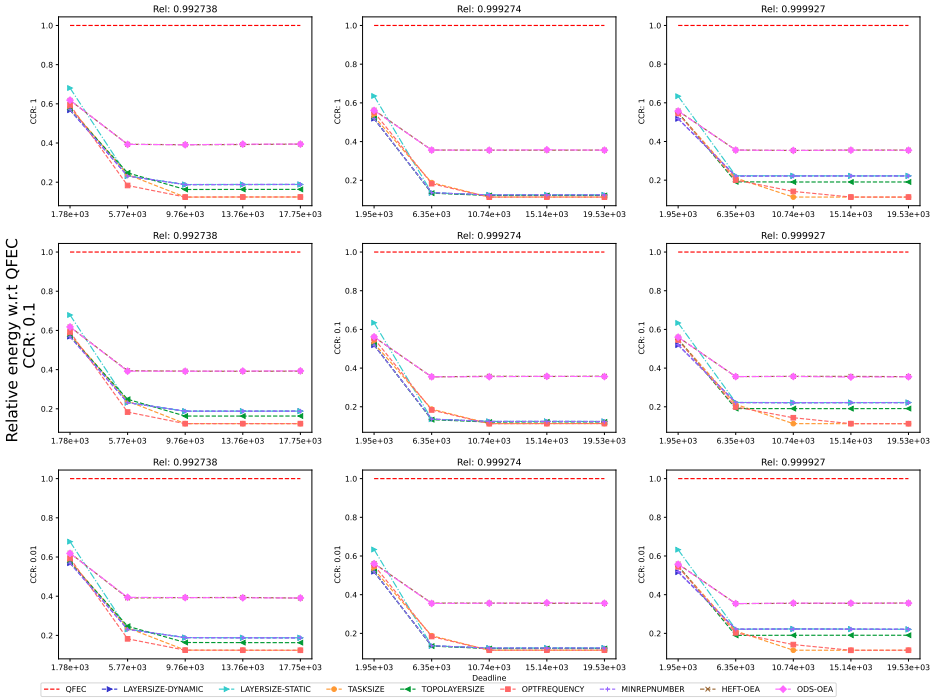

Figure 504: Performance of the different heuristics on the Cycles workflow.

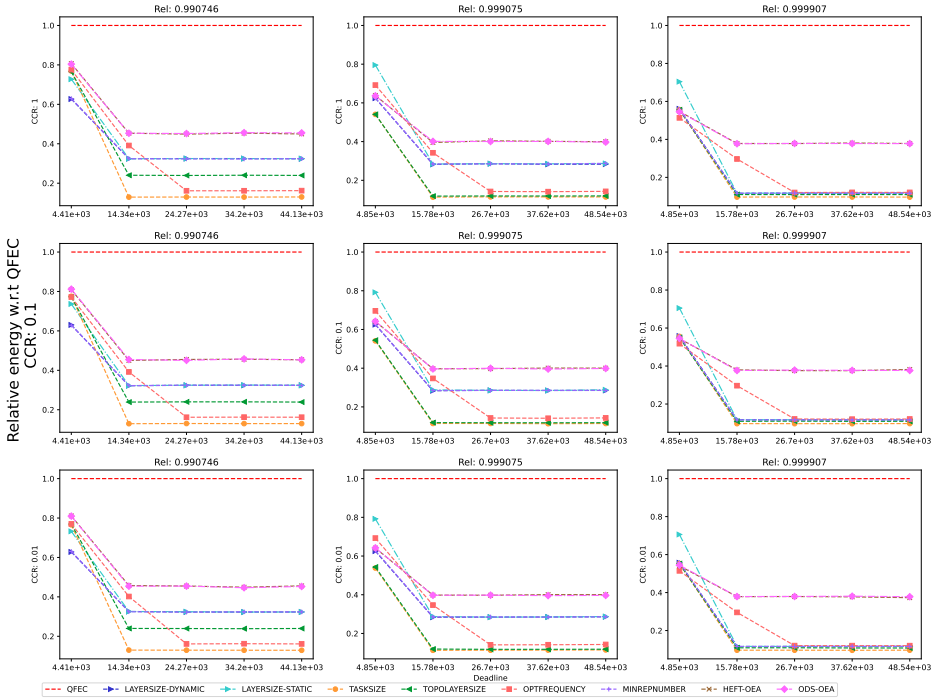

Figure 505: Performance of the different heuristics on the Epigenomics workflow.

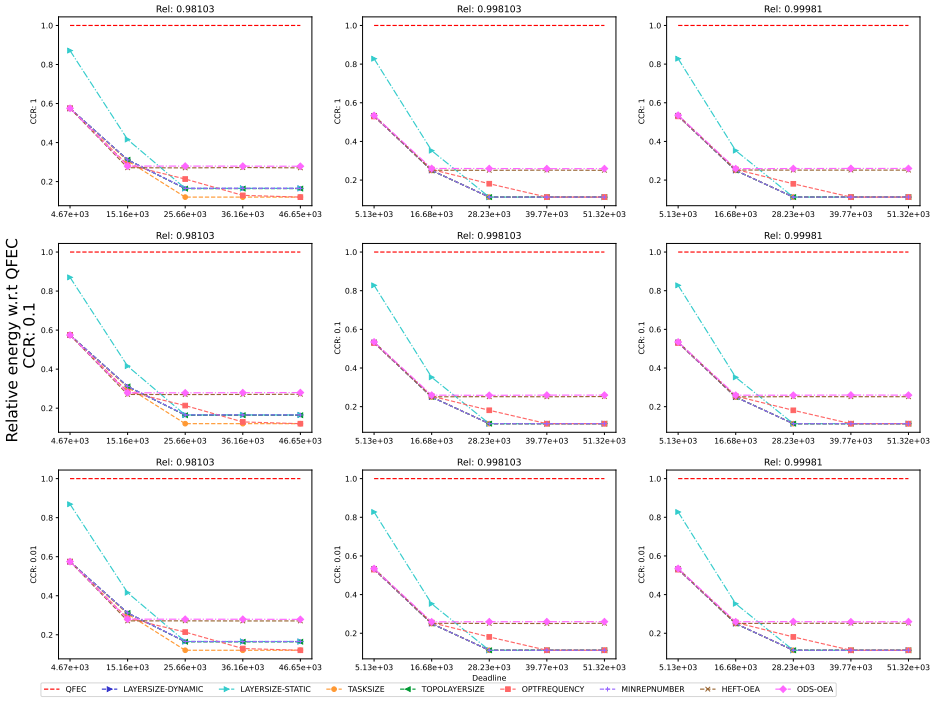

Figure 506: Performance of the different heuristics on the LU workflow.

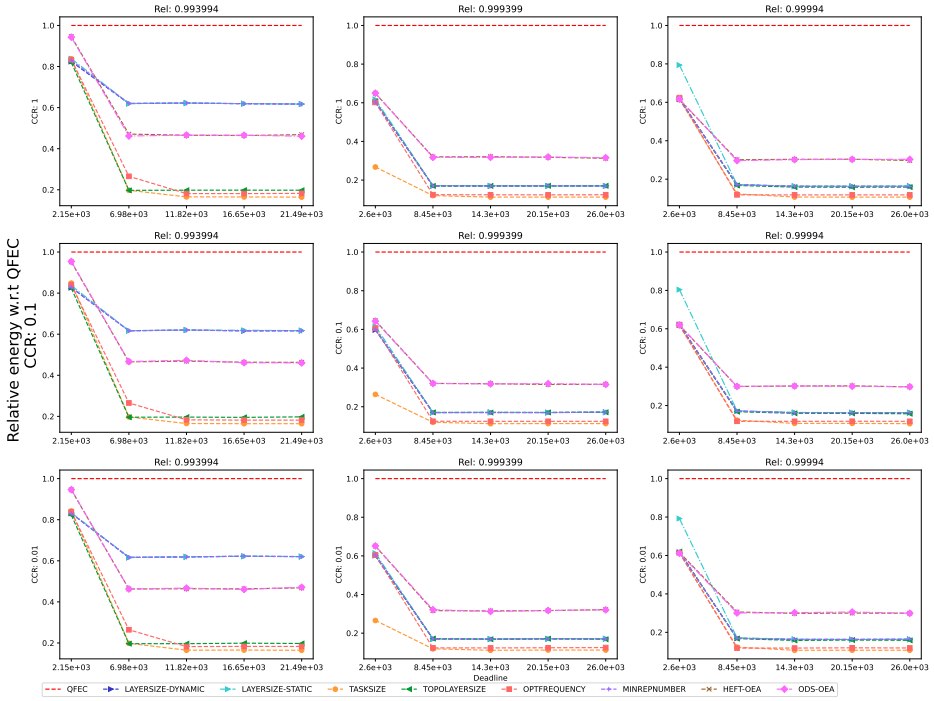

Figure 507: Performance of the different heuristics on the Montage workflow.

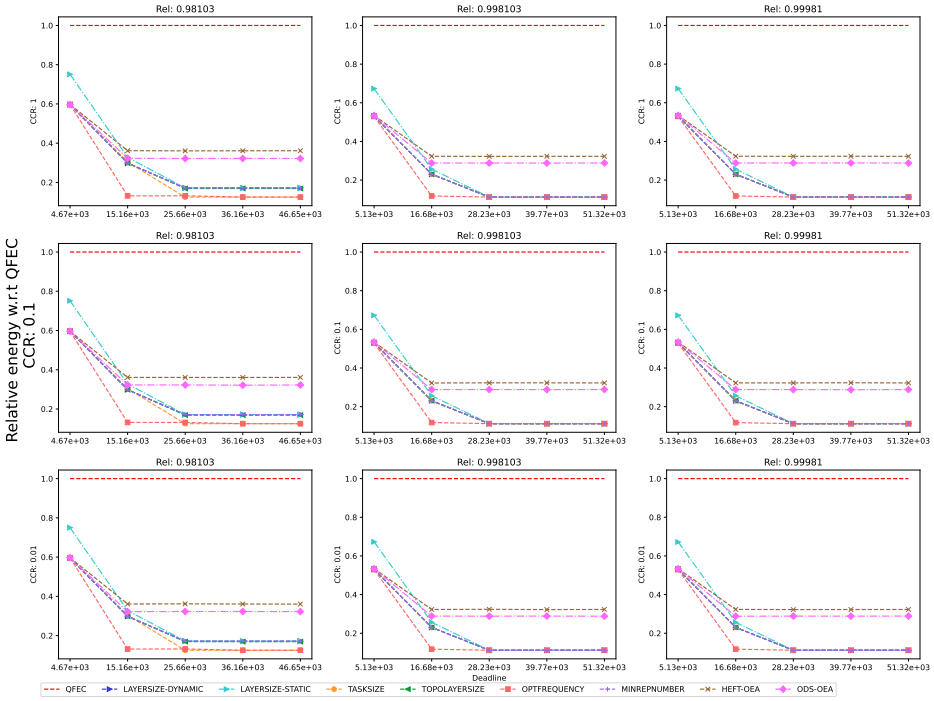

Figure 508: Performance of the different heuristics on the QR workflow.

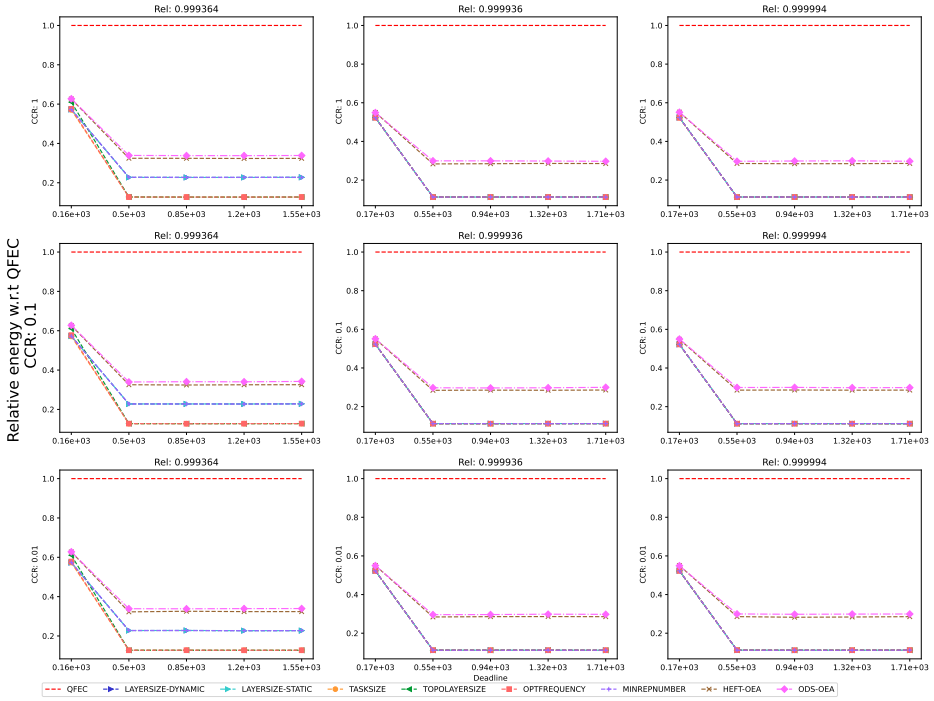

Figure 509: Performance of the different heuristics on the Seismology workflow.

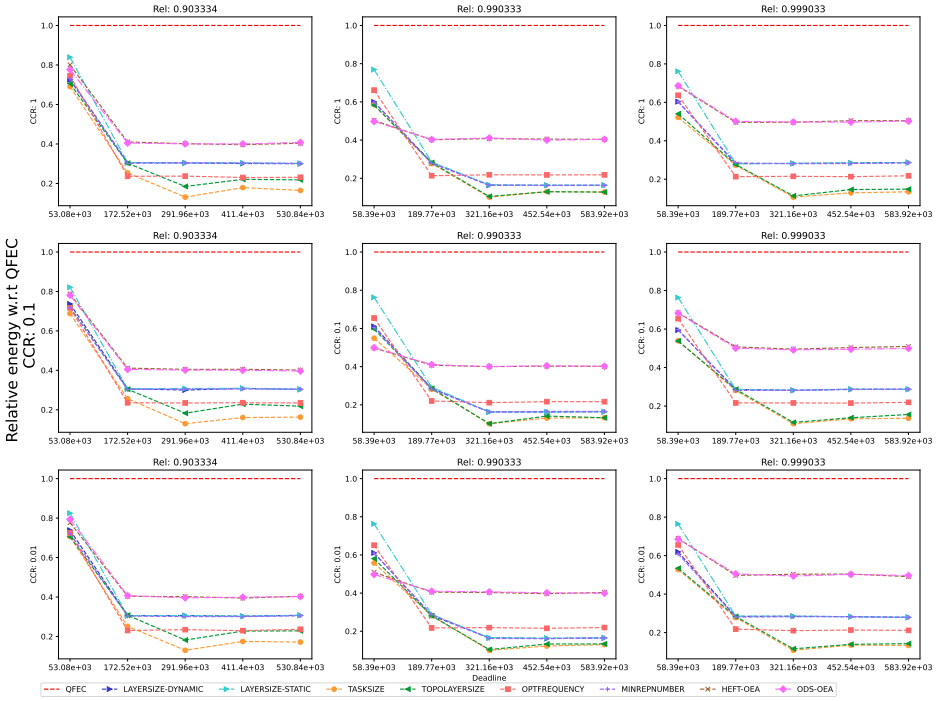

Figure 510: Performance of the different heuristics on the SoyKB workflow.

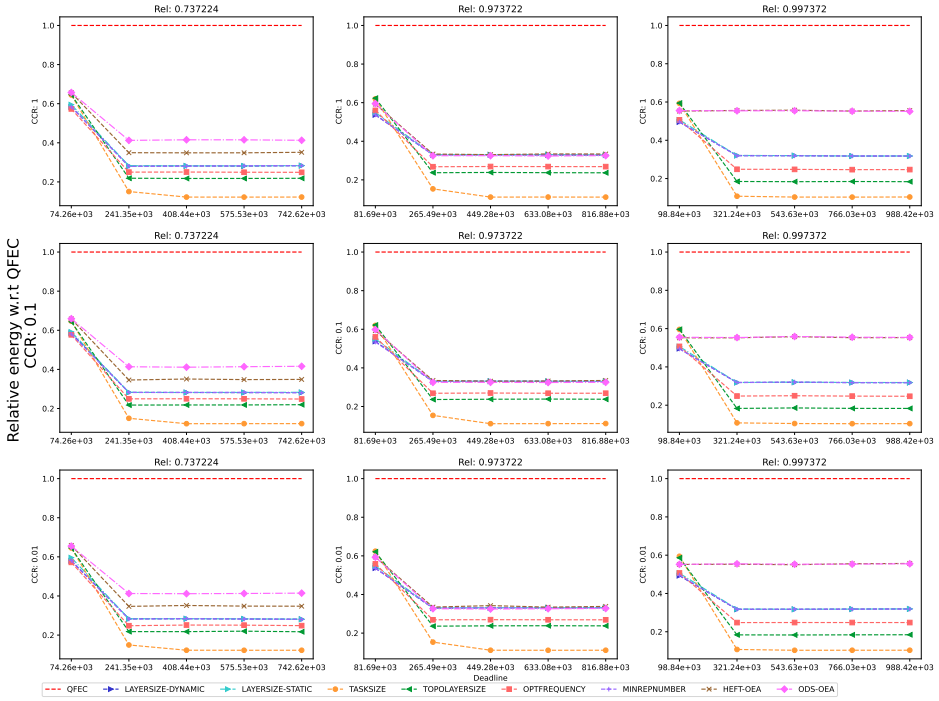

Figure 511: Performance of the different heuristics on the SRASearch.

E.2.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

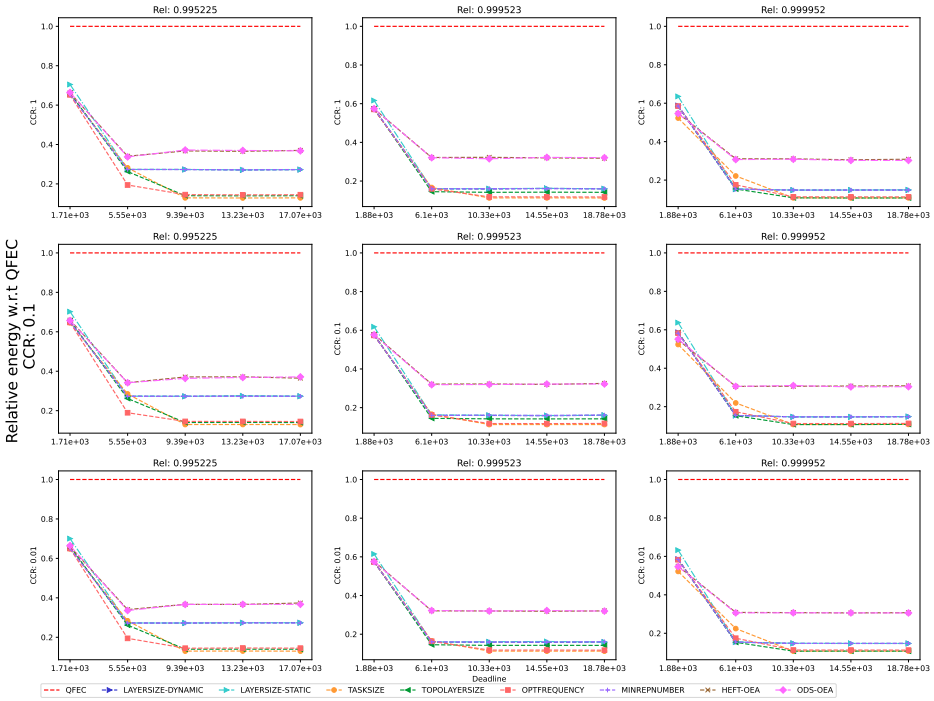

Figure 512: Performance of the different heuristics on the BWA workflow.

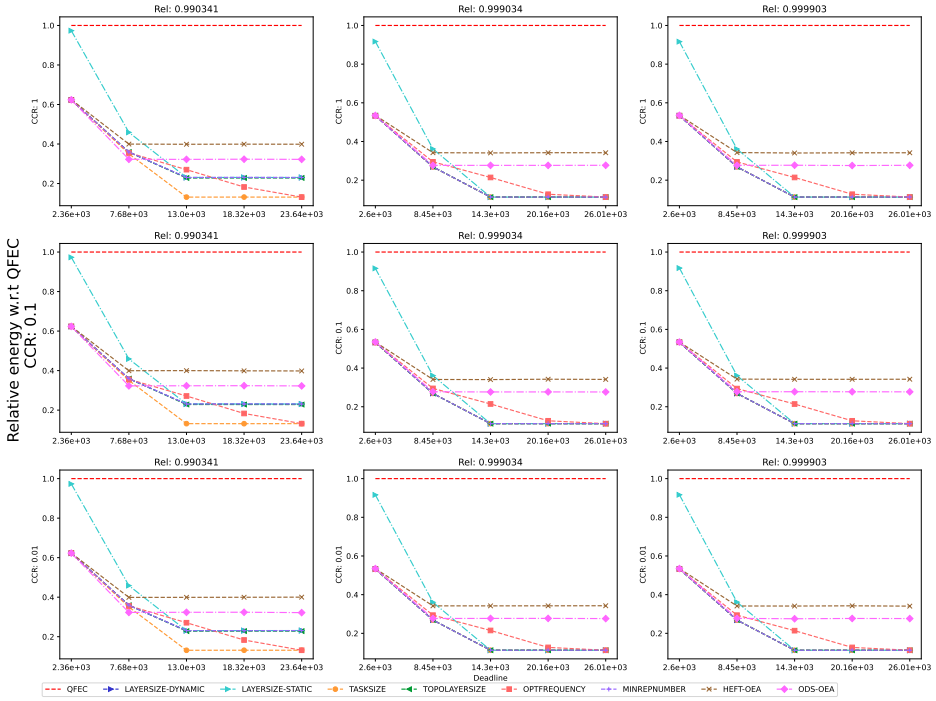

Figure 513: Performance of the different heuristics on the Cholesky workflow.

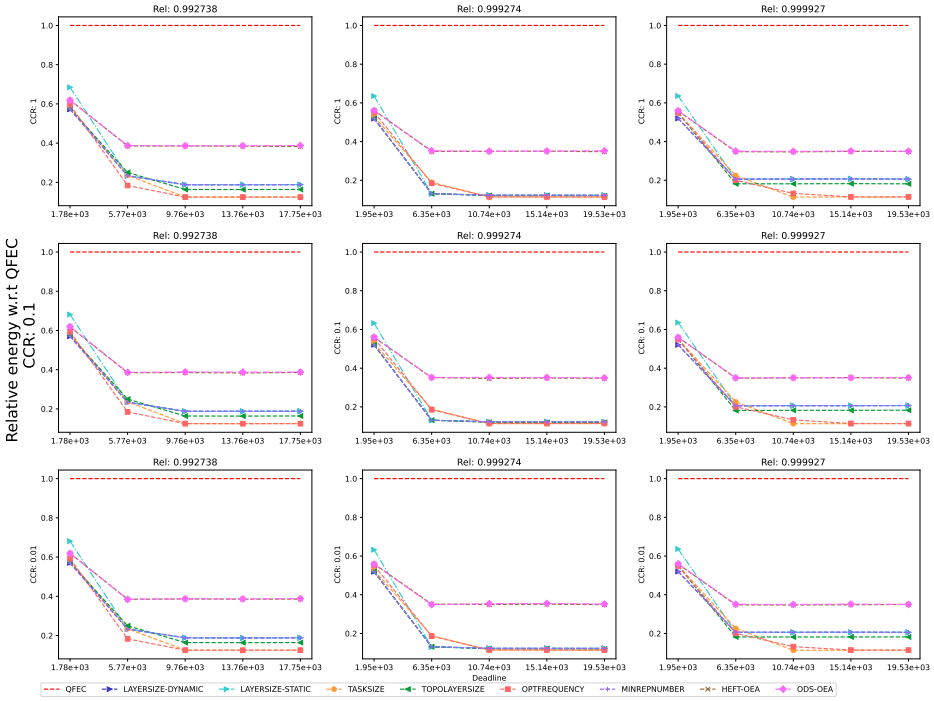

Figure 514: Performance of the different heuristics on the Cycles workflow.

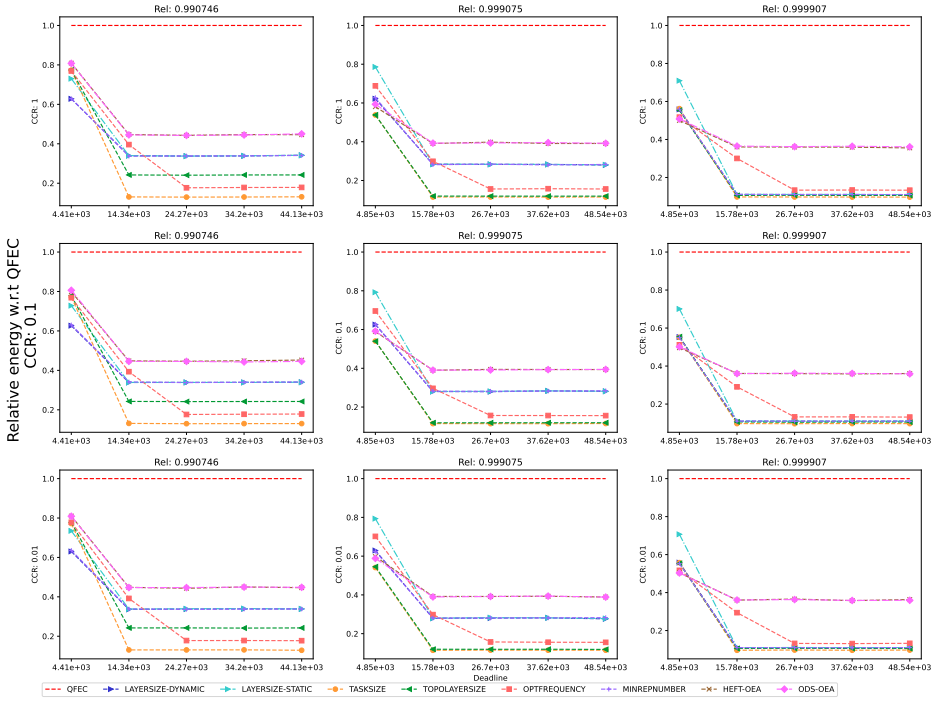

Figure 515: Performance of the different heuristics on the Epigenomics workflow.

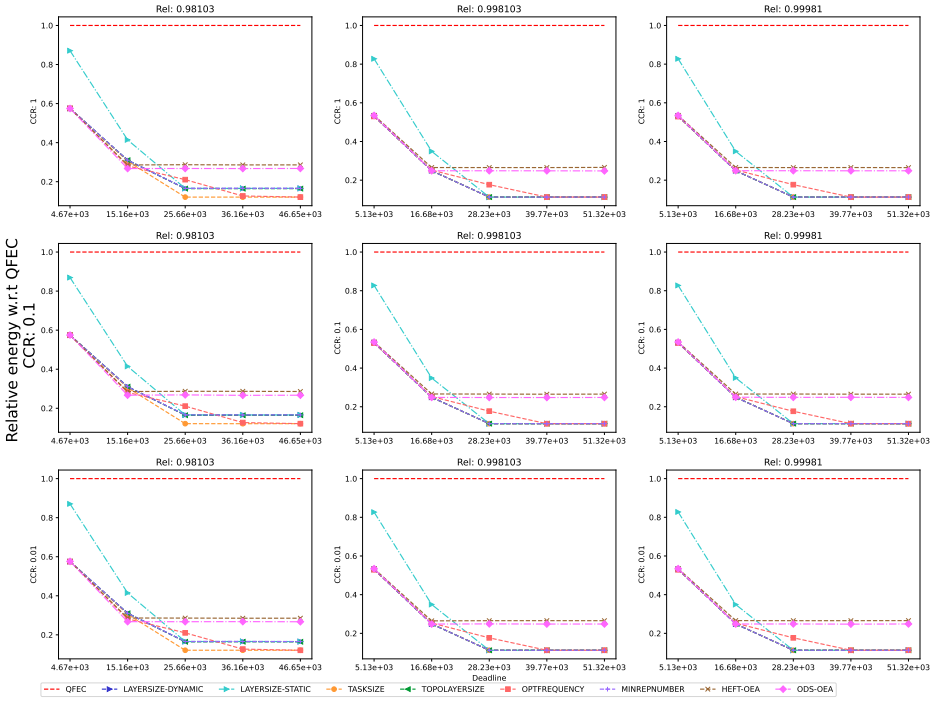

Figure 516: Performance of the different heuristics on the LU workflow.

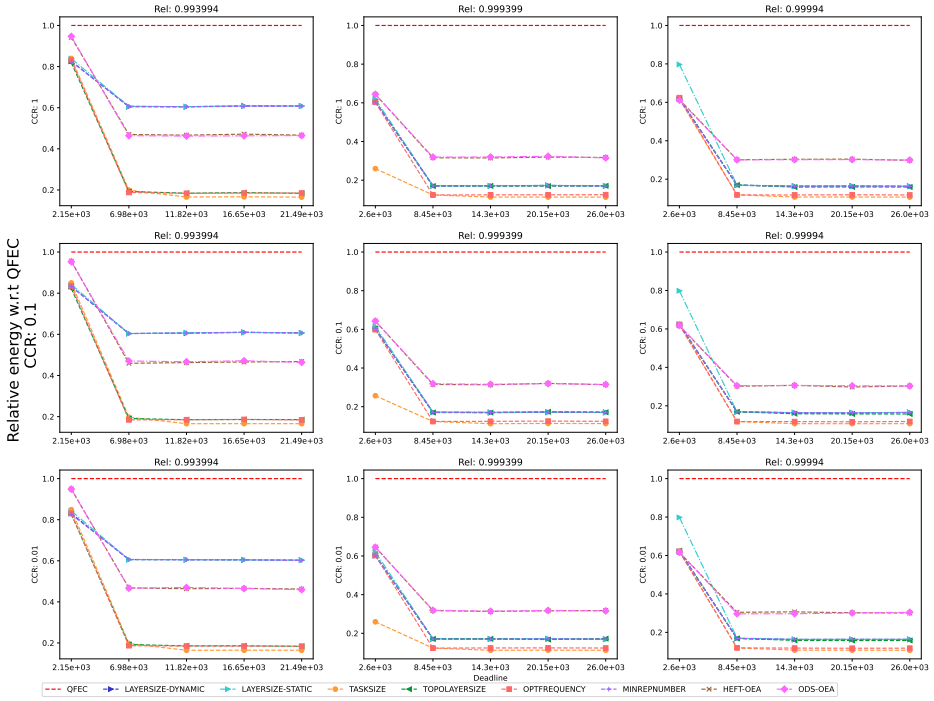

Figure 517: Performance of the different heuristics on the Montage workflow.

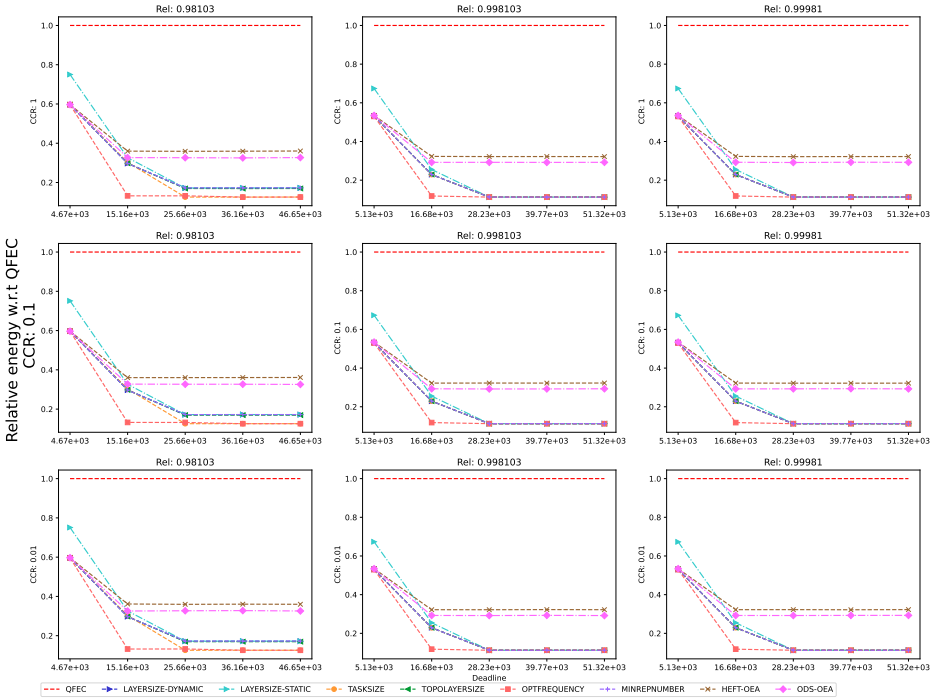

Figure 518: Performance of the different heuristics on the QR workflow.

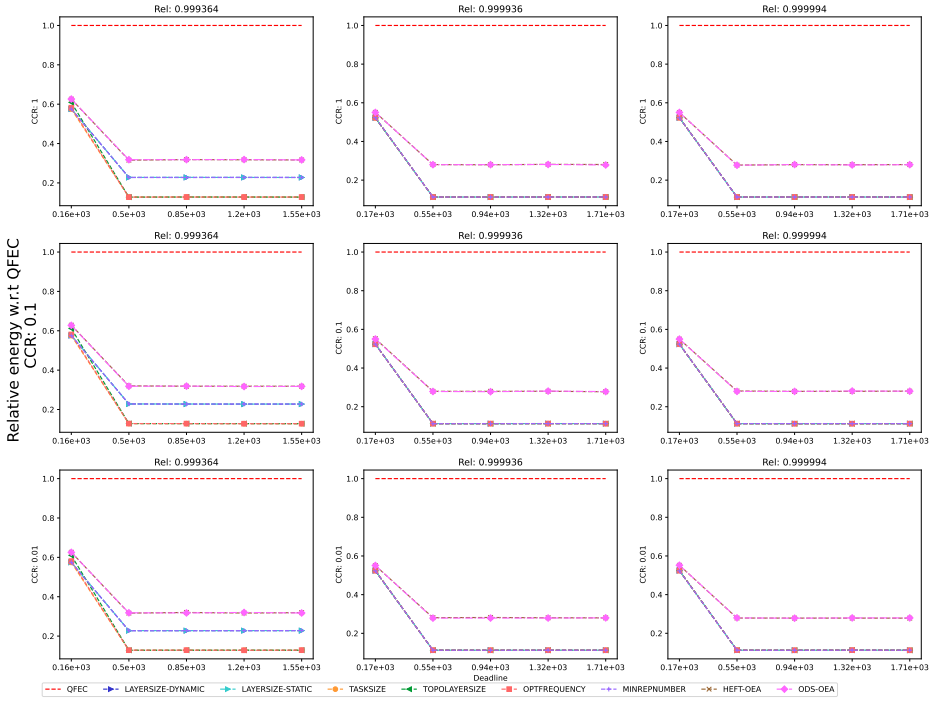

Figure 519: Performance of the different heuristics on the Seismology workflow.

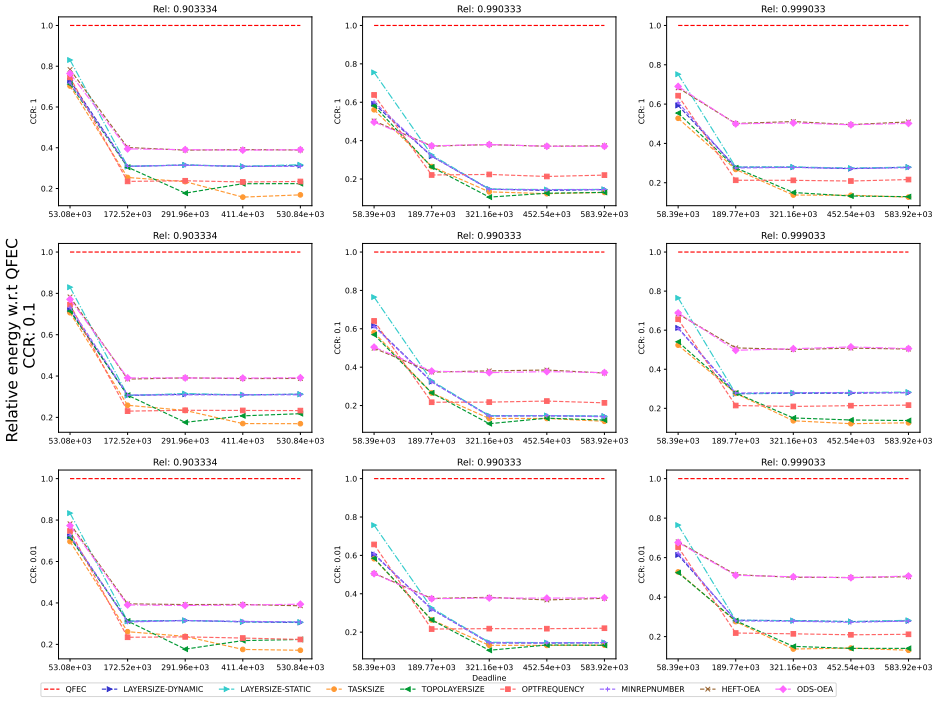

Figure 520: Performance of the different heuristics on the SoyKB workflow.

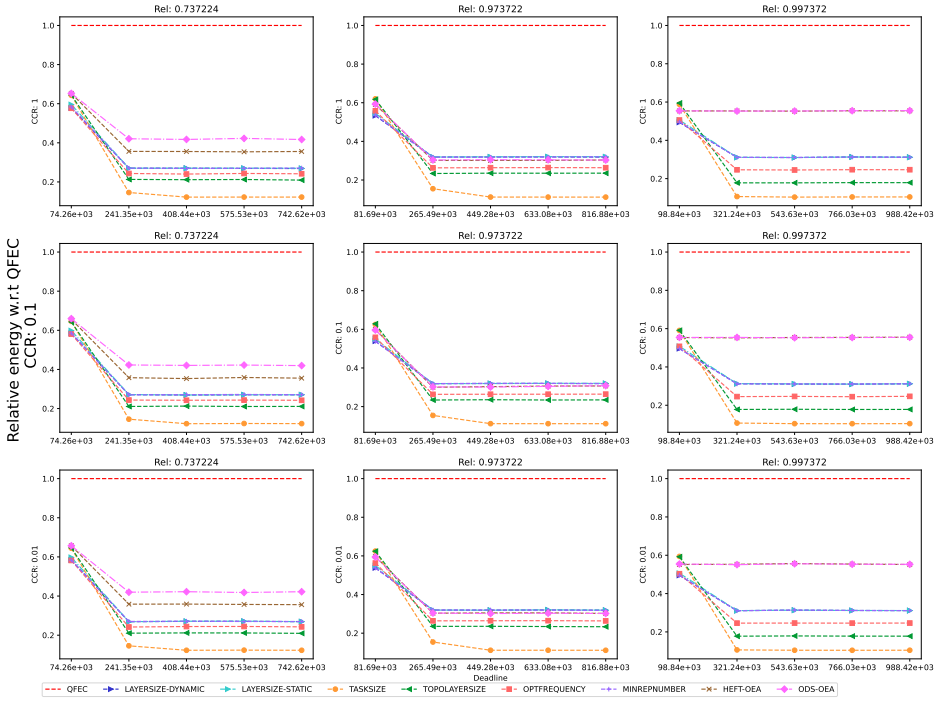

Figure 521: Performance of the different heuristics on the SRASearch.

E.3 $BC/WC = 0.3$

E.3.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

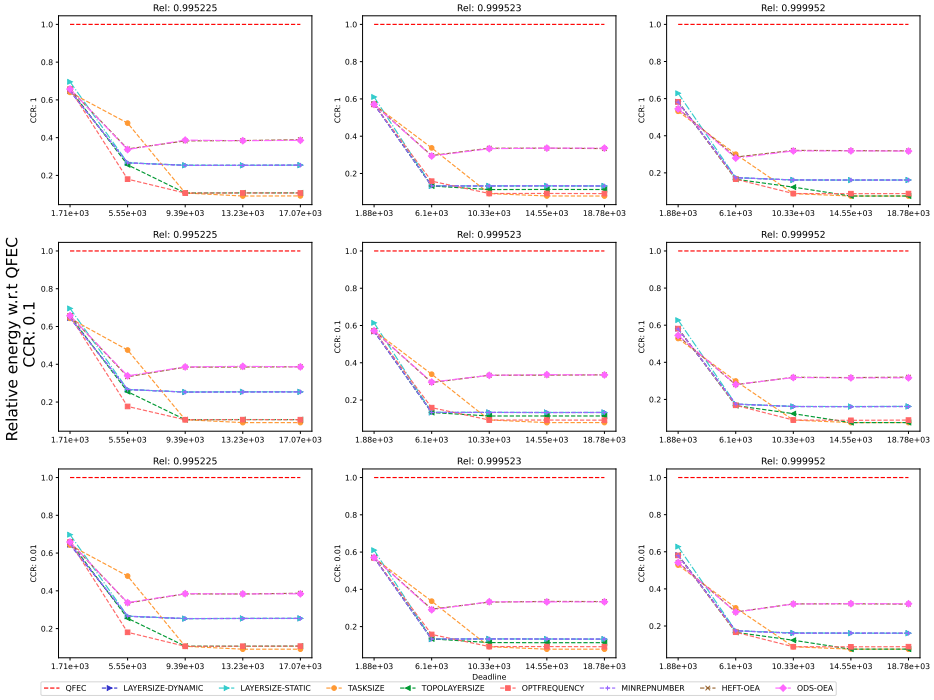

Figure 522: Performance of the different heuristics on the BWA workflow.

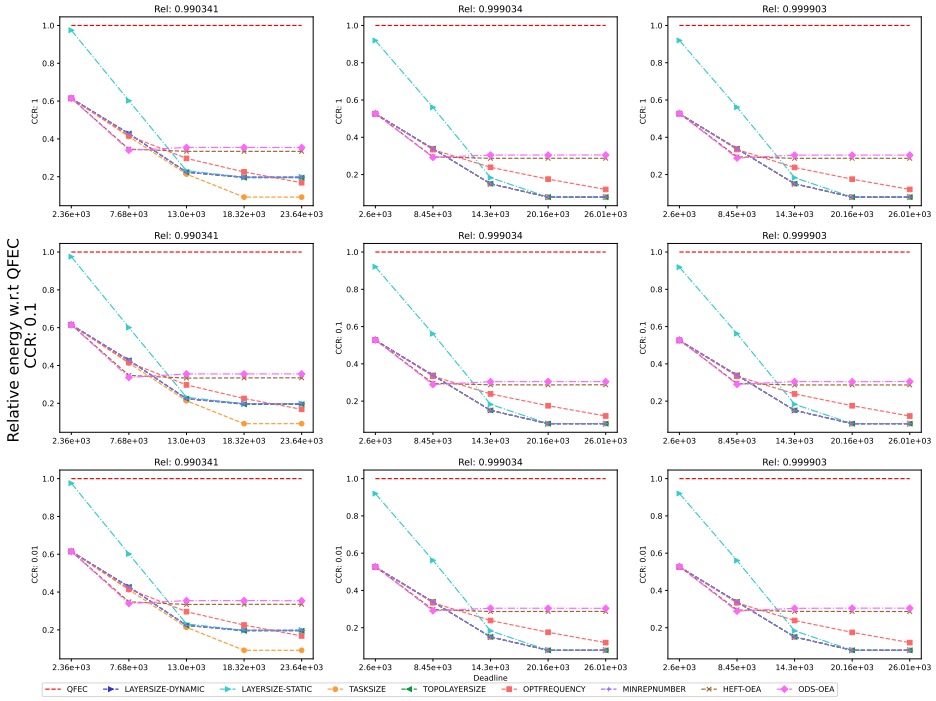

Figure 523: Performance of the different heuristics on the Cholesky workflow.

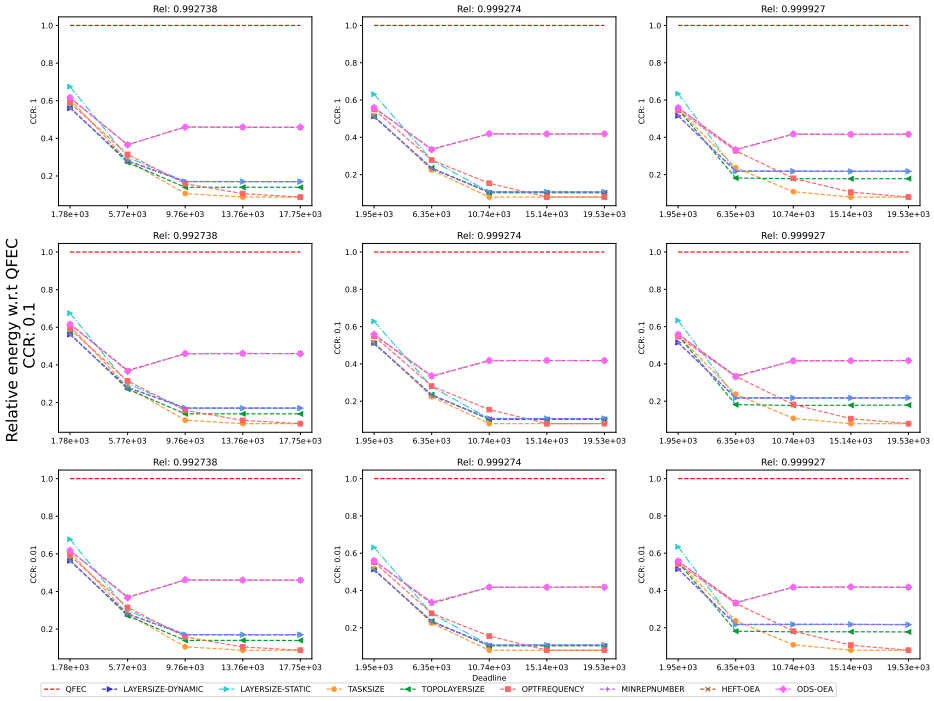

Figure 524: Performance of the different heuristics on the Cycles workflow.

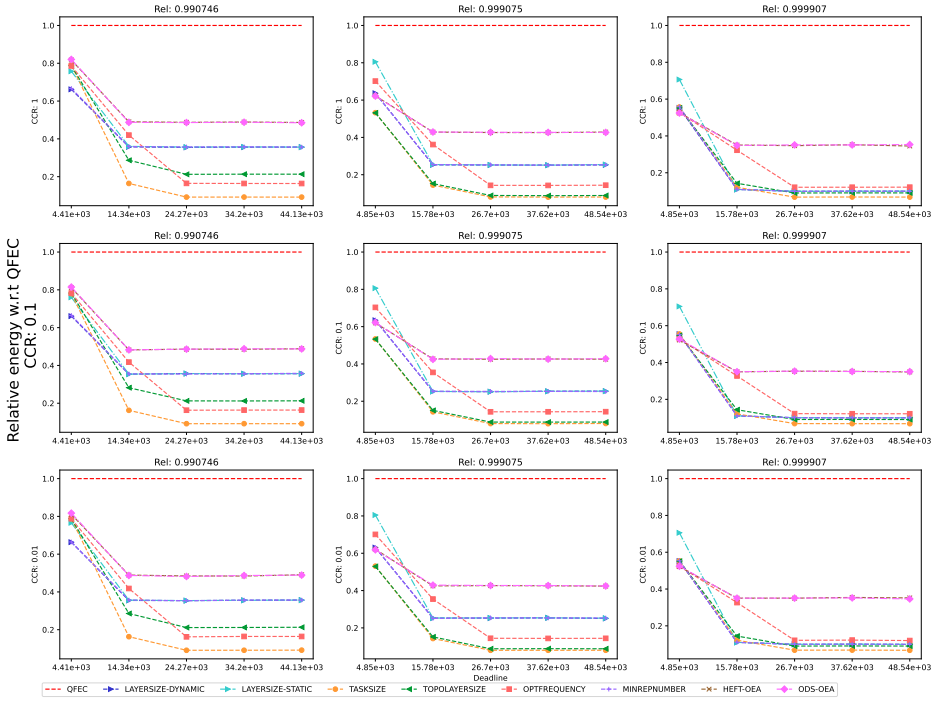

Figure 525: Performance of the different heuristics on the Epigenomics workflow.

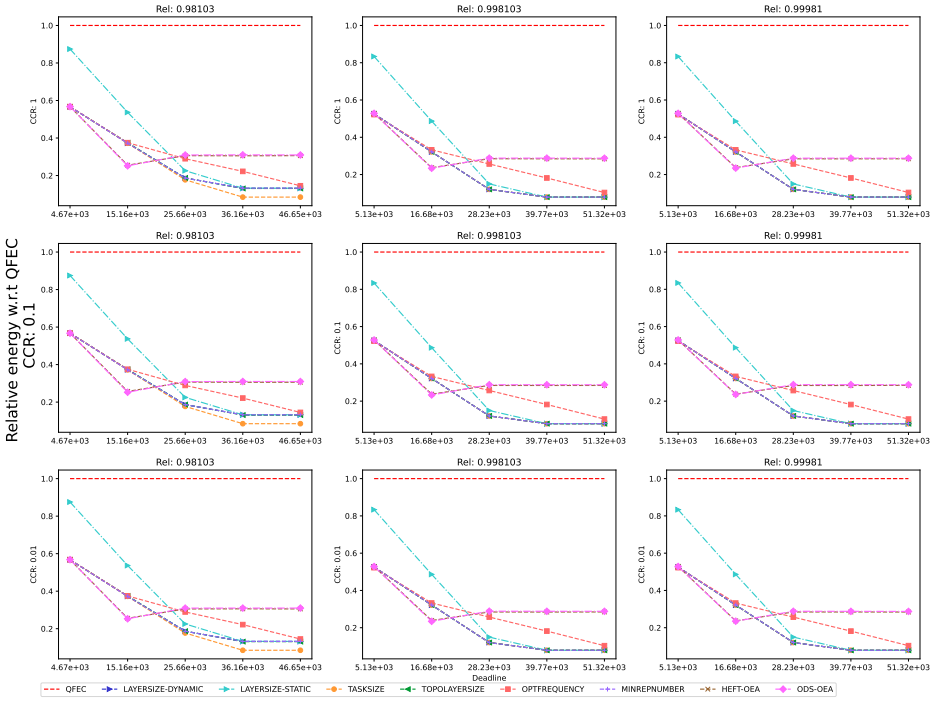

Figure 526: Performance of the different heuristics on the LU workflow.

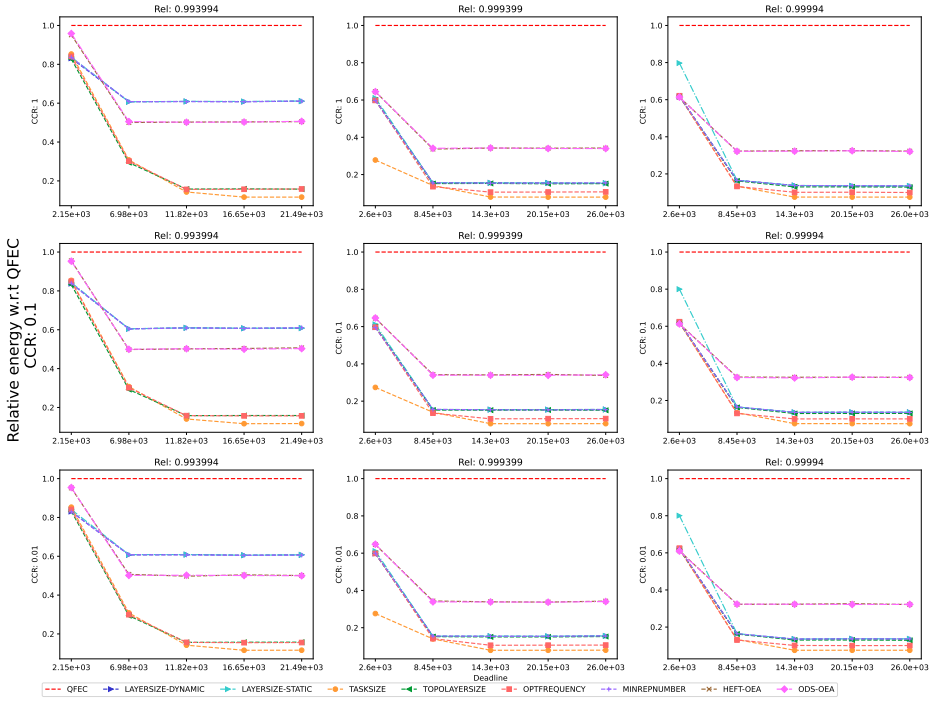

Figure 527: Performance of the different heuristics on the Montage workflow.

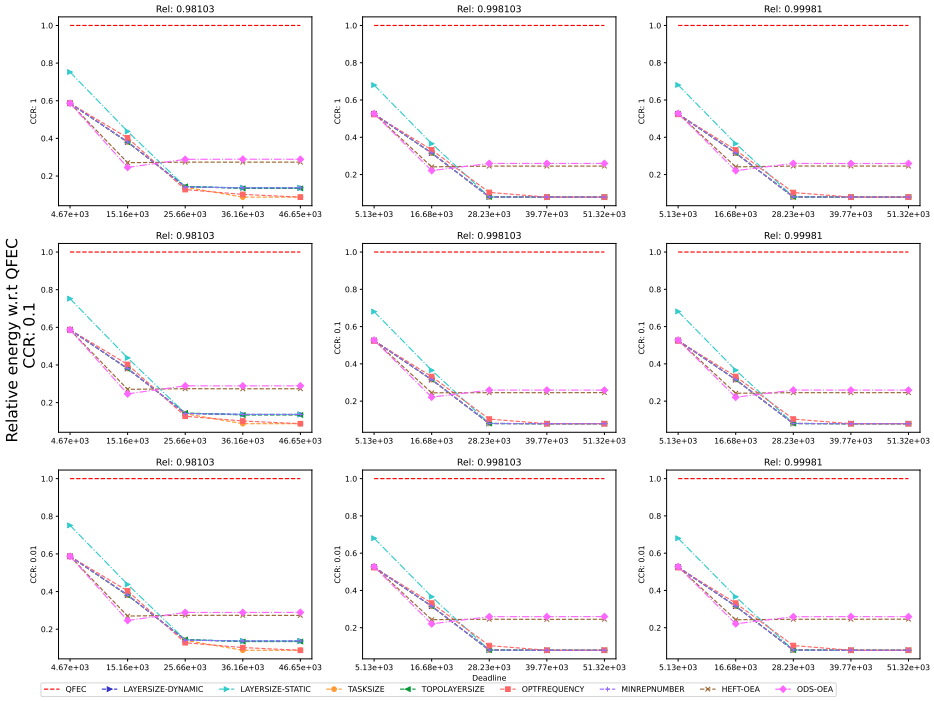

Figure 528: Performance of the different heuristics on the QR workflow.

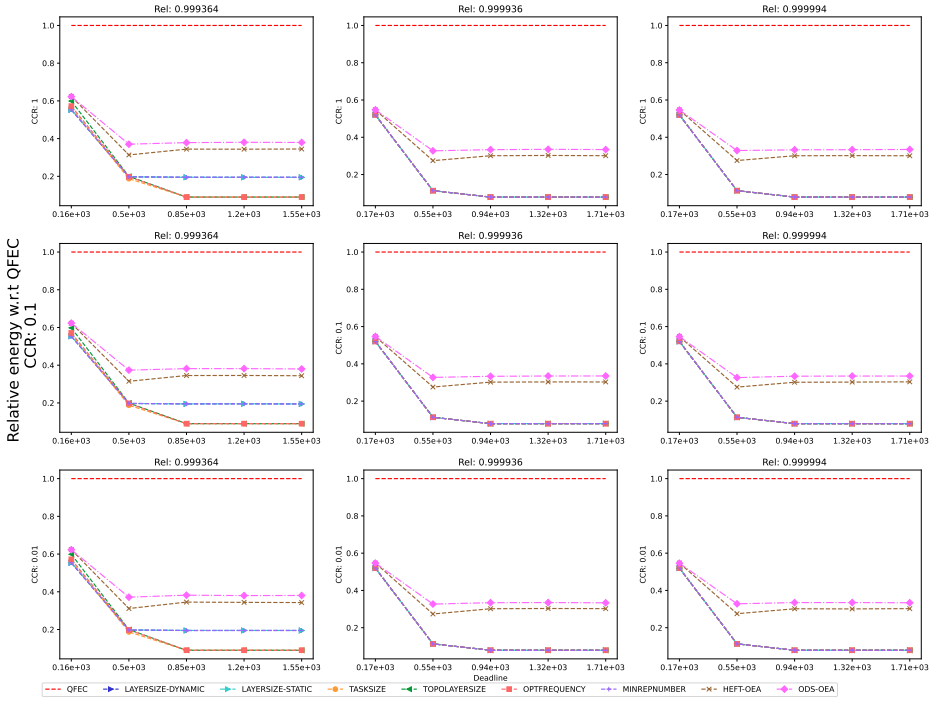

Figure 529: Performance of the different heuristics on the Seismology workflow.

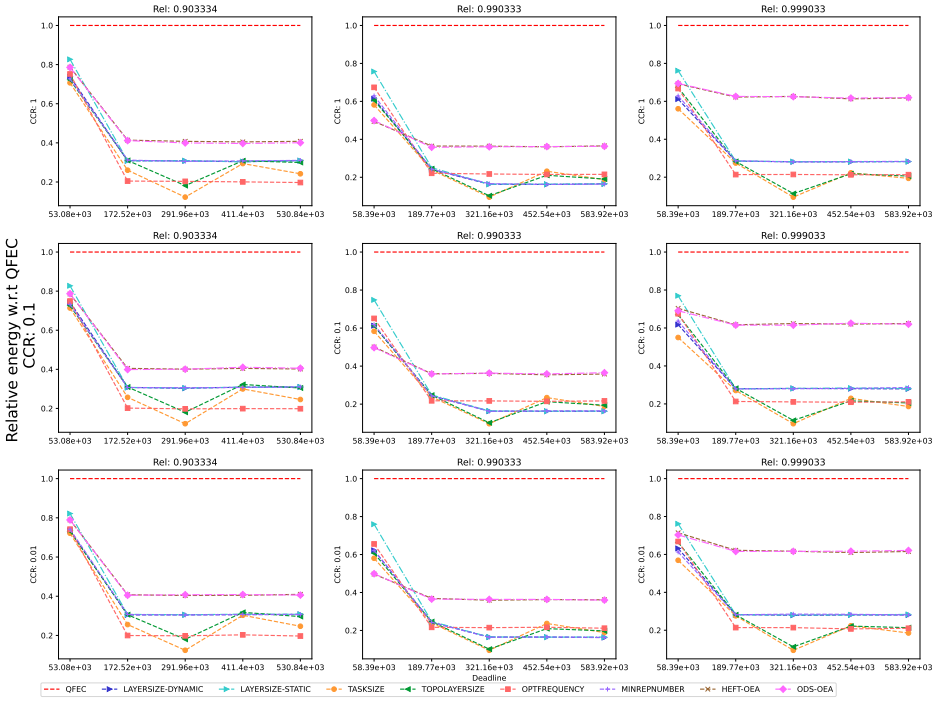

Figure 530: Performance of the different heuristics on the SoyKB workflow.

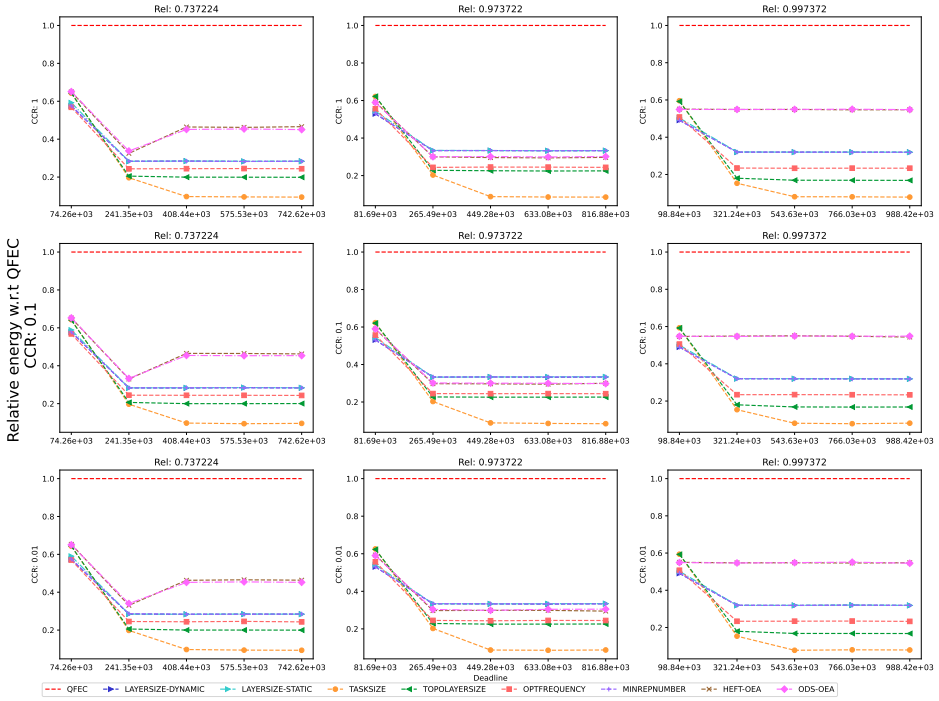

Figure 531: Performance of the different heuristics on the SRASearch.

E.3.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

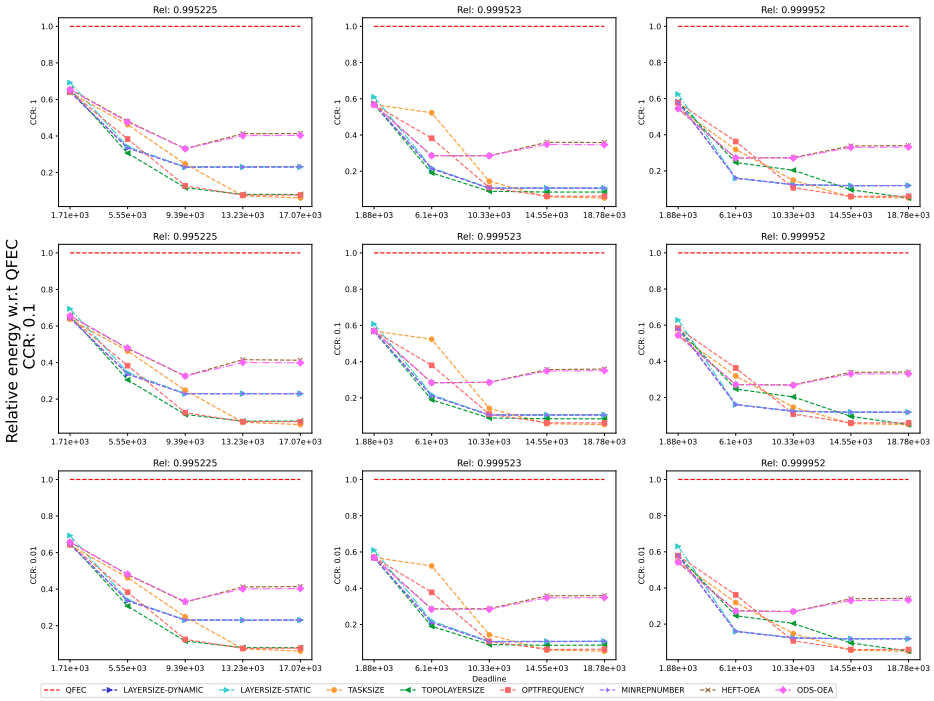

Figure 532: Performance of the different heuristics on the BWA workflow.

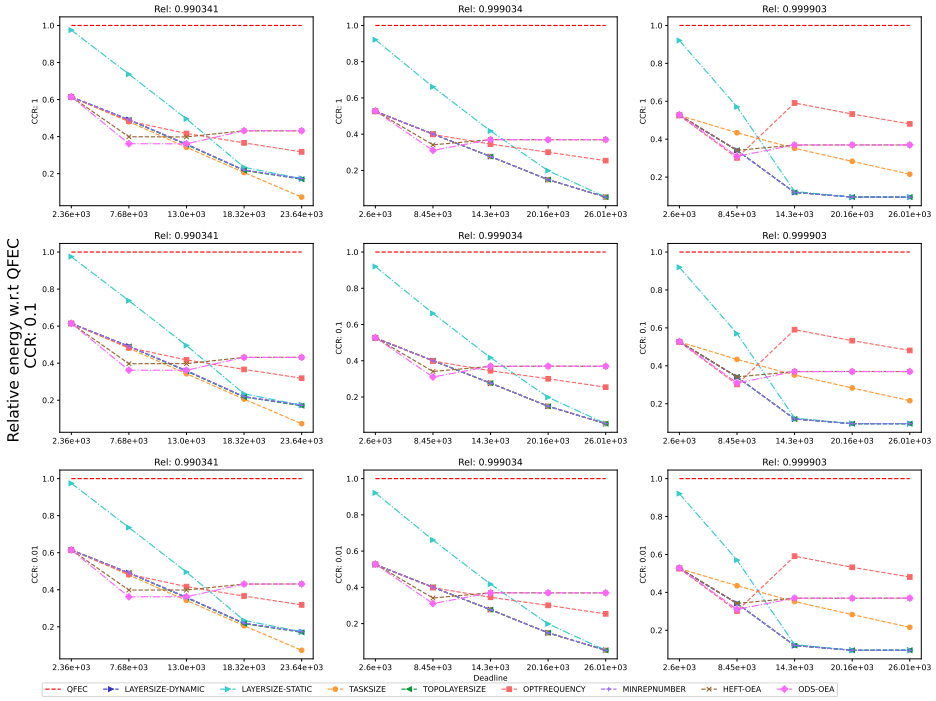

Figure 533: Performance of the different heuristics on the Cholesky workflow.

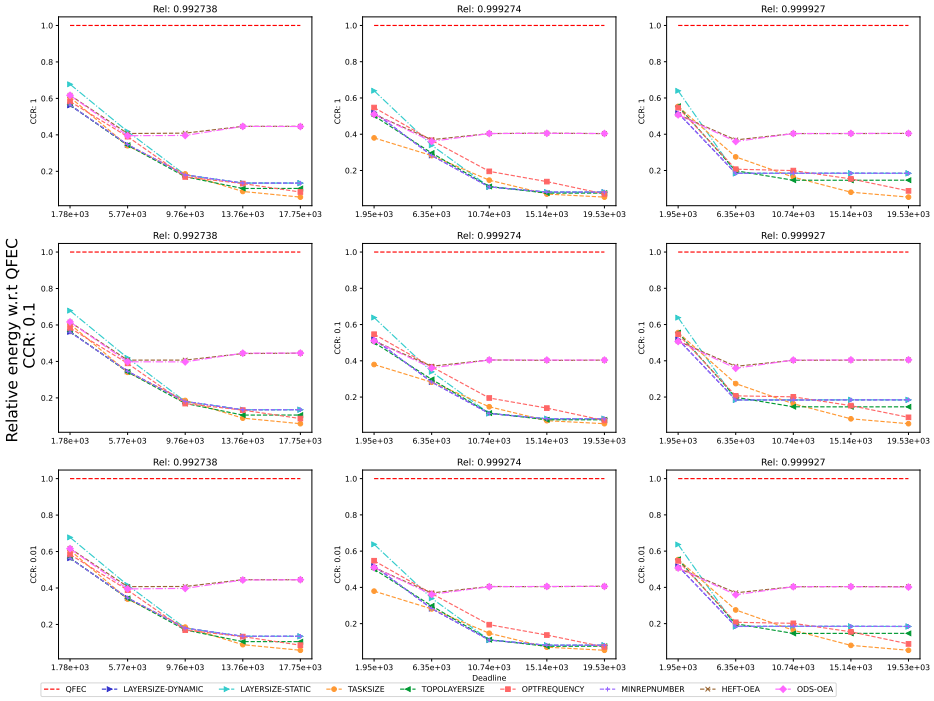

Figure 534: Performance of the different heuristics on the Cycles workflow.

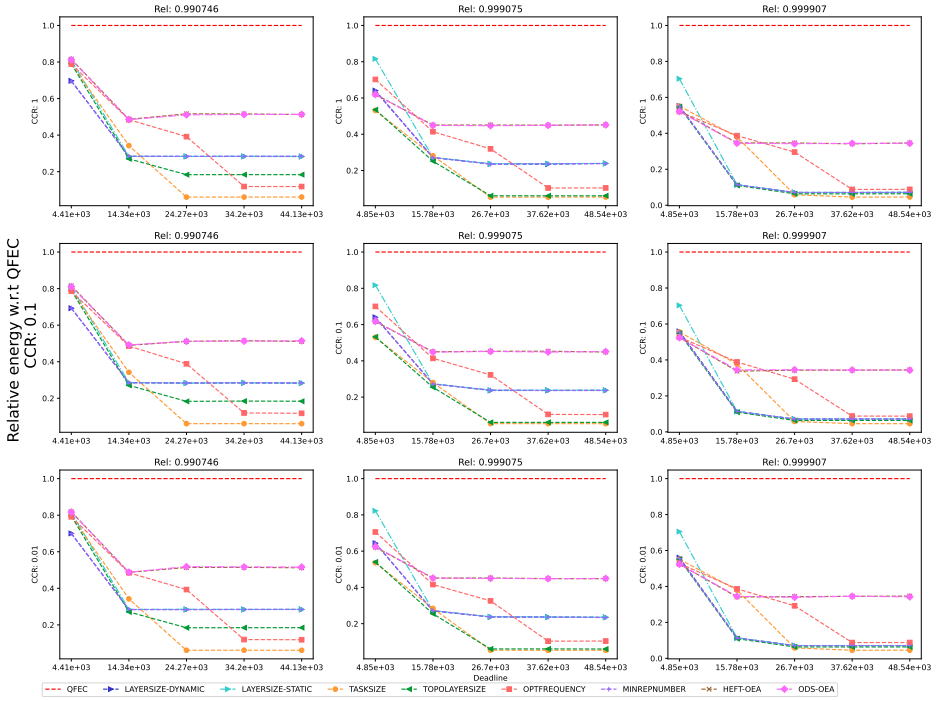

Figure 535: Performance of the different heuristics on the Epigenomics workflow.

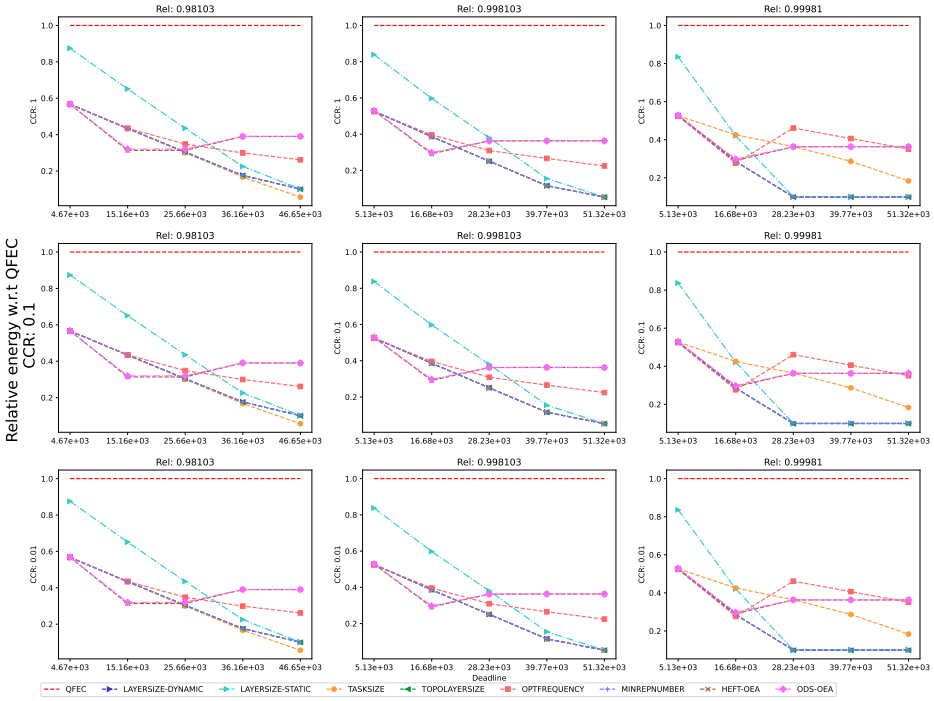

Figure 536: Performance of the different heuristics on the LU workflow.

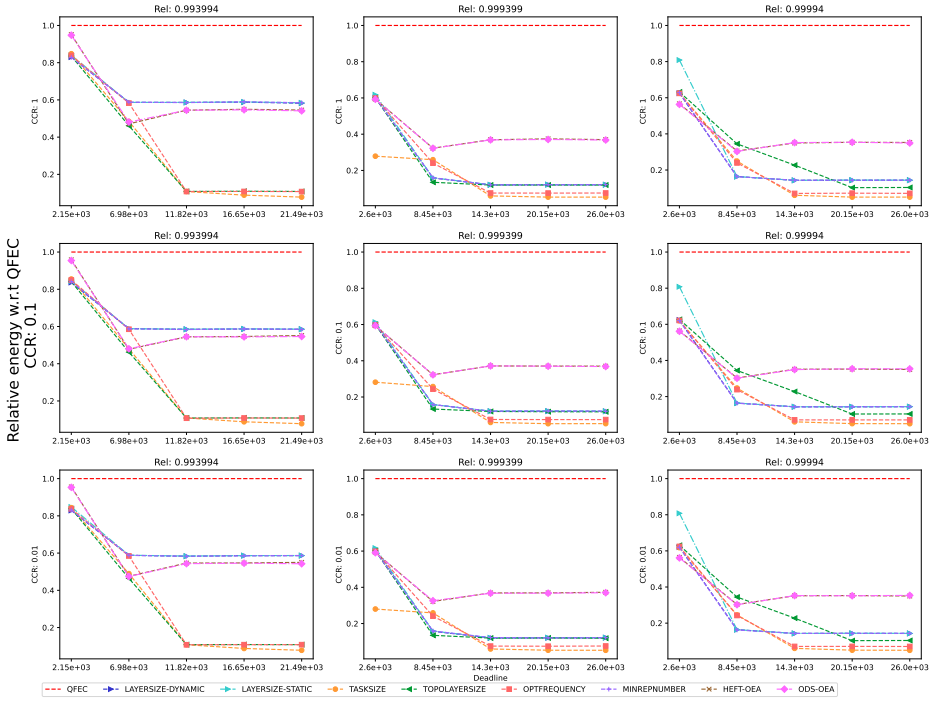

Figure 537: Performance of the different heuristics on the Montage workflow.

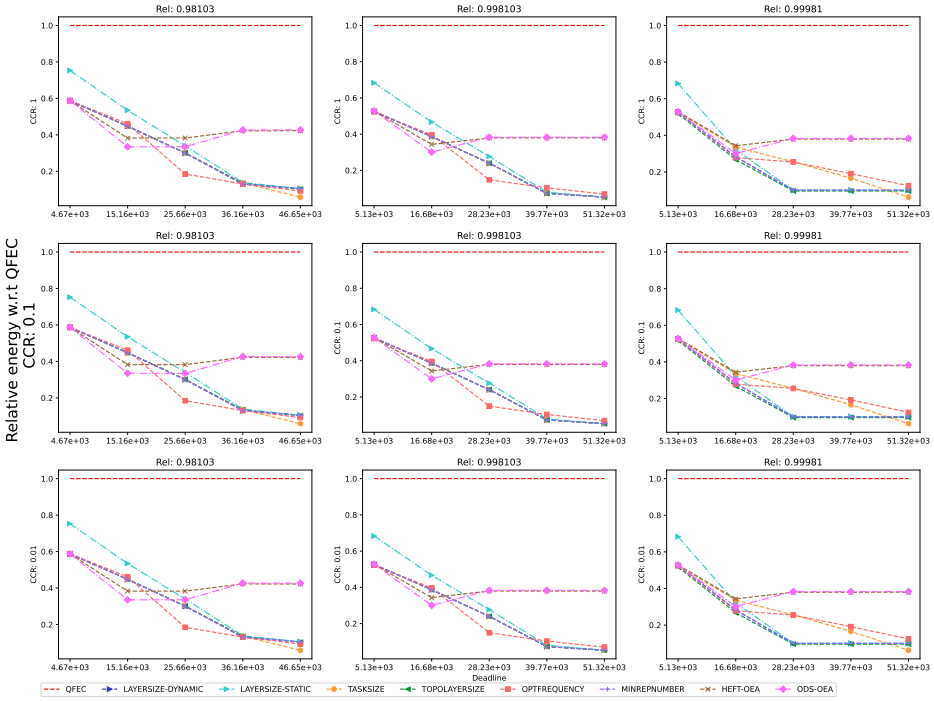

Figure 538: Performance of the different heuristics on the QR workflow.

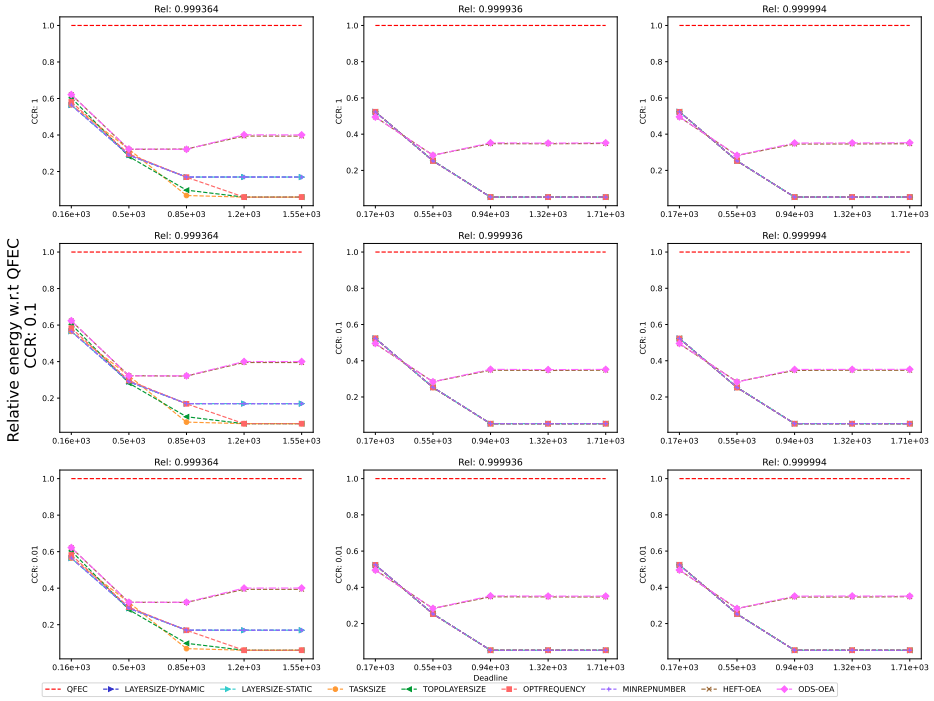

Figure 539: Performance of the different heuristics on the Seismology workflow.

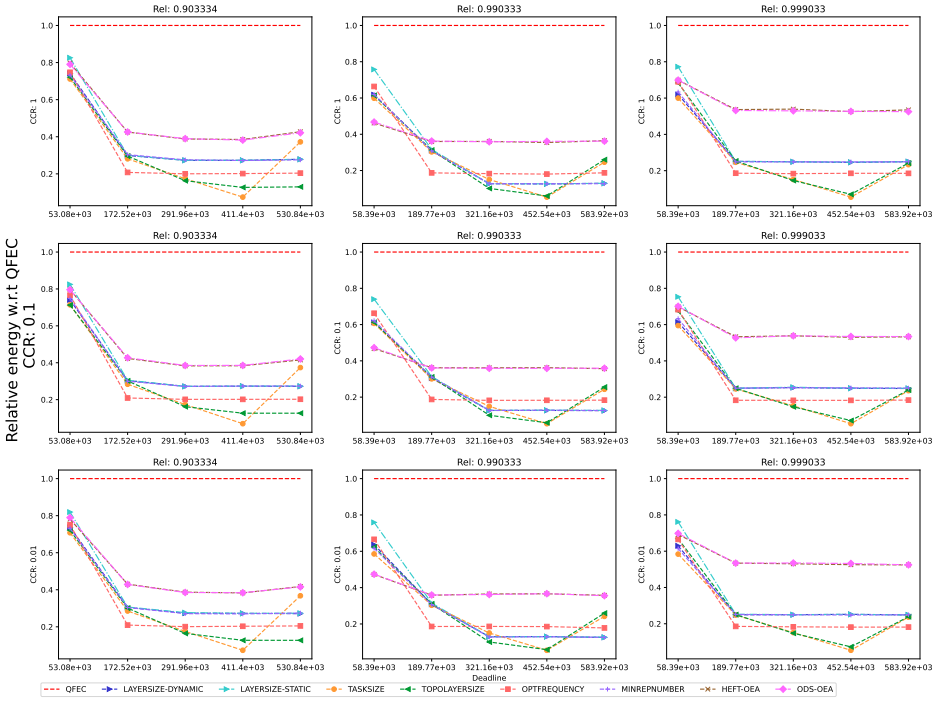

Figure 540: Performance of the different heuristics on the SoyKB workflow.

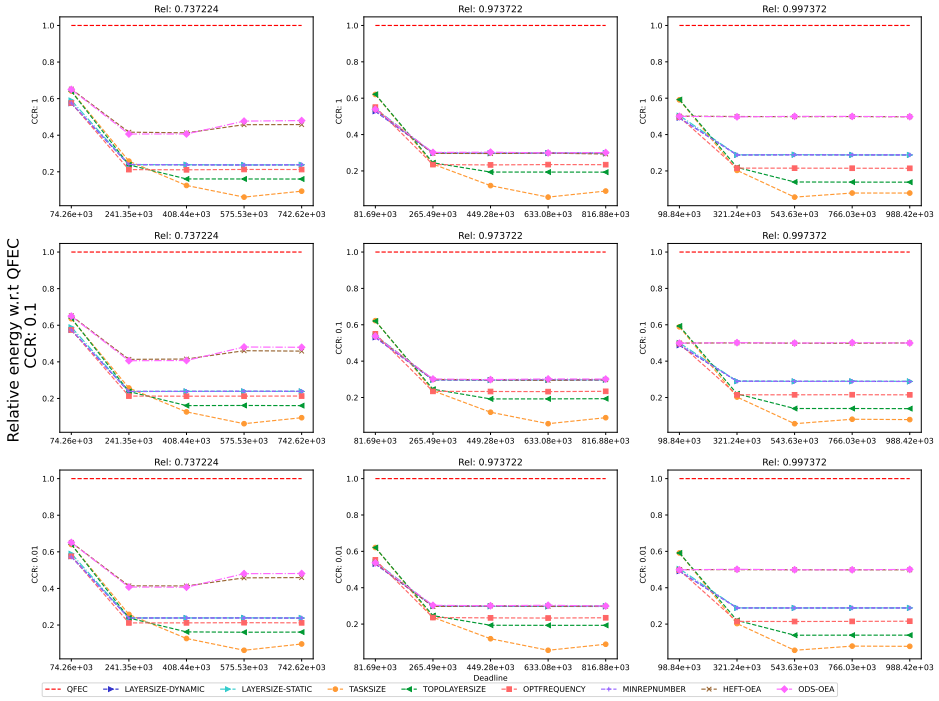

Figure 541: Performance of the different heuristics on the SRASearch.

E.3.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

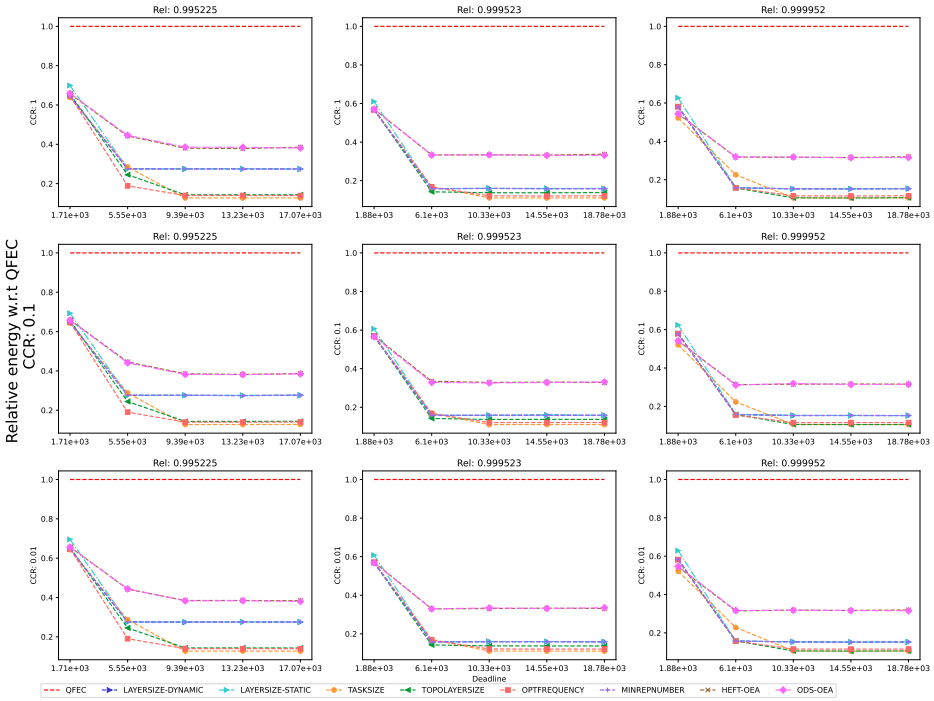

Figure 542: Performance of the different heuristics on the BWA workflow.

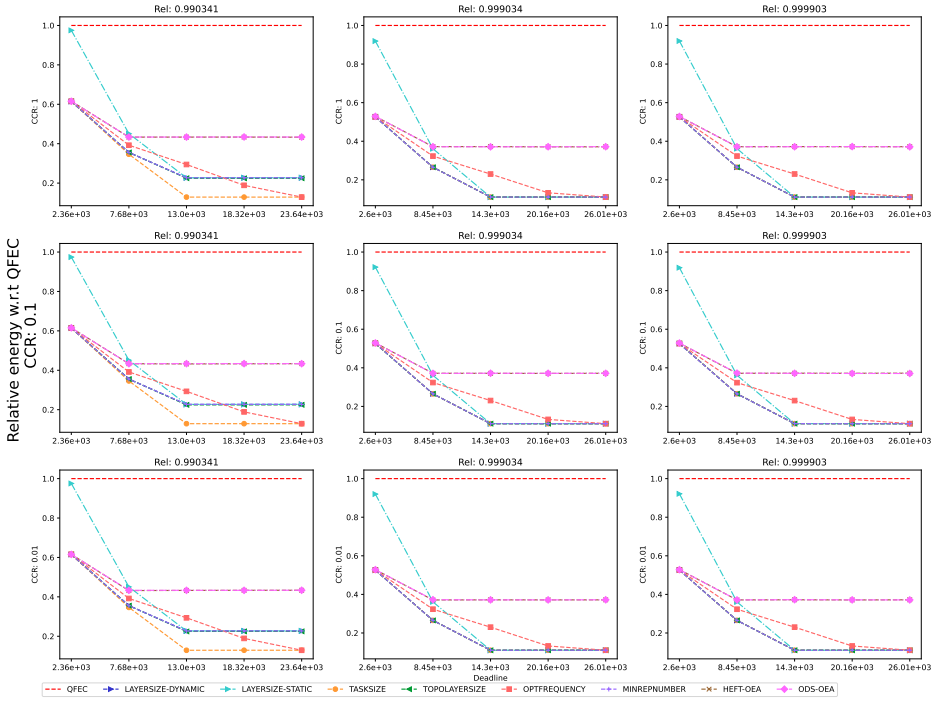

Figure 543: Performance of the different heuristics on the Cholesky workflow.

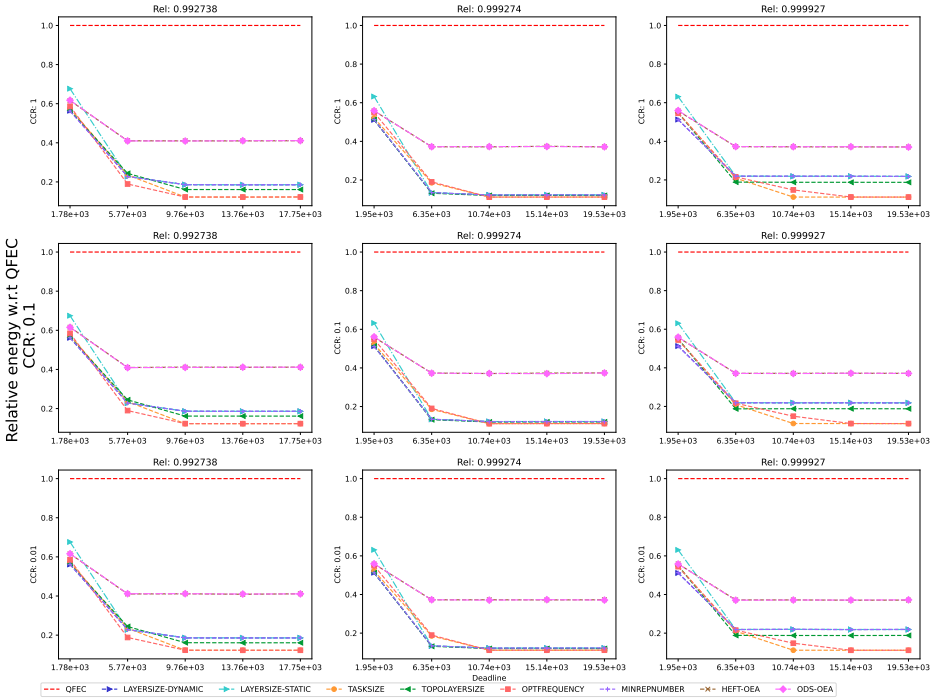

Figure 544: Performance of the different heuristics on the Cycles workflow.

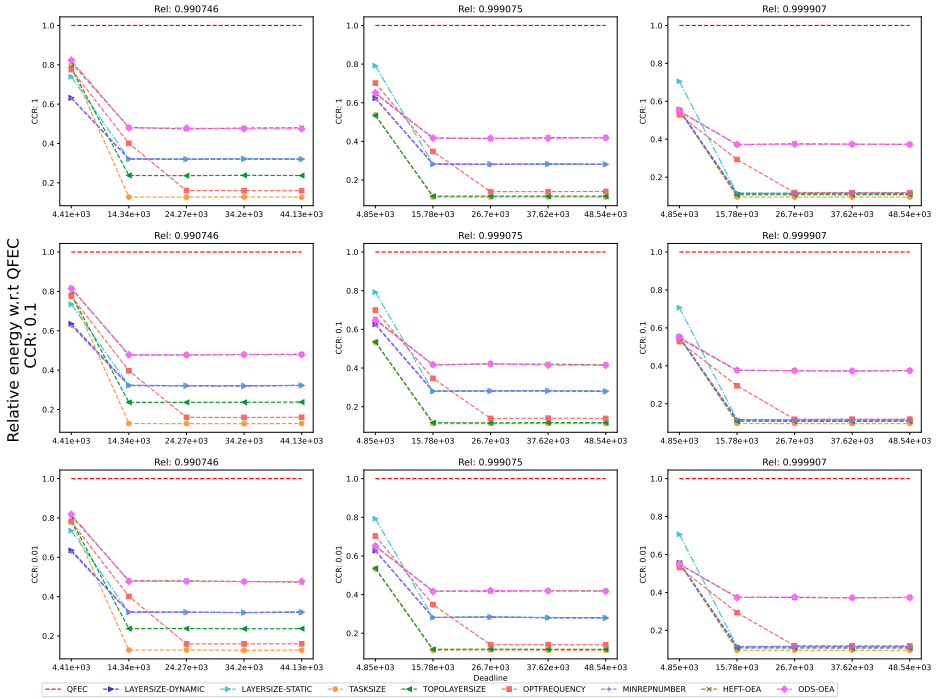

Figure 545: Performance of the different heuristics on the Epigenomics workflow.

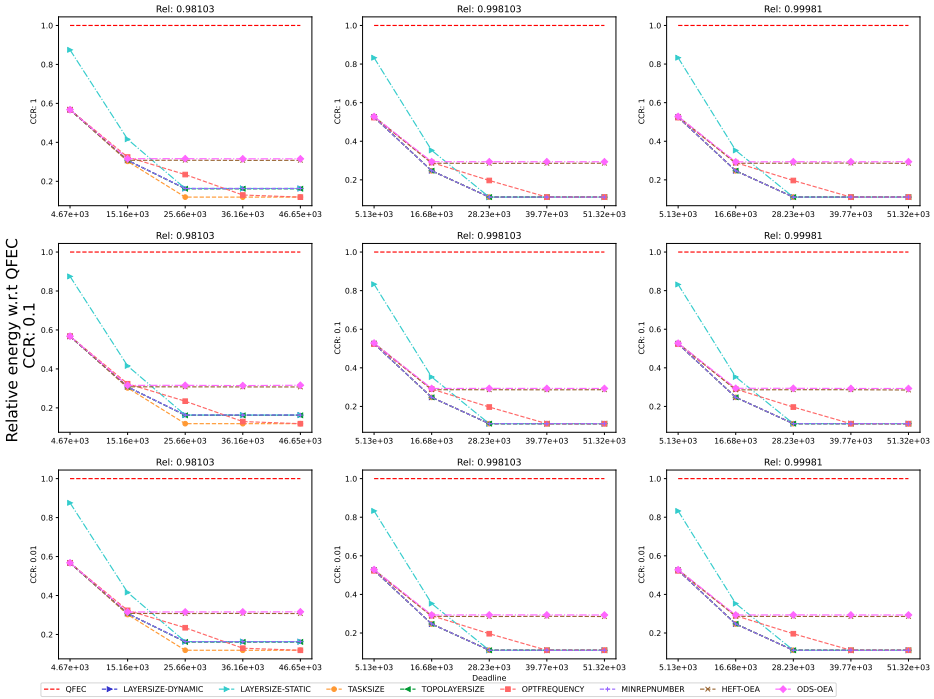

Figure 546: Performance of the different heuristics on the LU workflow.

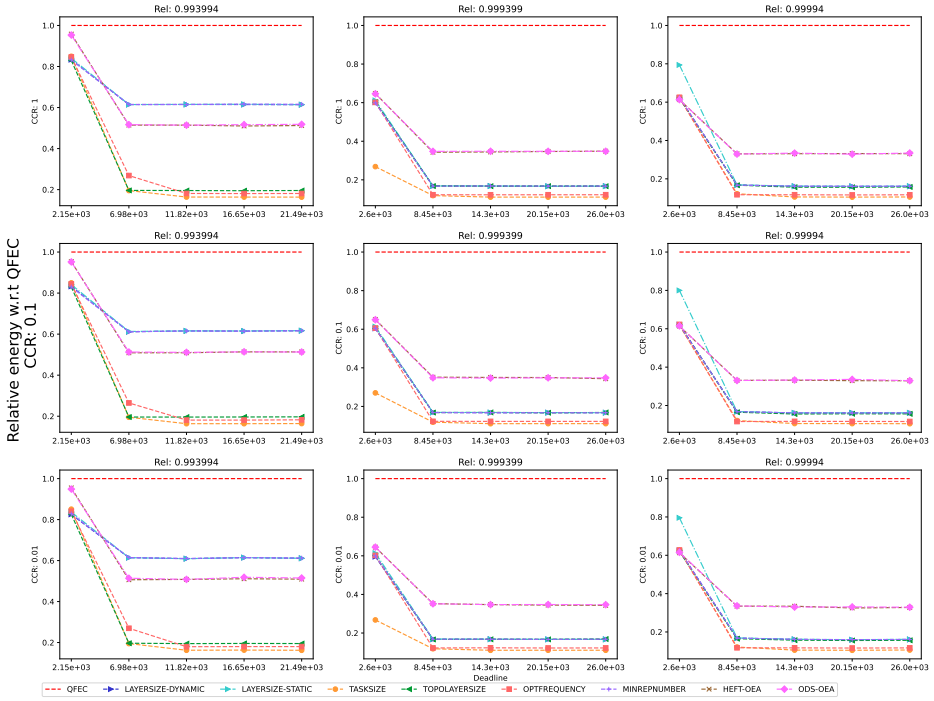

Figure 547: Performance of the different heuristics on the Montage workflow.

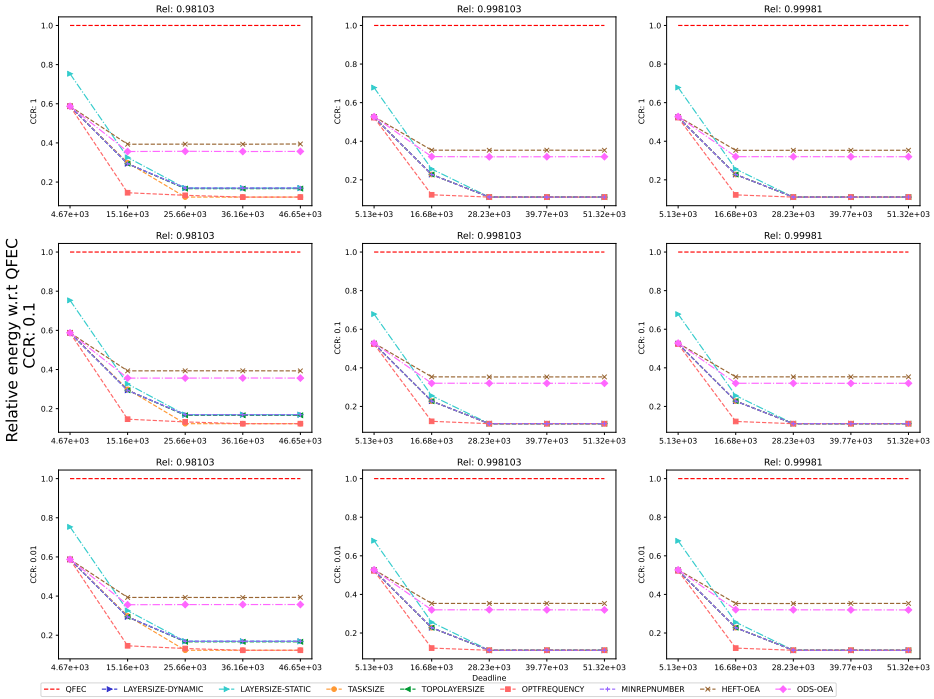

Figure 548: Performance of the different heuristics on the QR workflow.

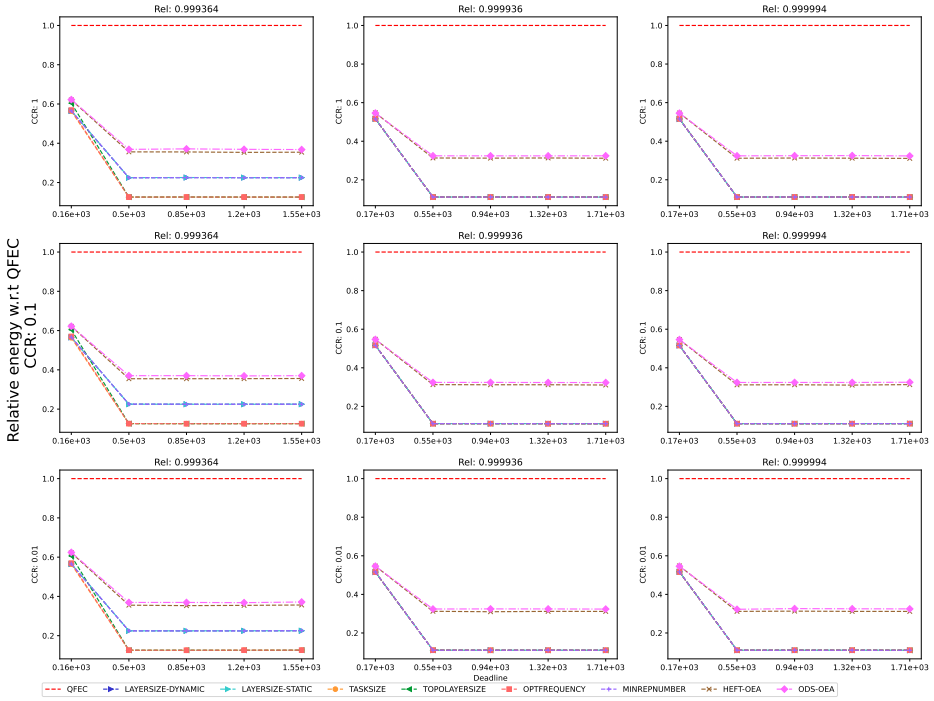

Figure 549: Performance of the different heuristics on the Seismology workflow.

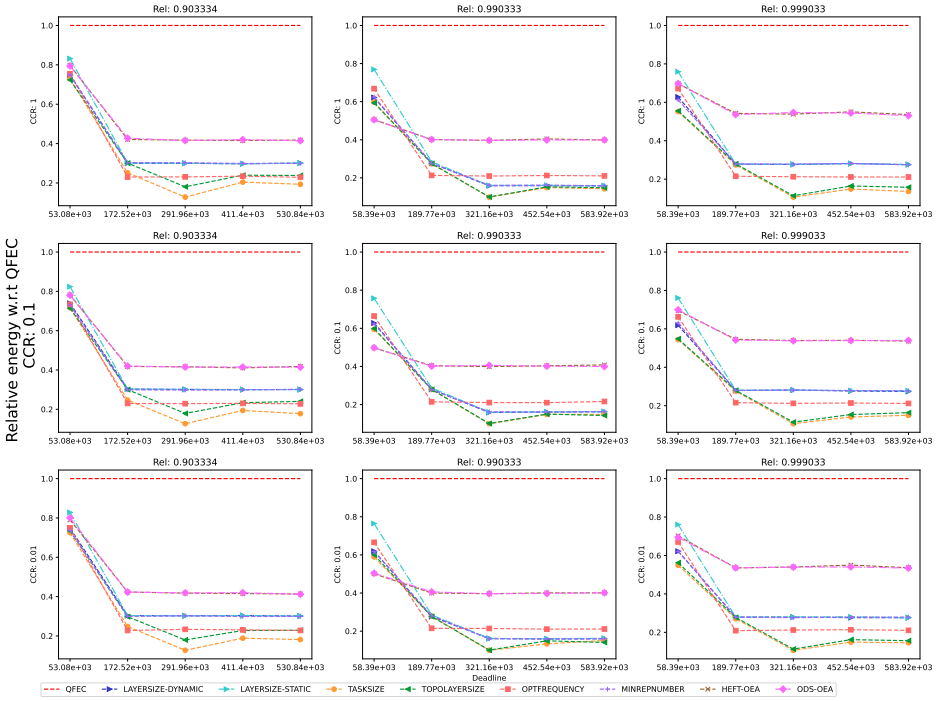

Figure 550: Performance of the different heuristics on the SoyKB workflow.

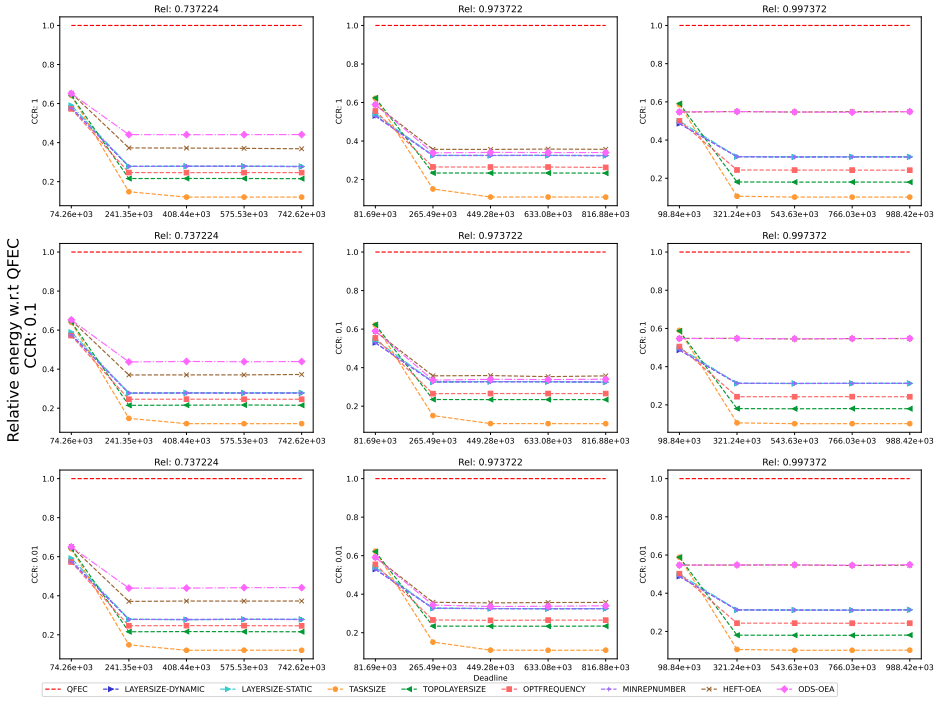

Figure 551: Performance of the different heuristics on the SRASearch.

E.3.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

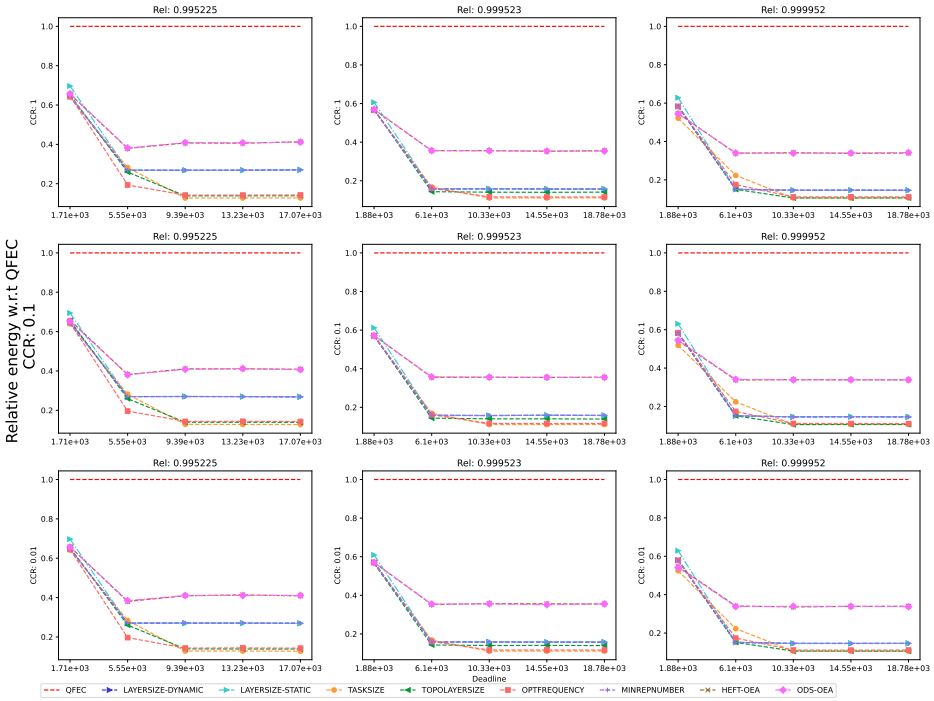

Figure 552: Performance of the different heuristics on the BWA workflow.

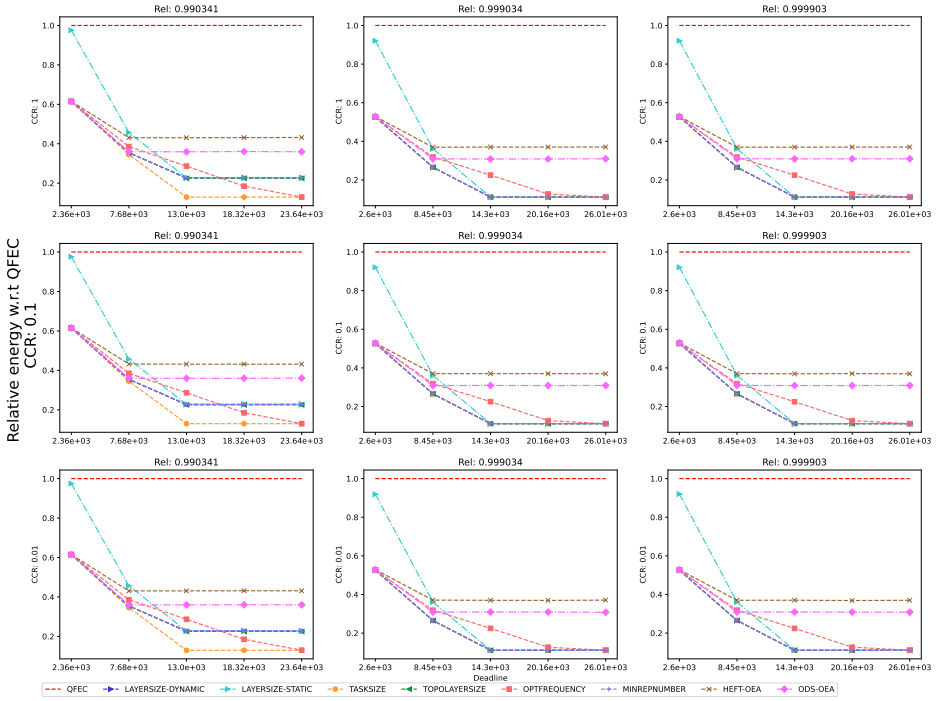

Figure 553: Performance of the different heuristics on the Cholesky workflow.

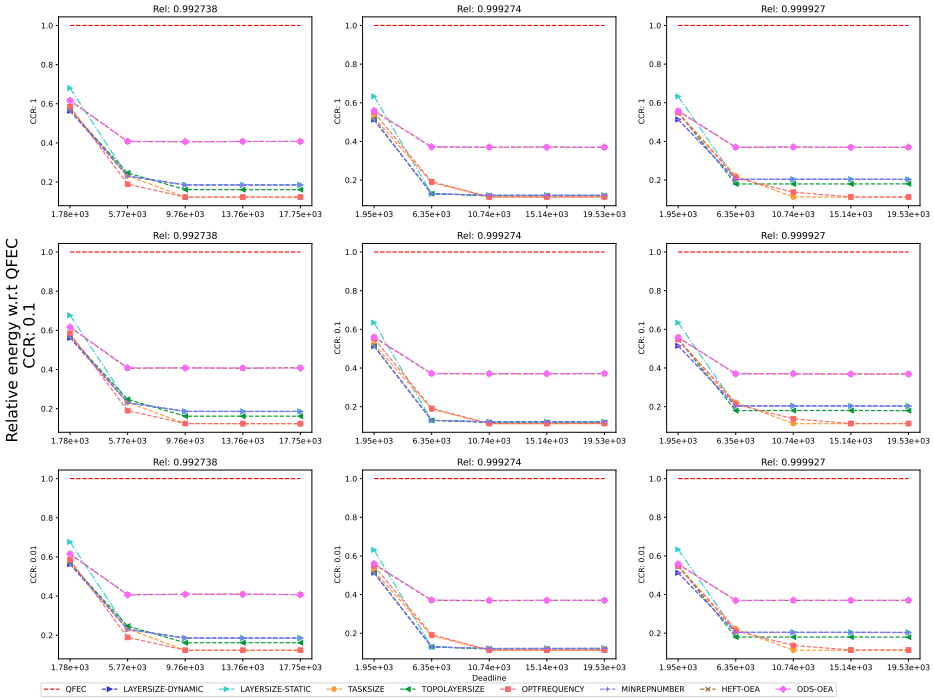

Figure 554: Performance of the different heuristics on the Cycles workflow.

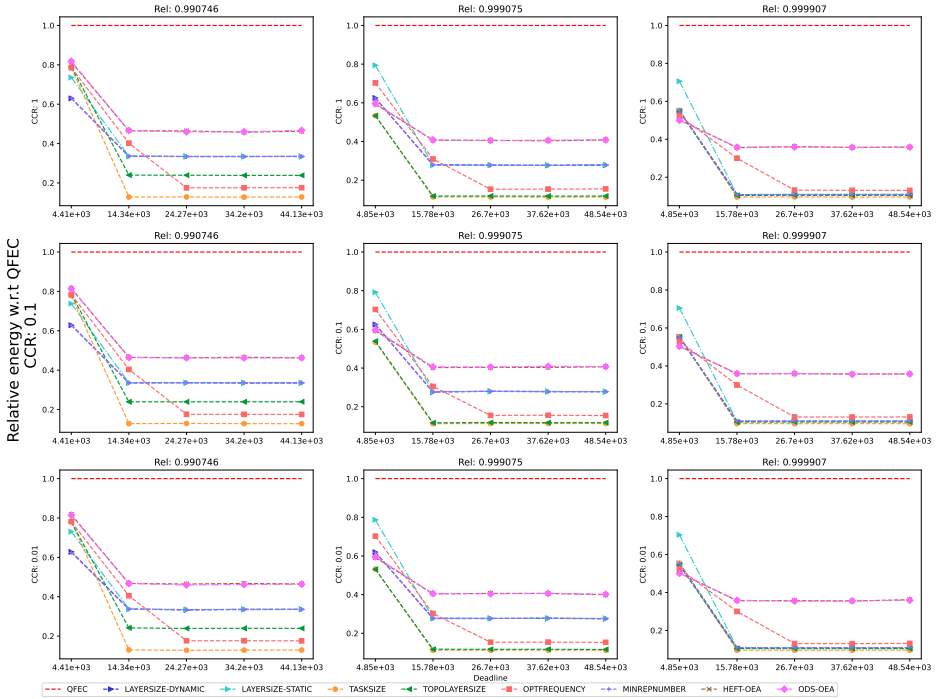

Figure 555: Performance of the different heuristics on the Epigenomics workflow.

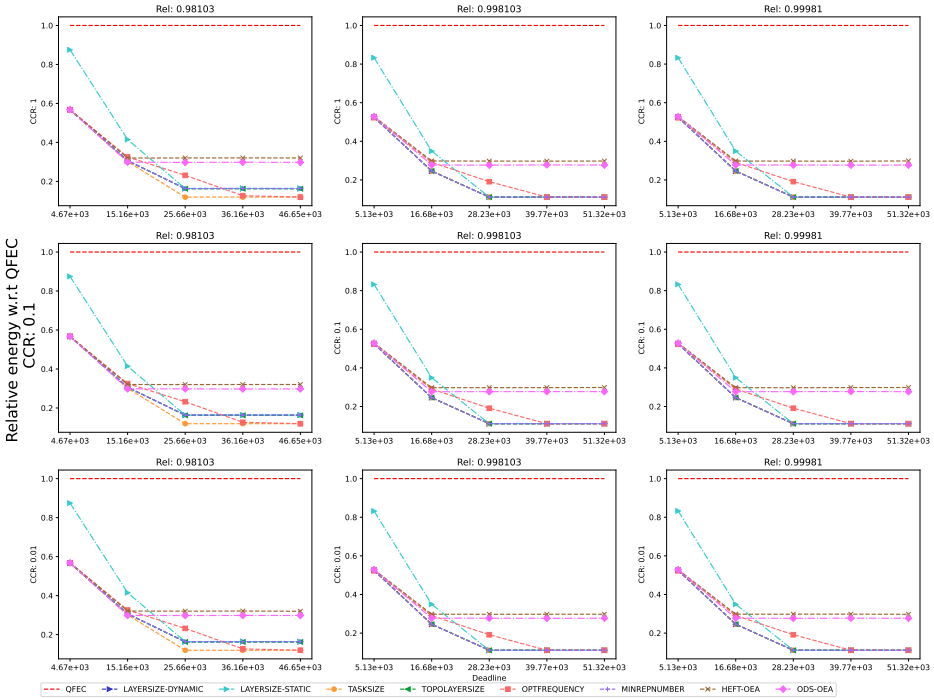

Figure 556: Performance of the different heuristics on the LU workflow.

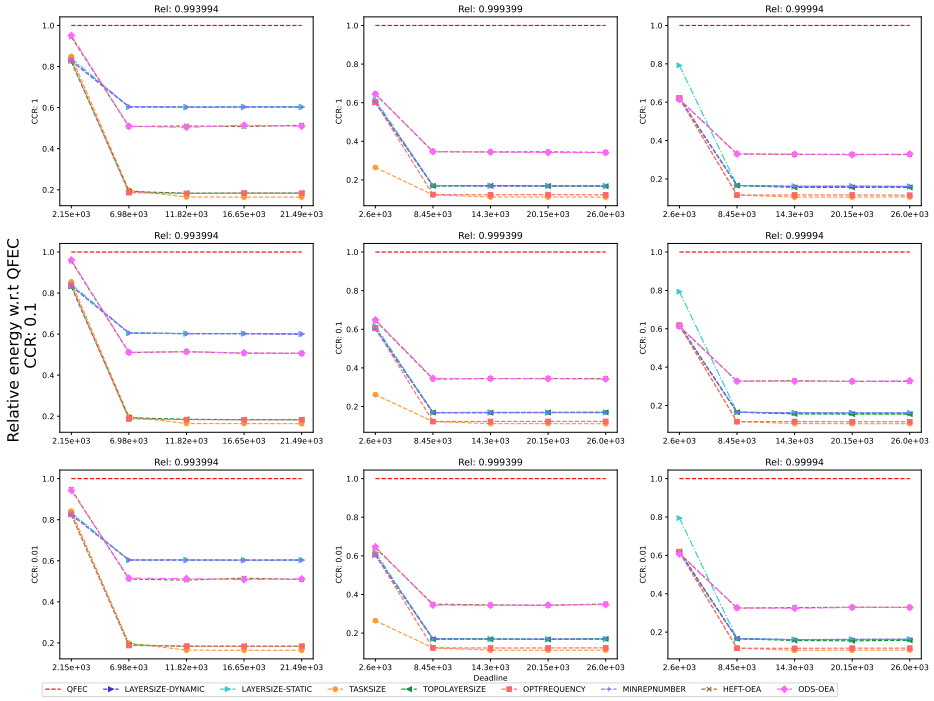

Figure 557: Performance of the different heuristics on the Montage workflow.

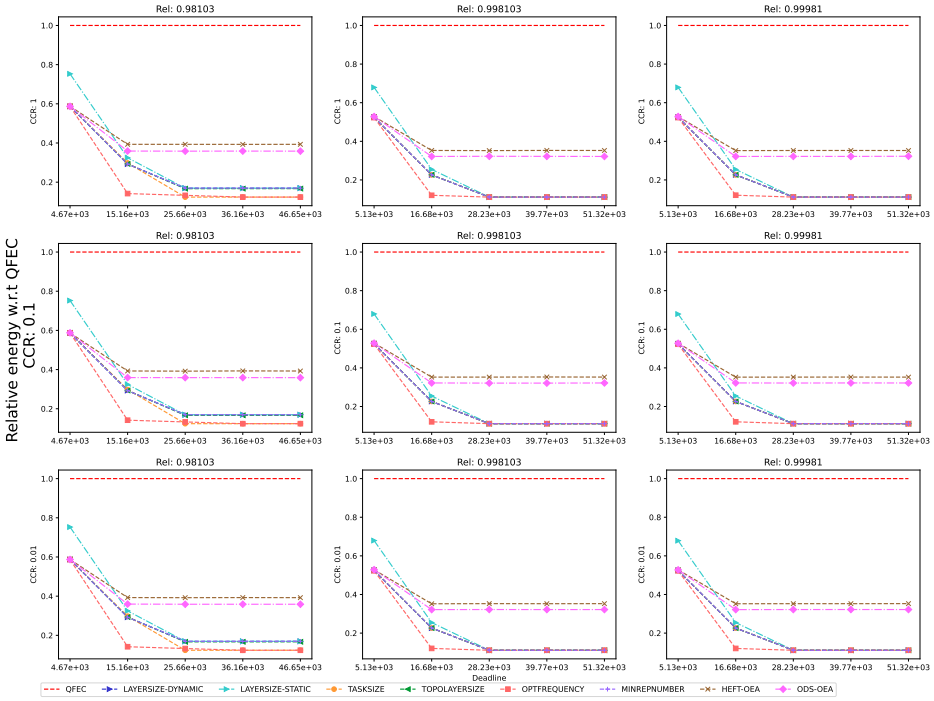

Figure 558: Performance of the different heuristics on the QR workflow.

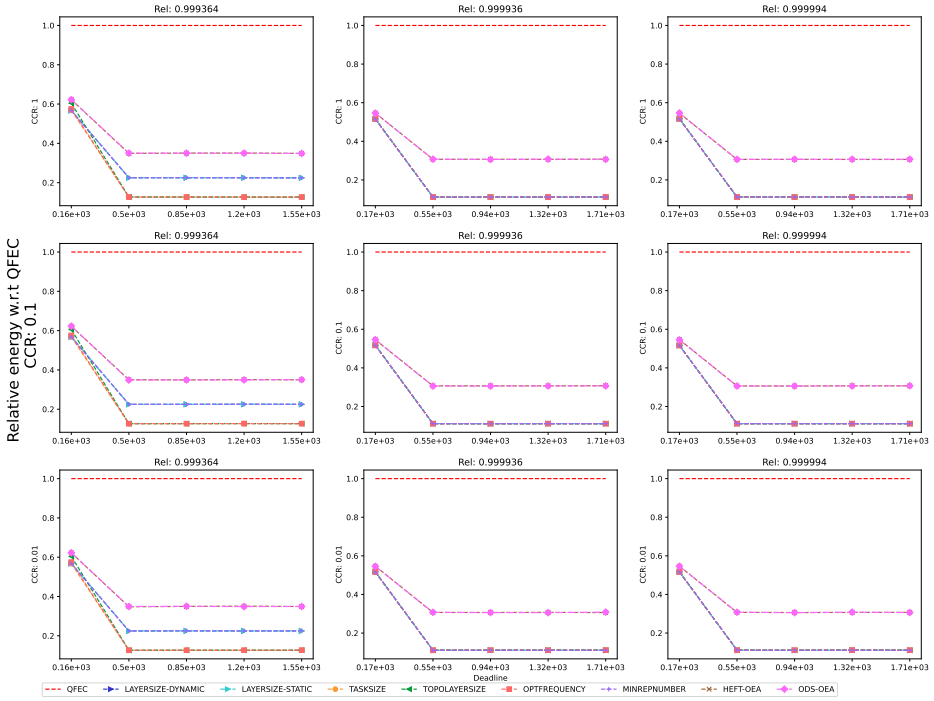

Figure 559: Performance of the different heuristics on the Seismology workflow.

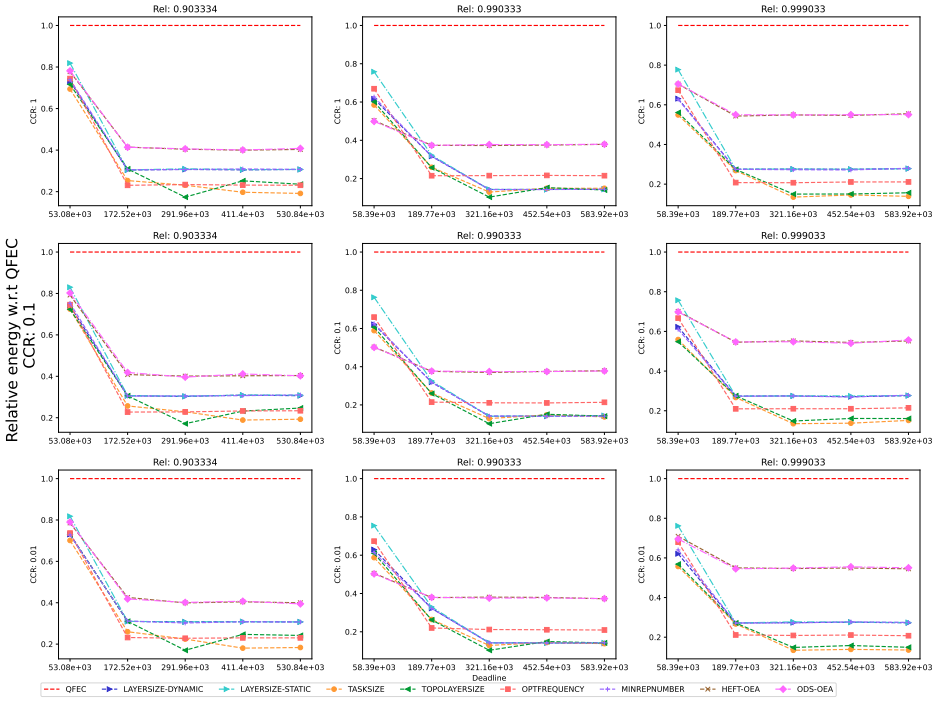

Figure 560: Performance of the different heuristics on the SoyKB workflow.

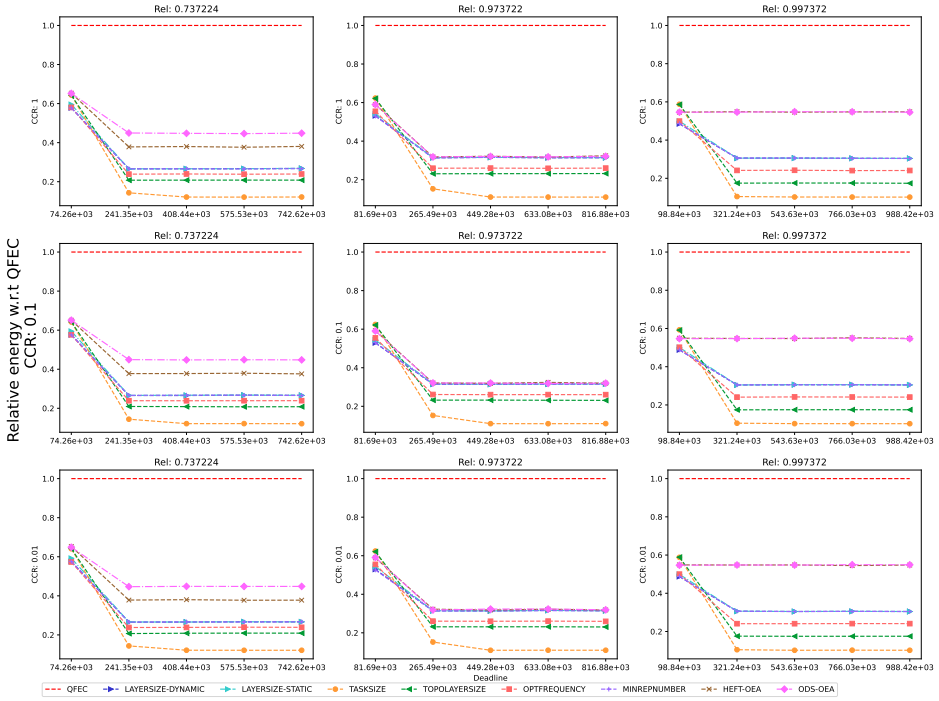

Figure 561: Performance of the different heuristics on the SRASearch.

E.4 $BC/WC = 0.4$

E.4.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

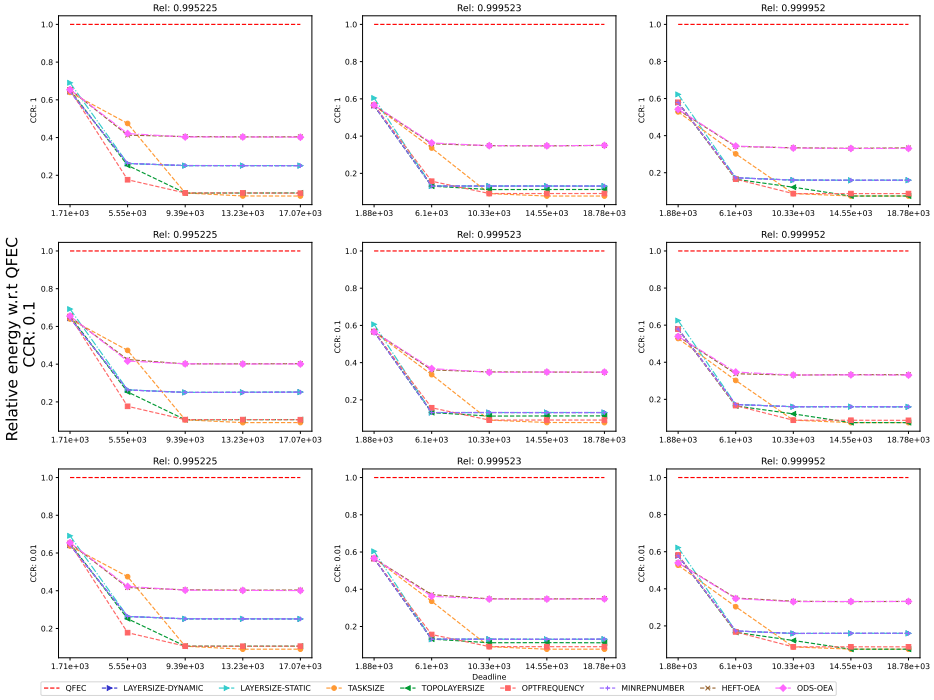

Figure 562: Performance of the different heuristics on the BWA workflow.

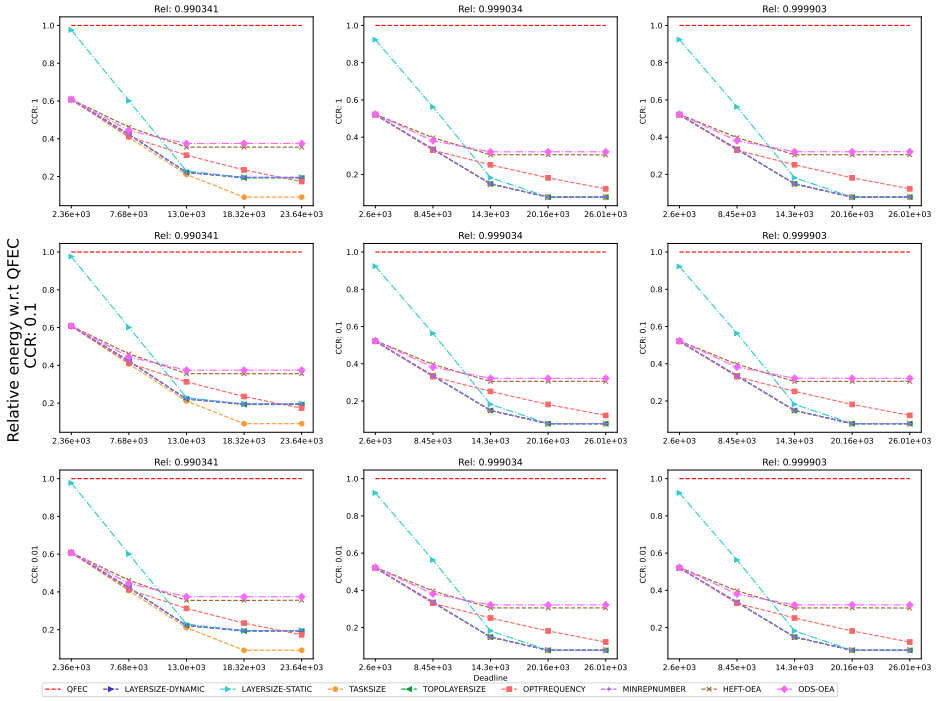

Figure 563: Performance of the different heuristics on the Cholesky workflow.

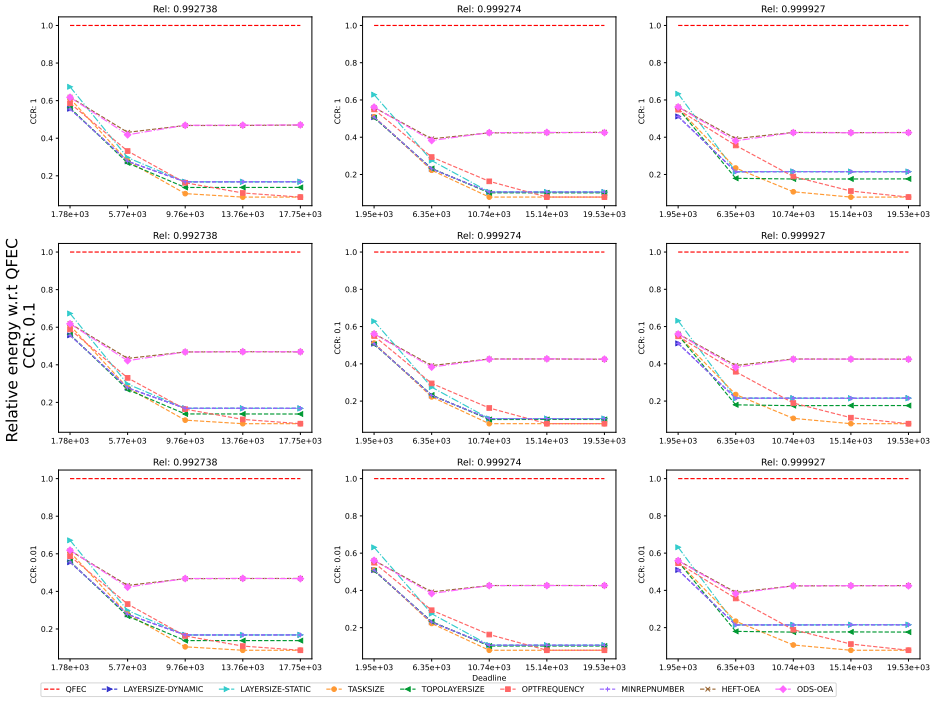

Figure 564: Performance of the different heuristics on the Cycles workflow.

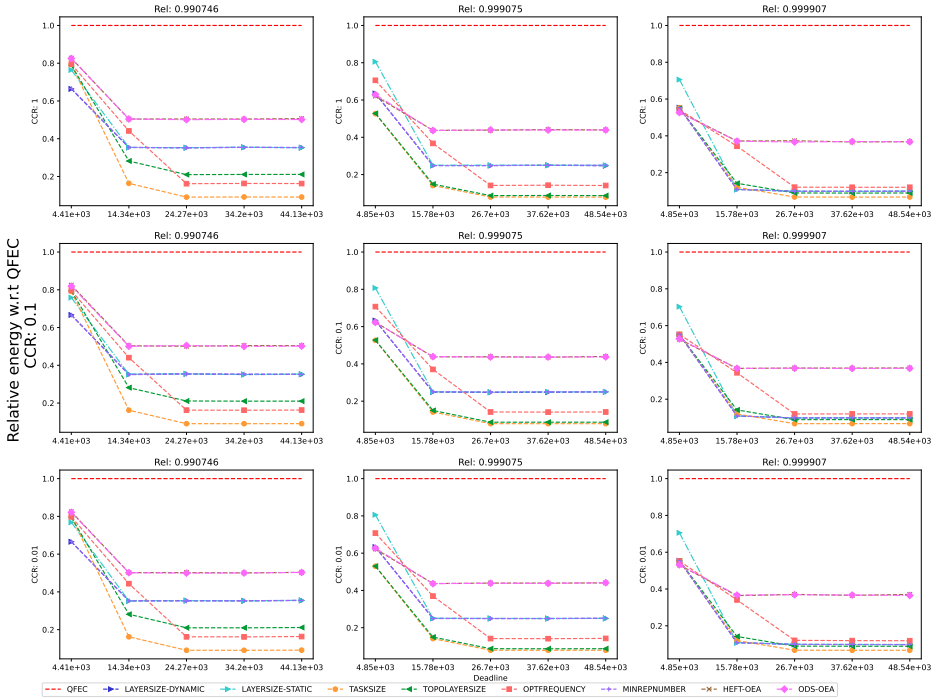

Figure 565: Performance of the different heuristics on the Epigenomics workflow.

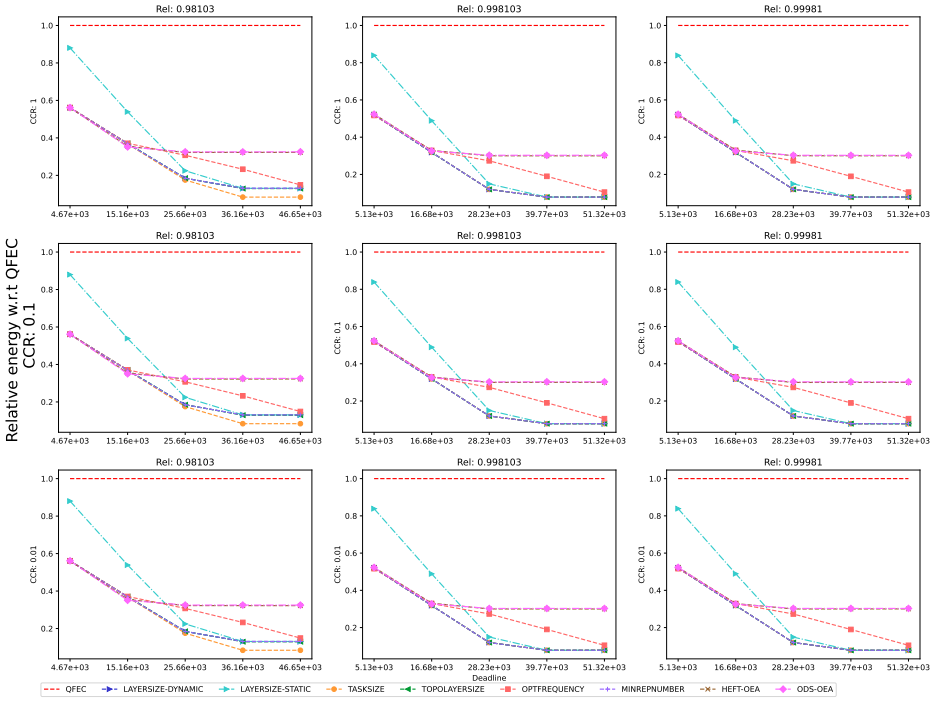

Figure 566: Performance of the different heuristics on the LU workflow.

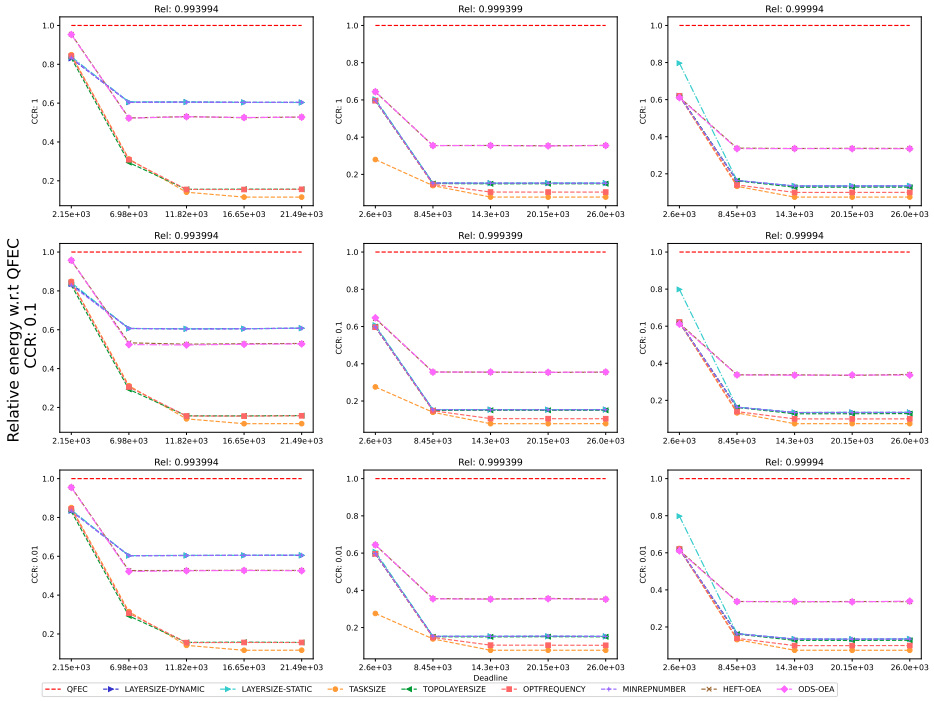

Figure 567: Performance of the different heuristics on the Montage workflow.

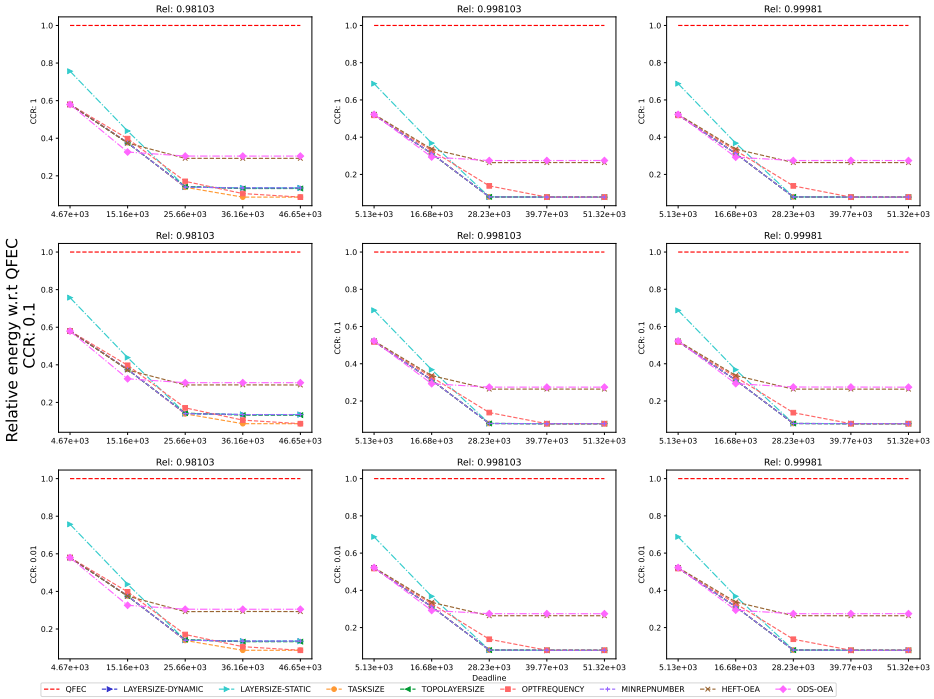

Figure 568: Performance of the different heuristics on the QR workflow.

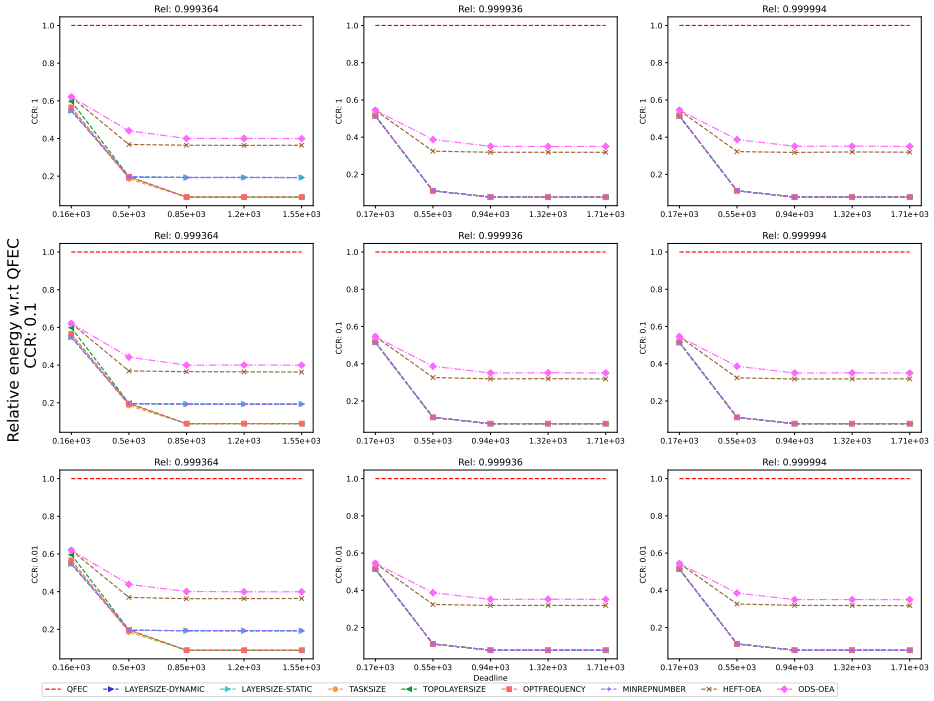

Figure 569: Performance of the different heuristics on the Seismology workflow.

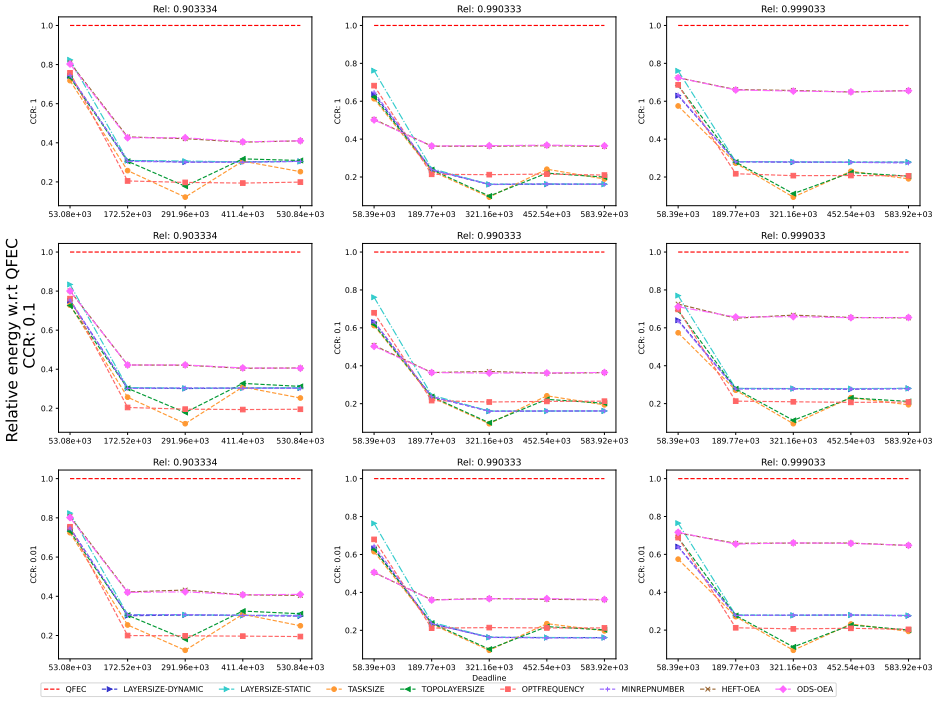

Figure 570: Performance of the different heuristics on the SoyKB workflow.

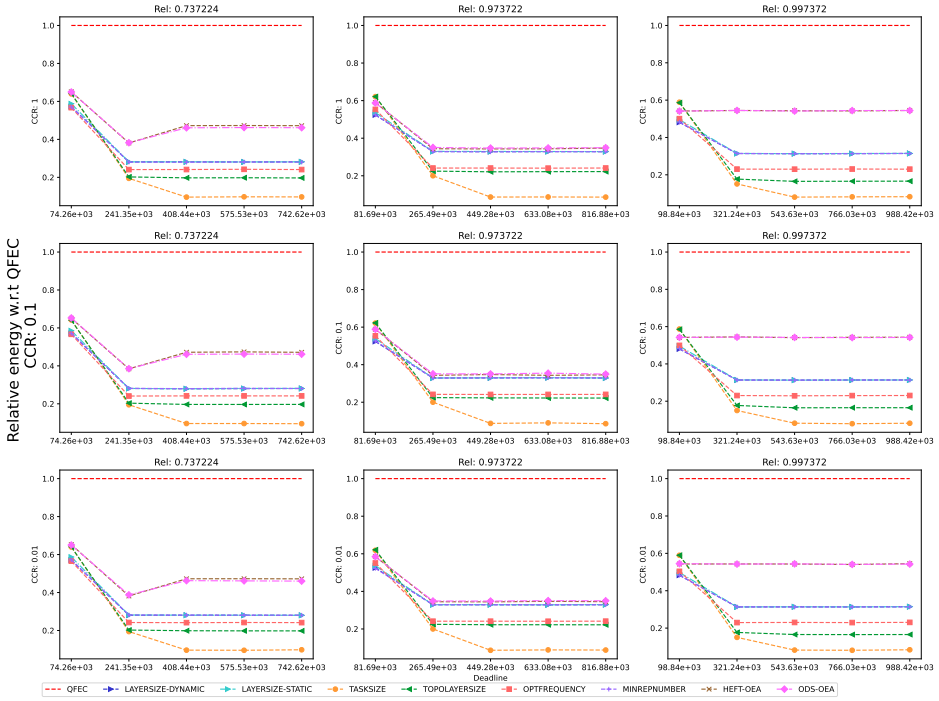

Figure 571: Performance of the different heuristics on the SRASearch.

E.4.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

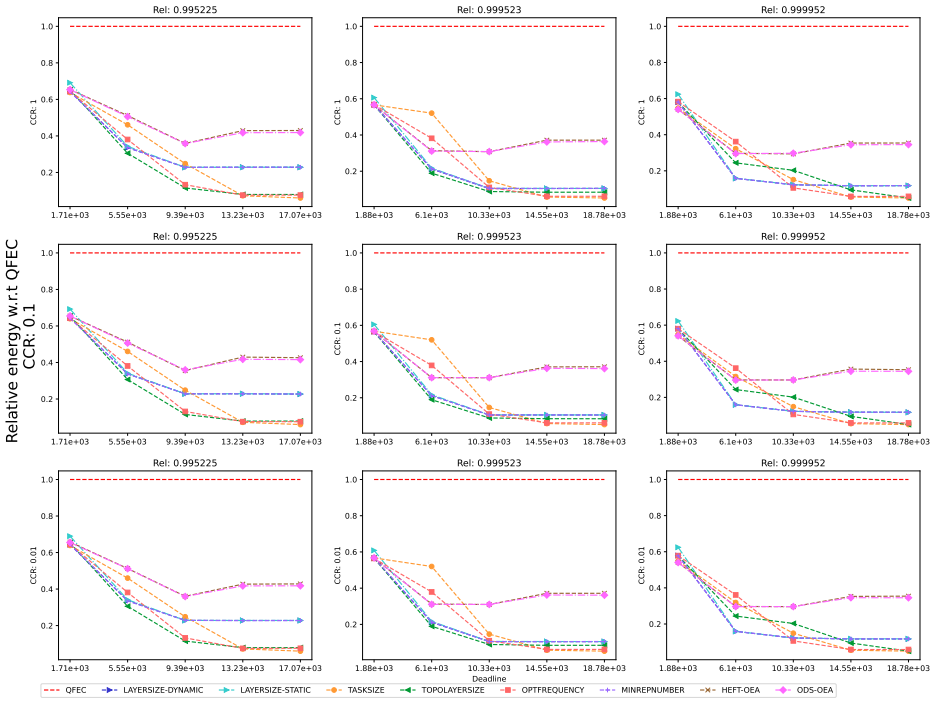

Figure 572: Performance of the different heuristics on the BWA workflow.

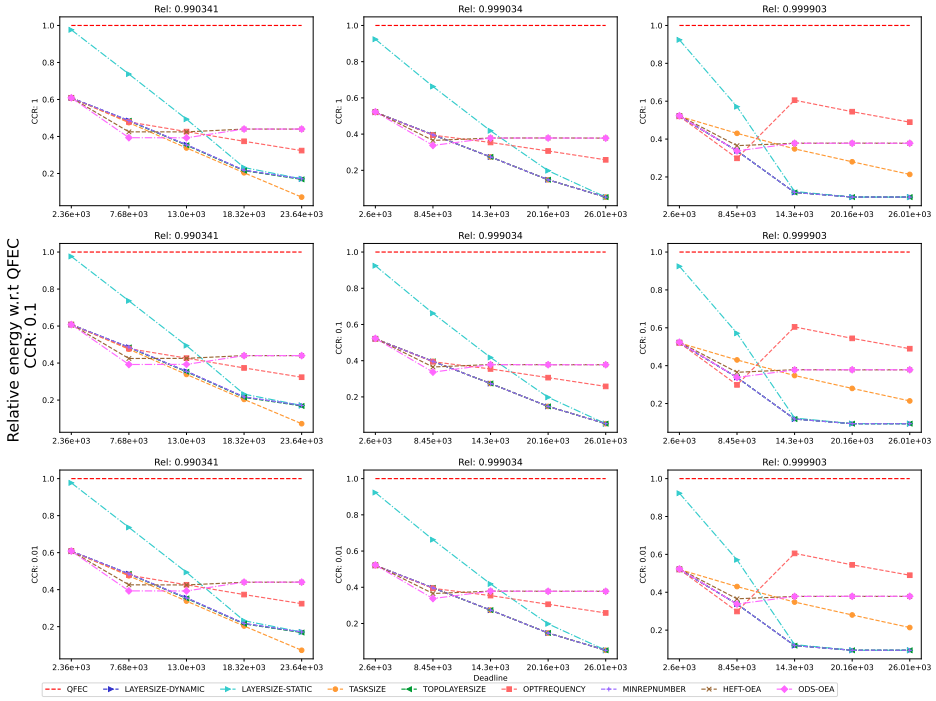

Figure 573: Performance of the different heuristics on the Cholesky workflow.

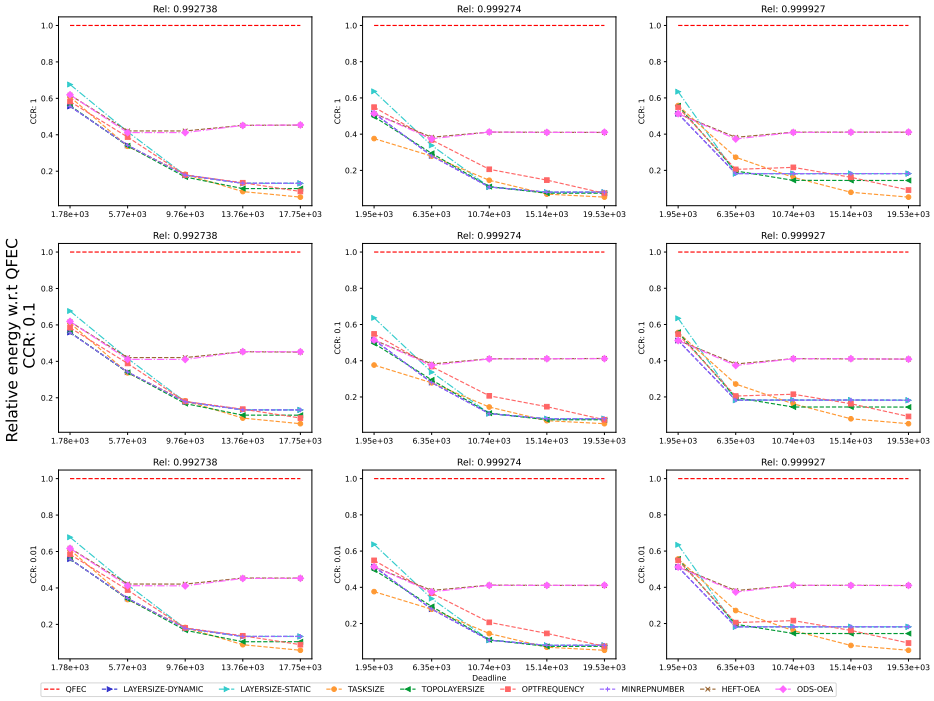

Figure 574: Performance of the different heuristics on the Cycles workflow.

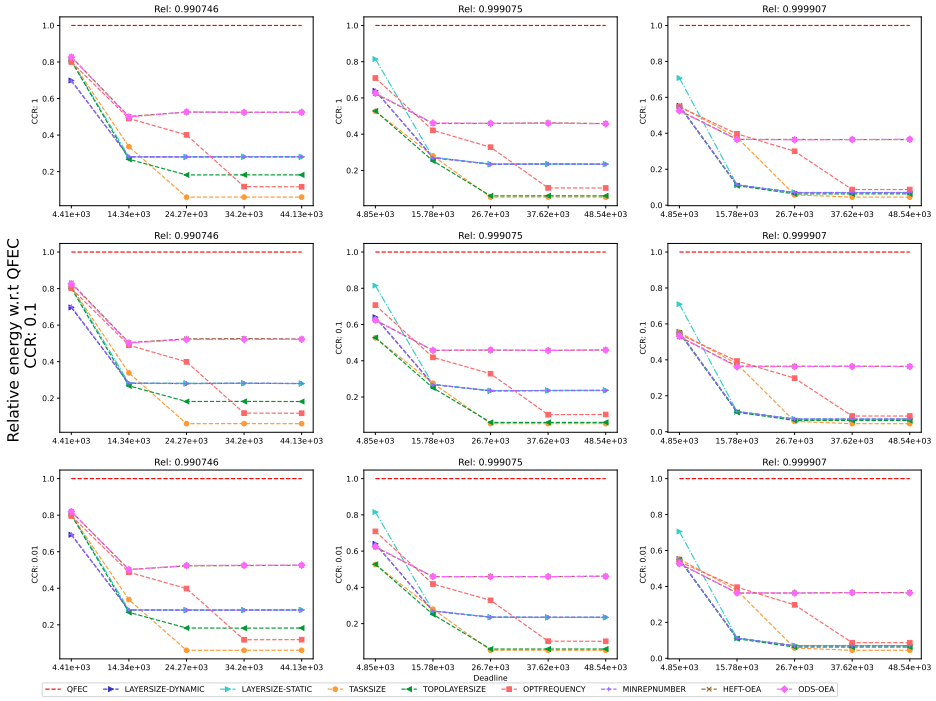

Figure 575: Performance of the different heuristics on the Epigenomics workflow.

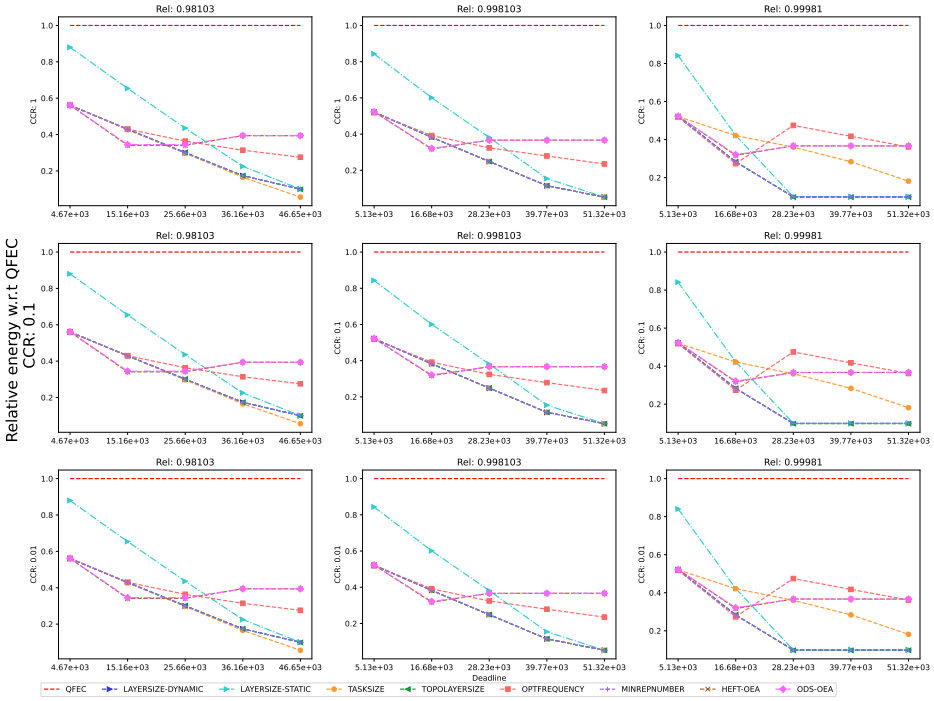

Figure 576: Performance of the different heuristics on the LU workflow.

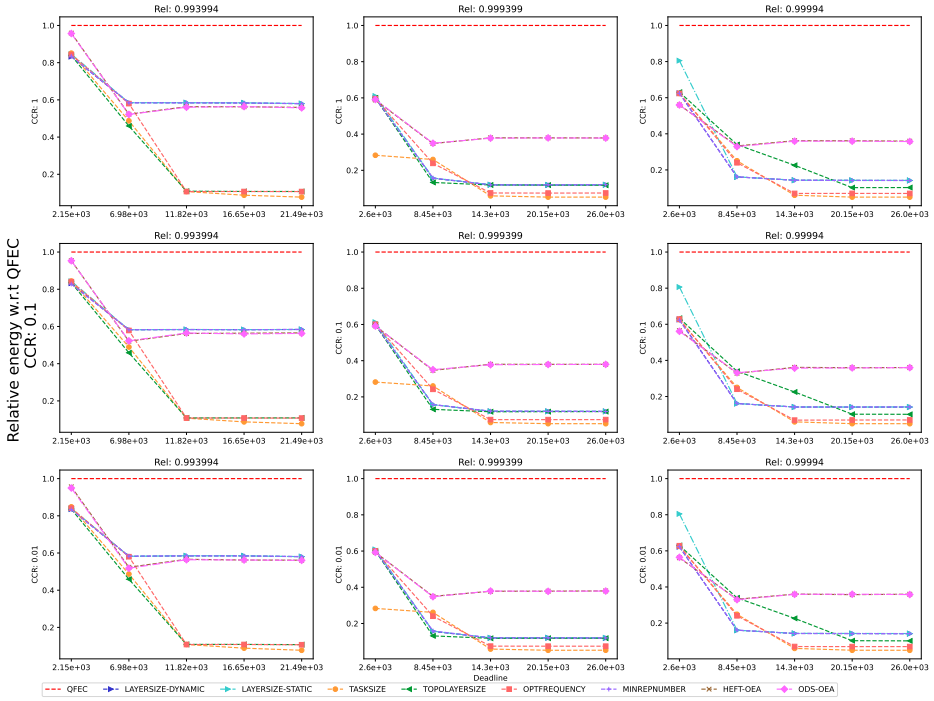

Figure 577: Performance of the different heuristics on the Montage workflow.

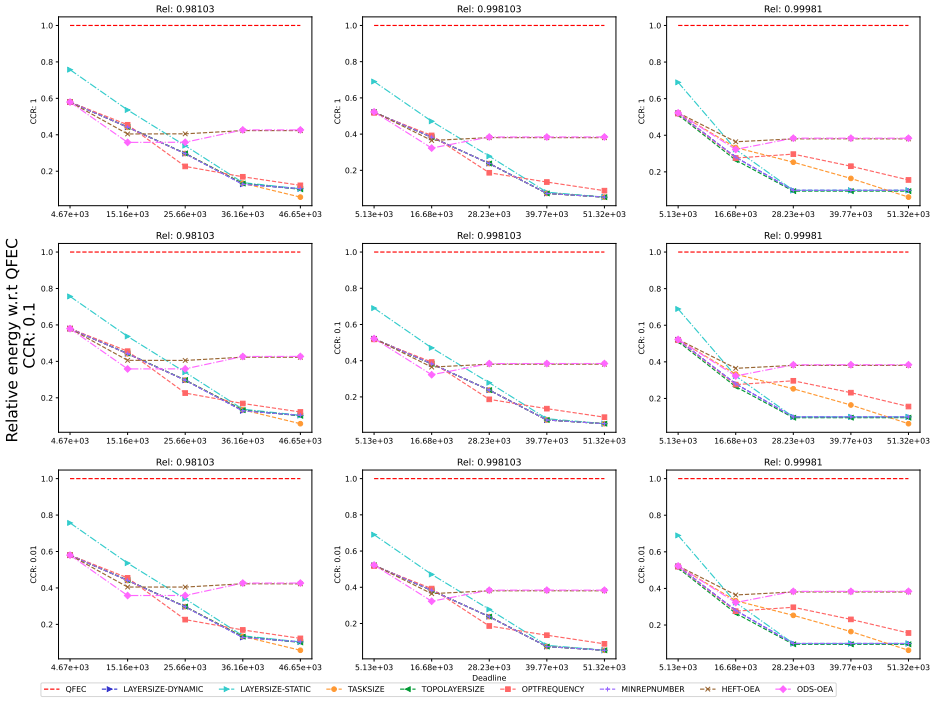

Figure 578: Performance of the different heuristics on the QR workflow.

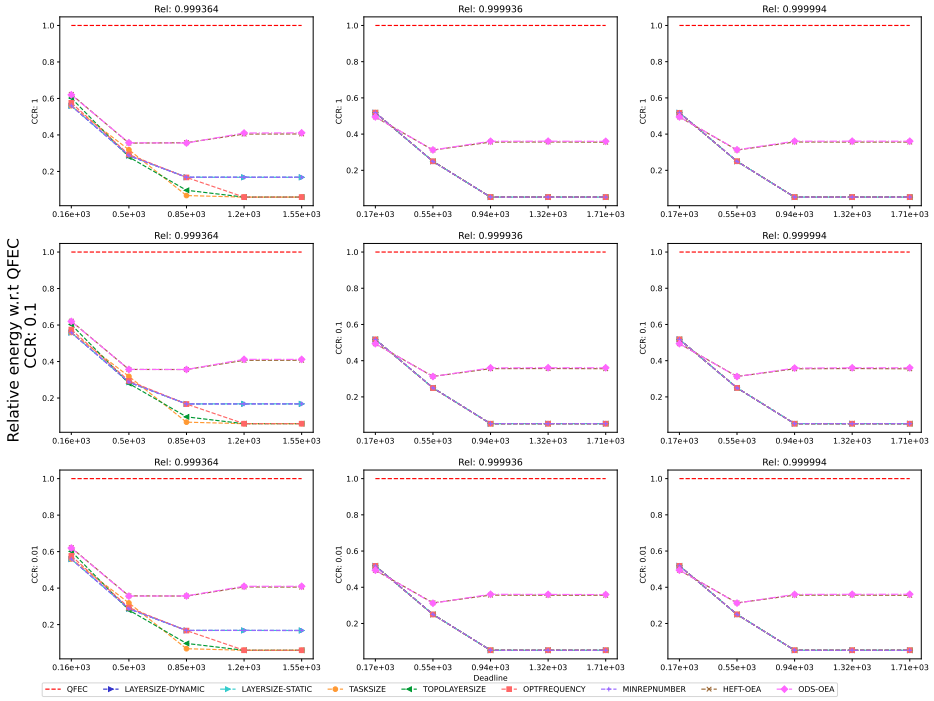

Figure 579: Performance of the different heuristics on the Seismology workflow.

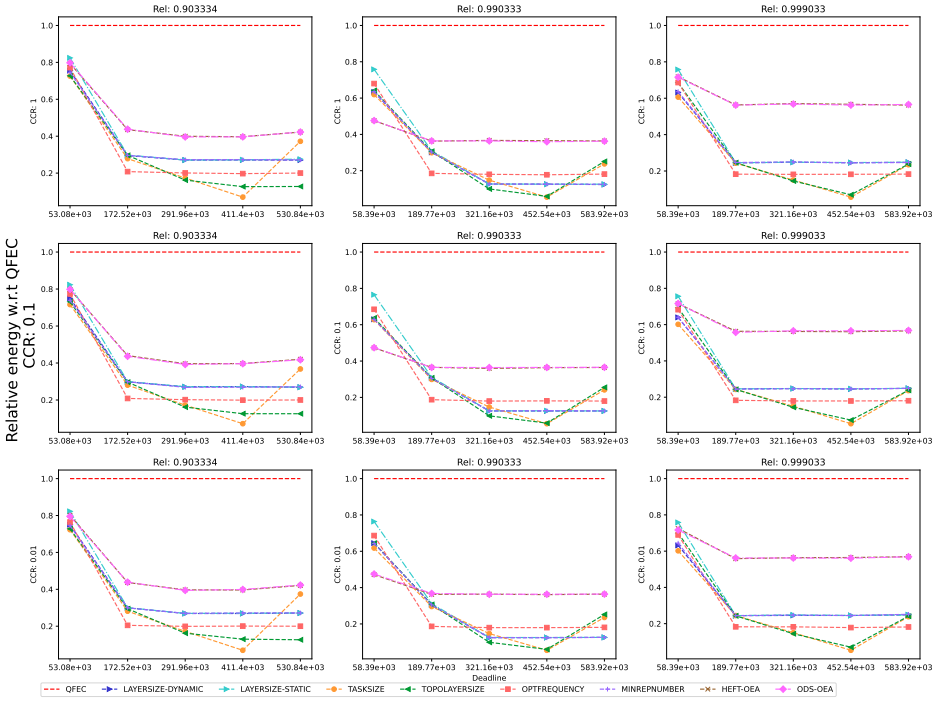

Figure 580: Performance of the different heuristics on the SoyKB workflow.

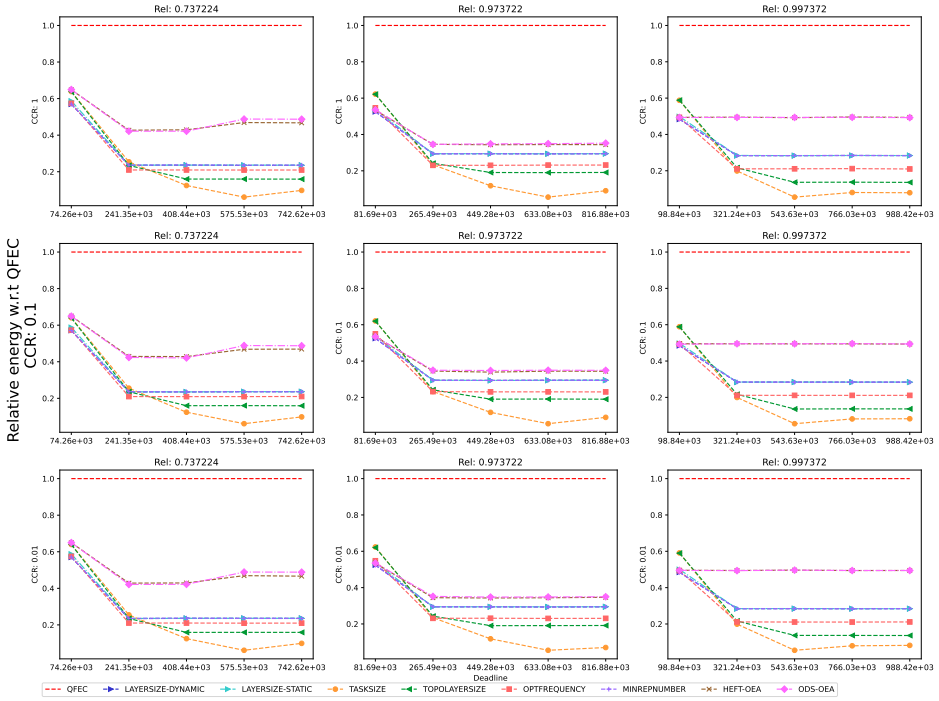

Figure 581: Performance of the different heuristics on the SRASearch.

E.4.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

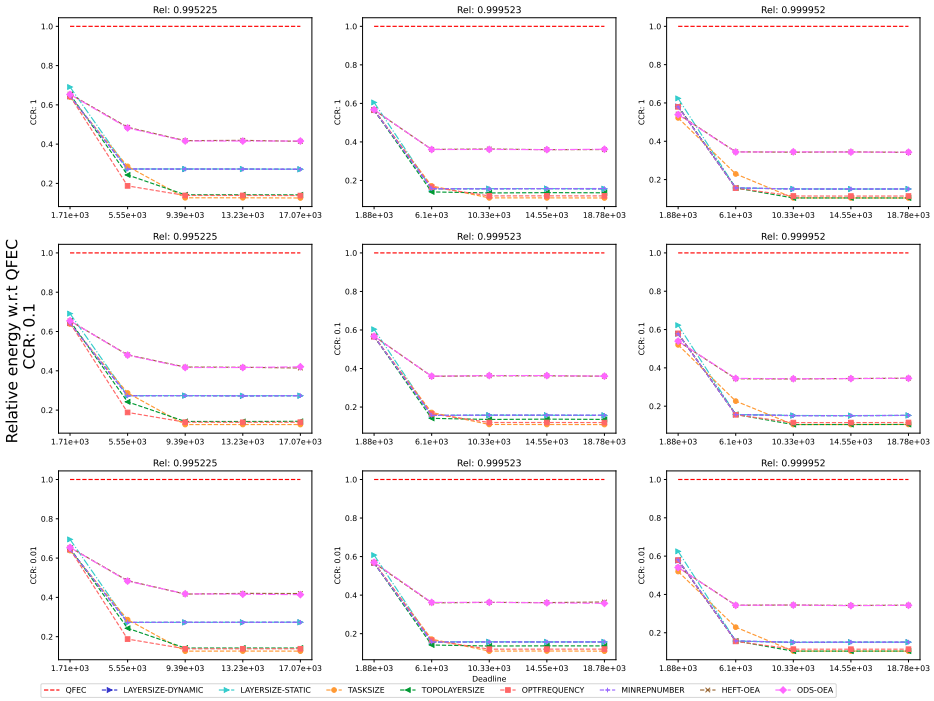

Figure 582: Performance of the different heuristics on the BWA workflow.

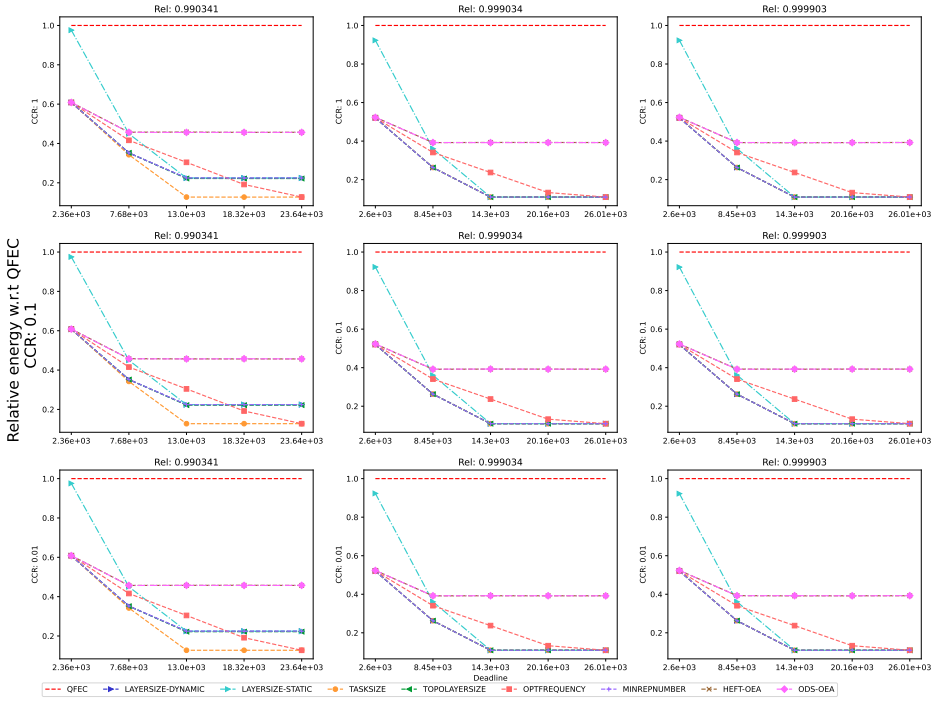

Figure 583: Performance of the different heuristics on the Cholesky workflow.

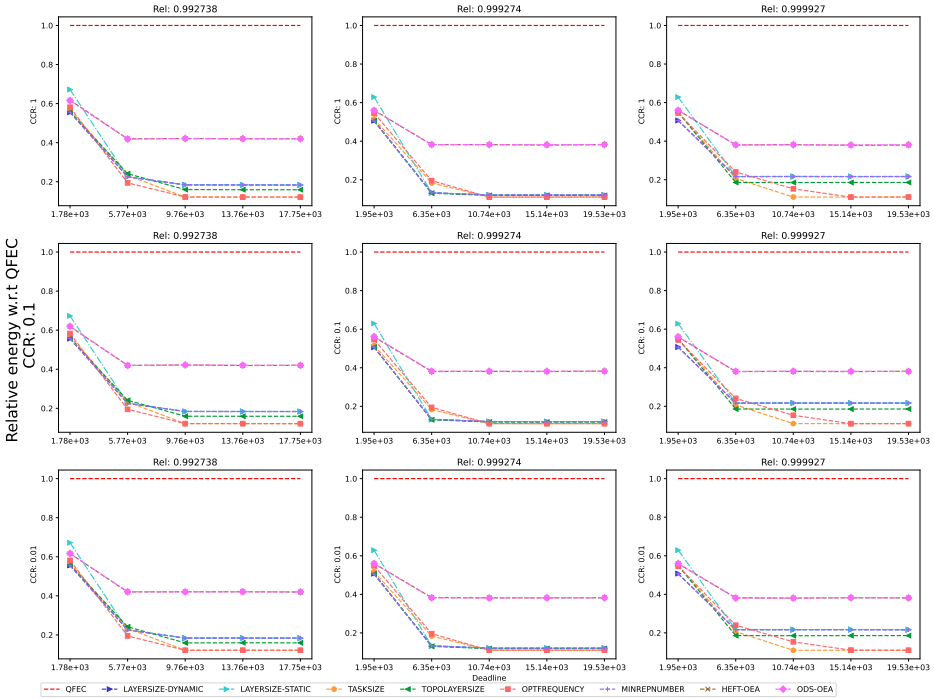

Figure 584: Performance of the different heuristics on the Cycles workflow.

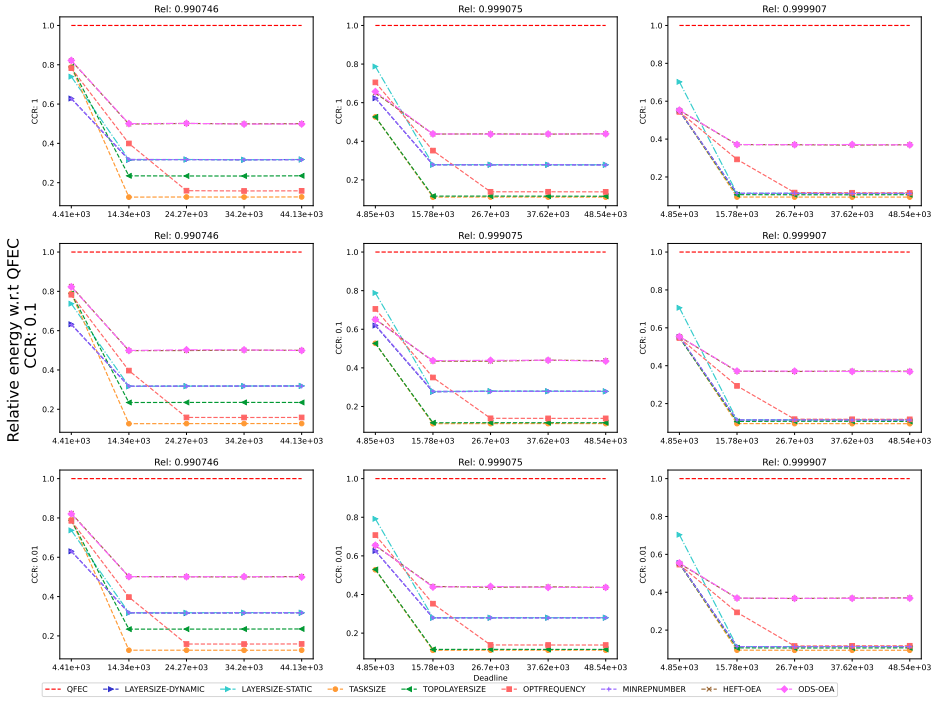

Figure 585: Performance of the different heuristics on the Epigenomics workflow.

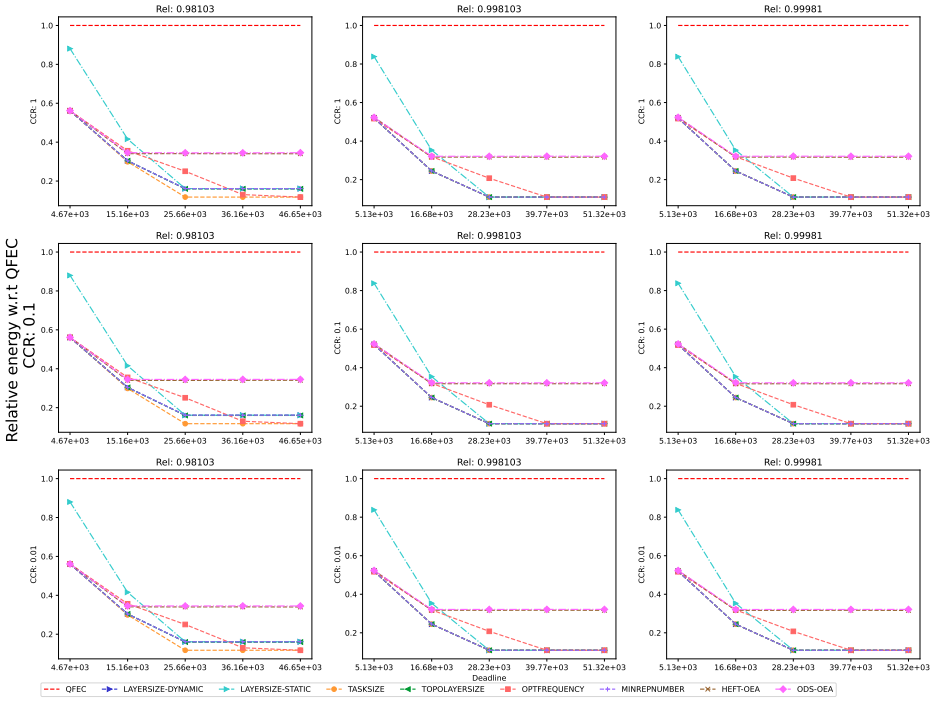

Figure 586: Performance of the different heuristics on the LU workflow.

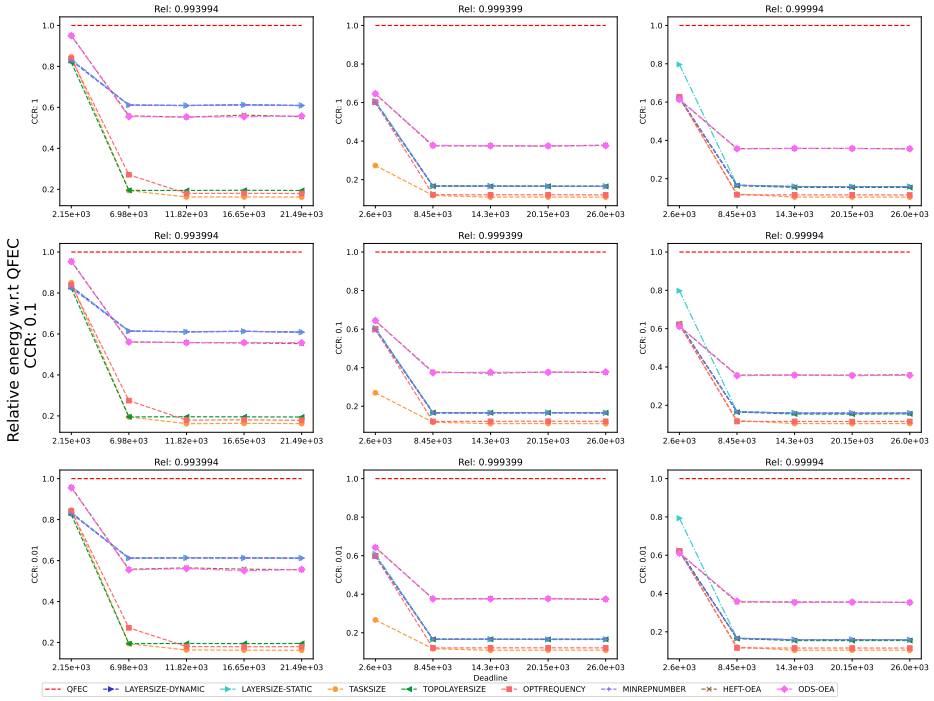

Figure 587: Performance of the different heuristics on the Montage workflow.

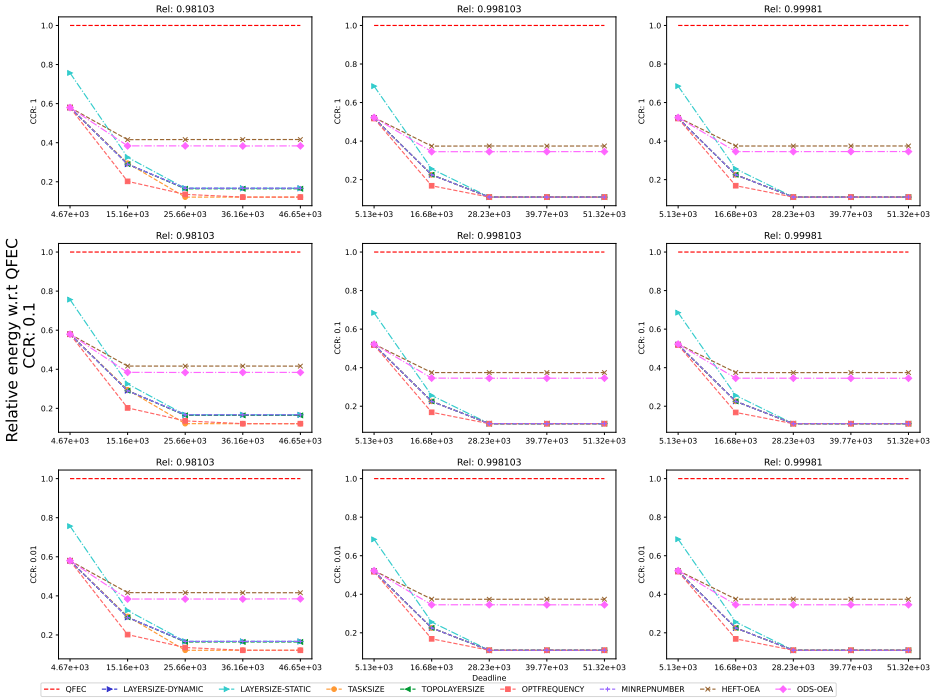

Figure 588: Performance of the different heuristics on the QR workflow.

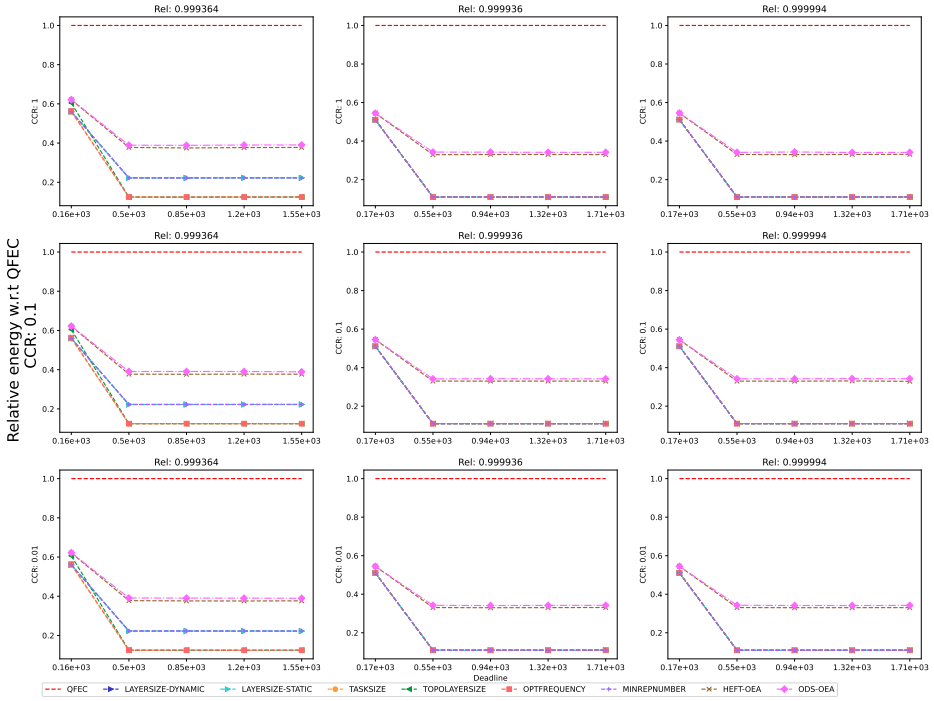

Figure 589: Performance of the different heuristics on the Seismology workflow.

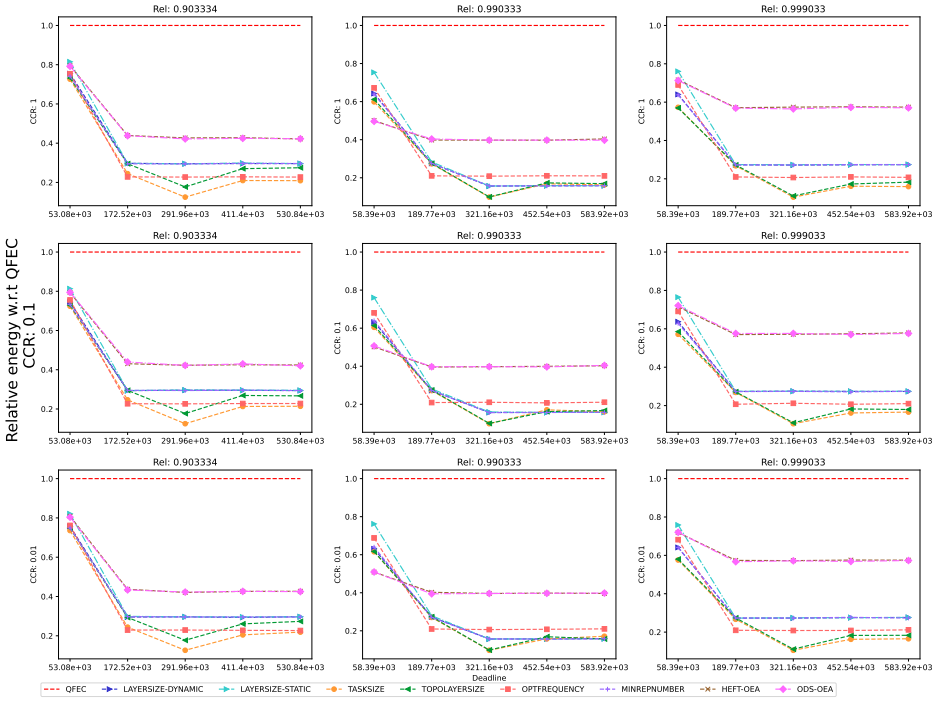

Figure 590: Performance of the different heuristics on the SoyKB workflow.

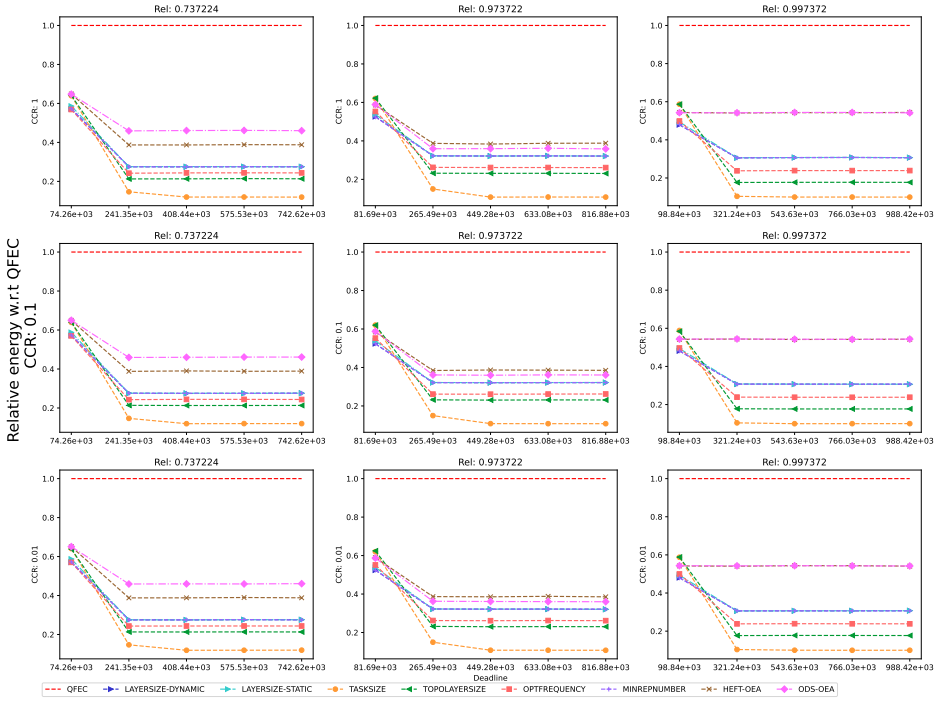

Figure 591: Performance of the different heuristics on the SRASearch.

E.4.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

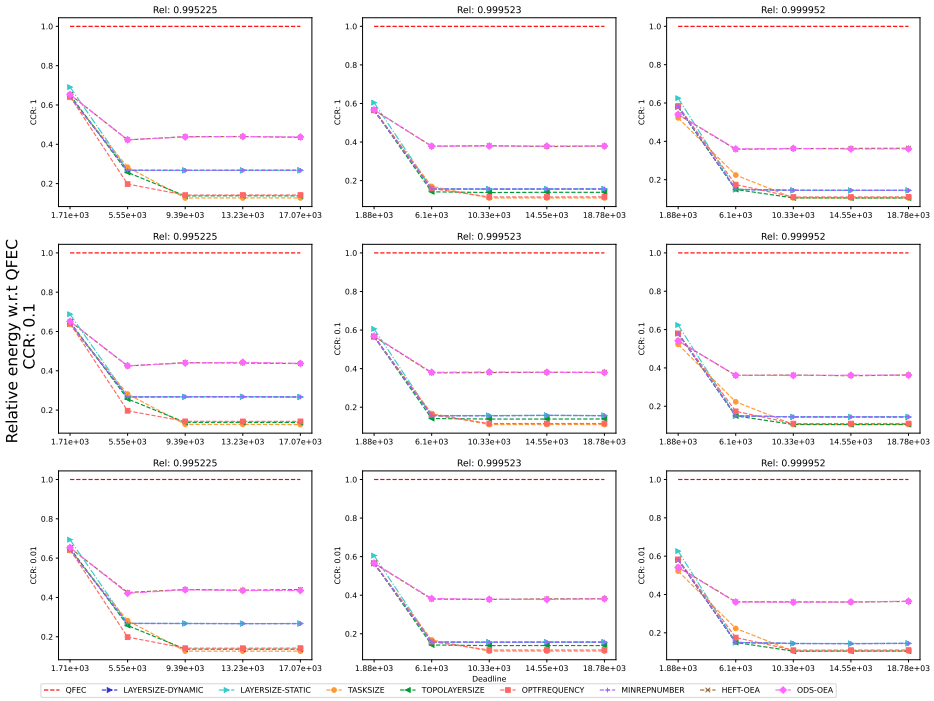

Figure 592: Performance of the different heuristics on the BWA workflow.

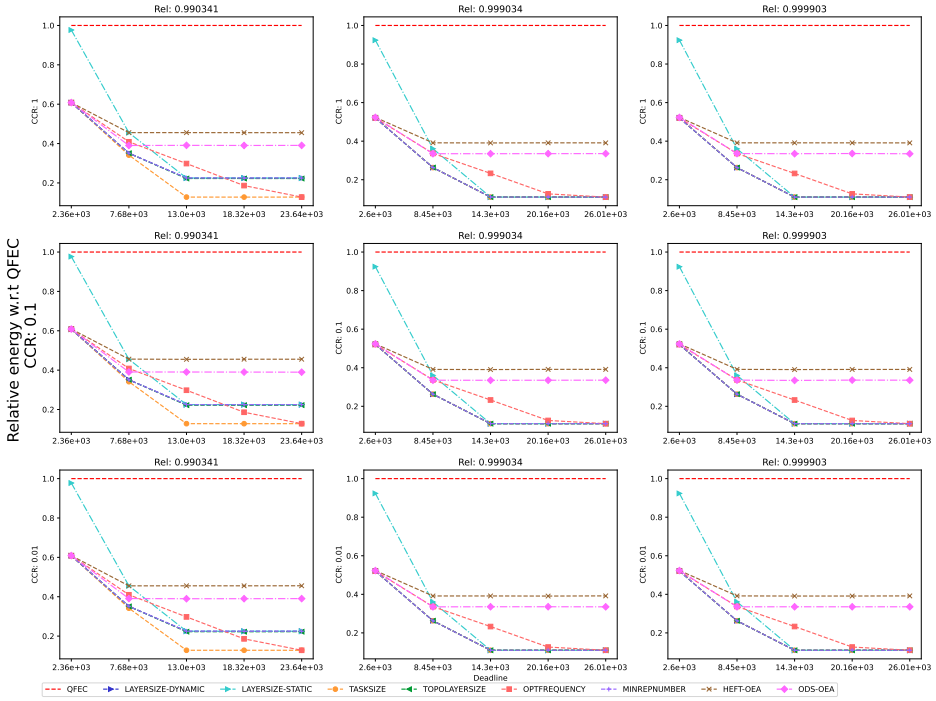

Figure 593: Performance of the different heuristics on the Cholesky workflow.

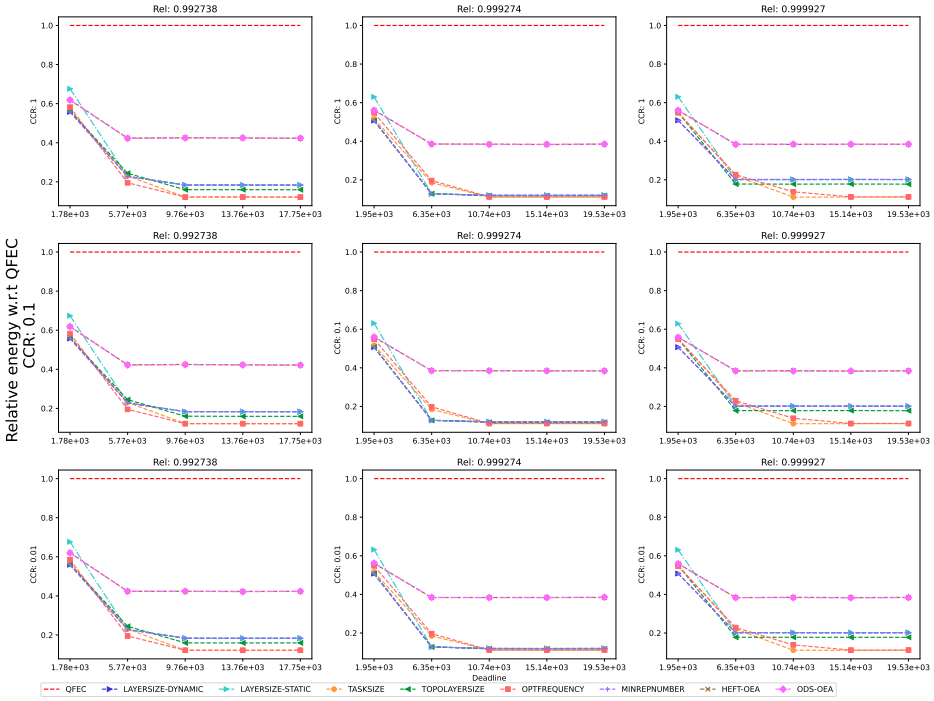

Figure 594: Performance of the different heuristics on the Cycles workflow.

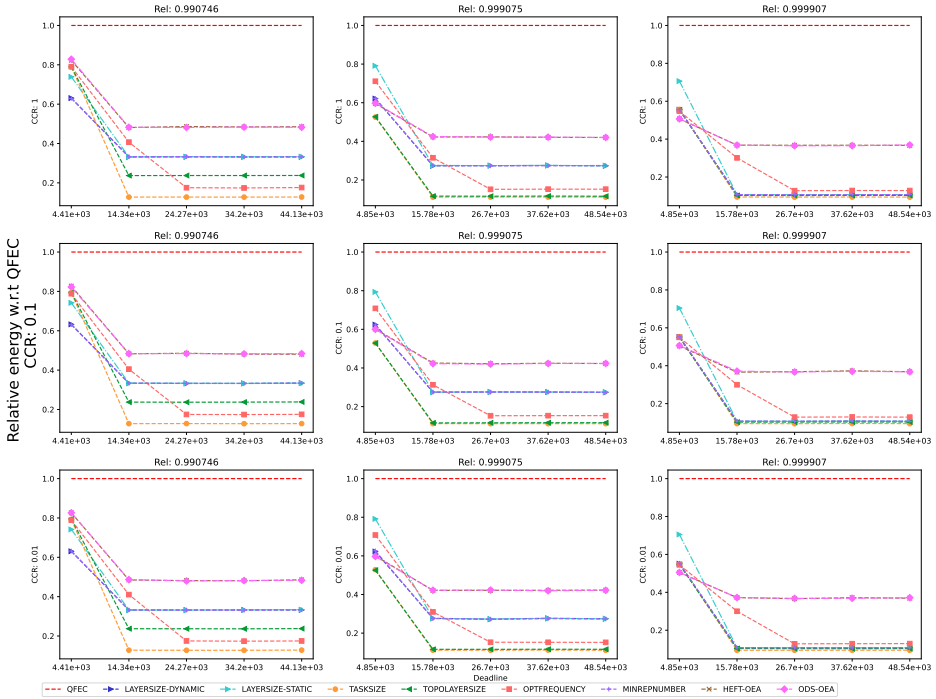

Figure 595: Performance of the different heuristics on the Epigenomics workflow.

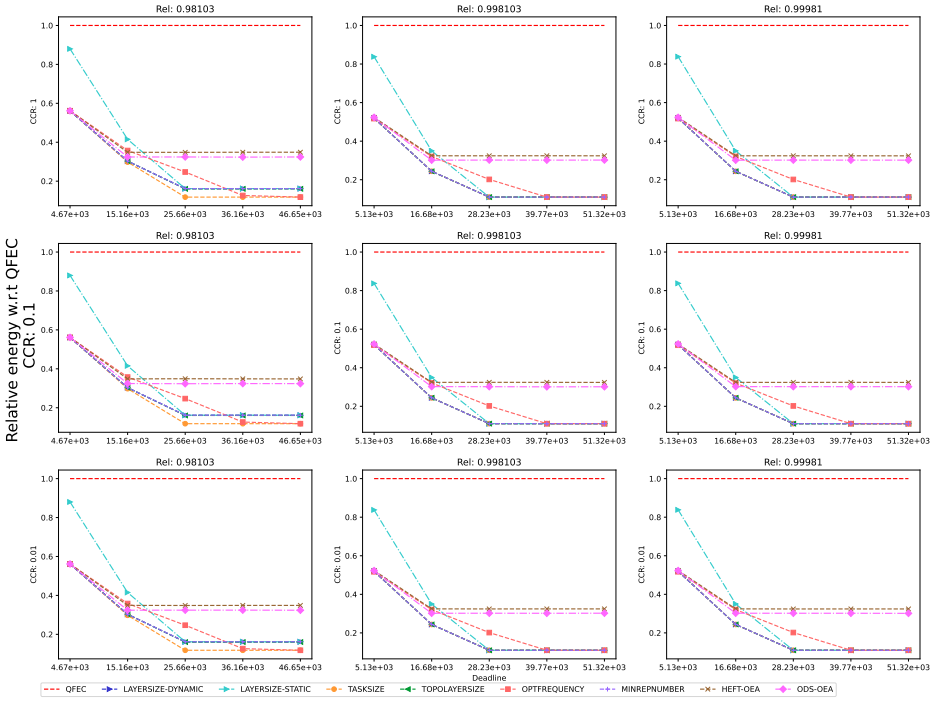

Figure 596: Performance of the different heuristics on the LU workflow.

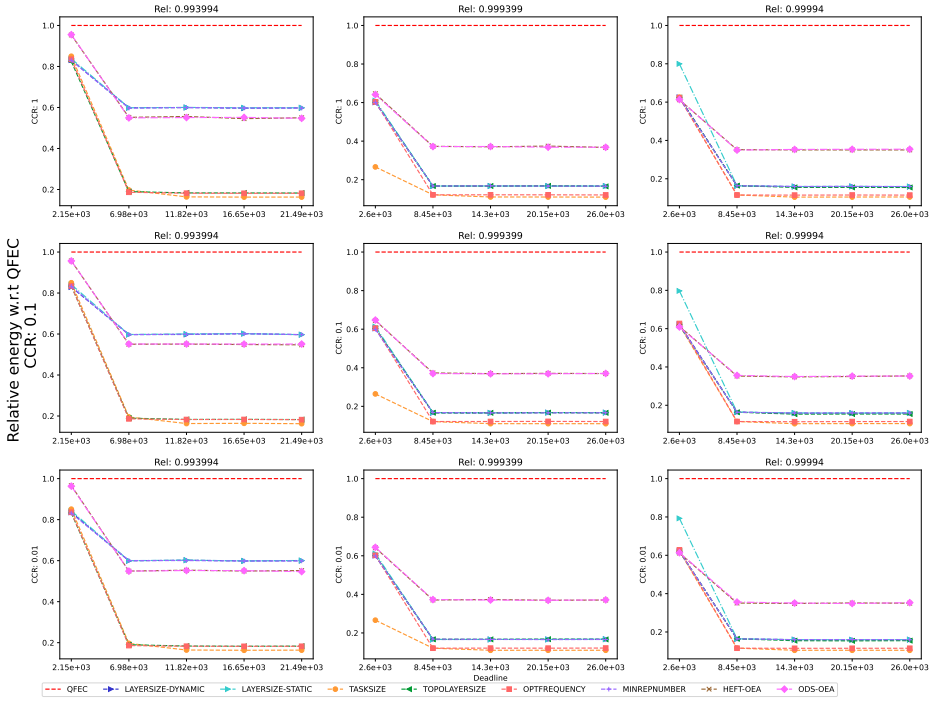

Figure 597: Performance of the different heuristics on the Montage workflow.

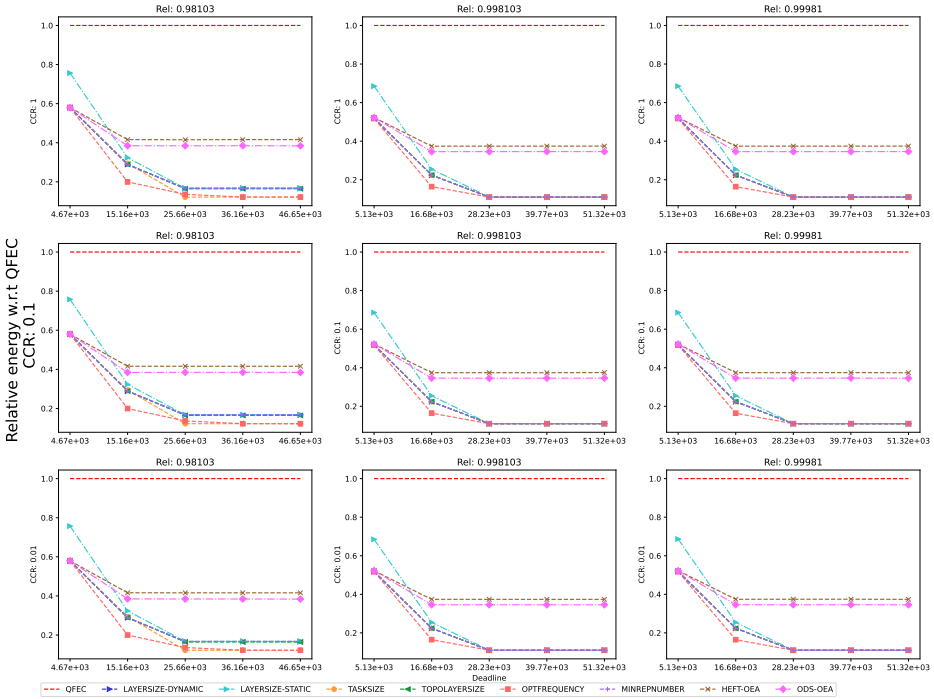

Figure 598: Performance of the different heuristics on the QR workflow.

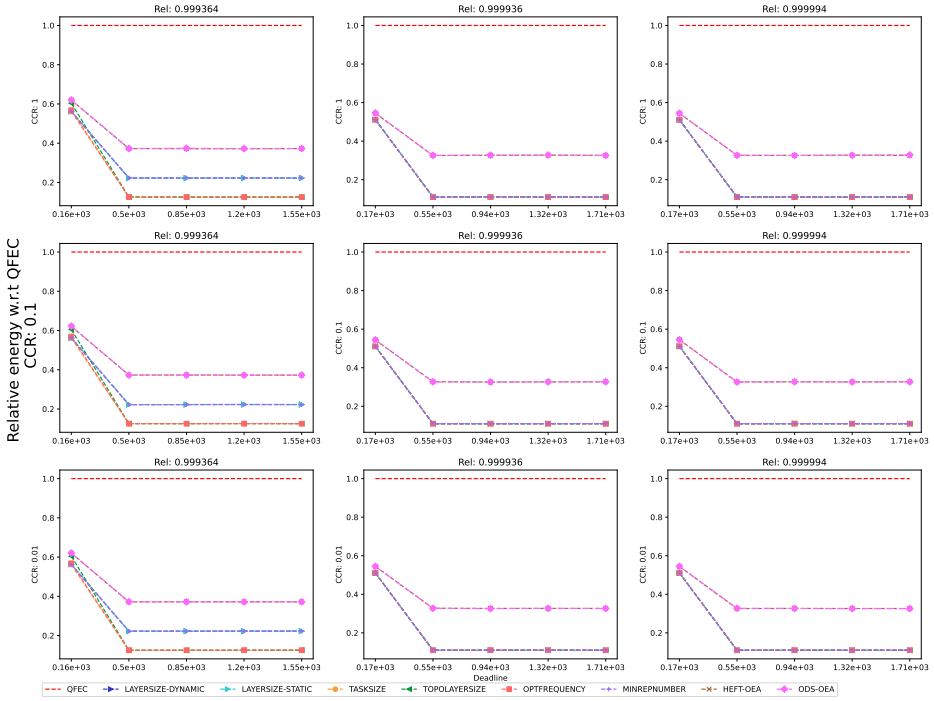

Figure 599: Performance of the different heuristics on the Seismology workflow.

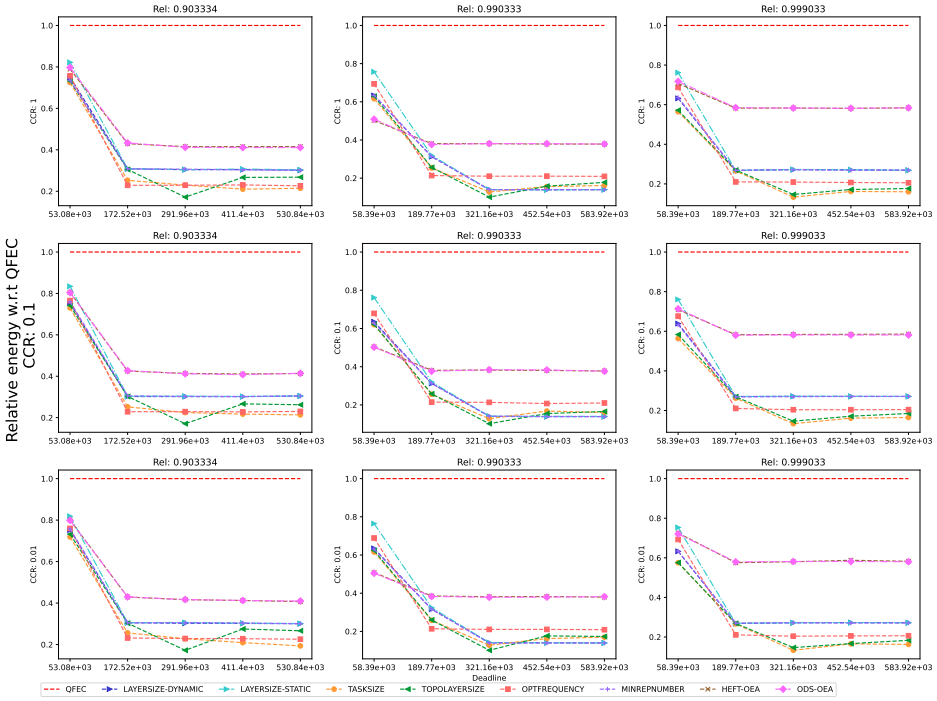

Figure 600: Performance of the different heuristics on the SoyKB workflow.

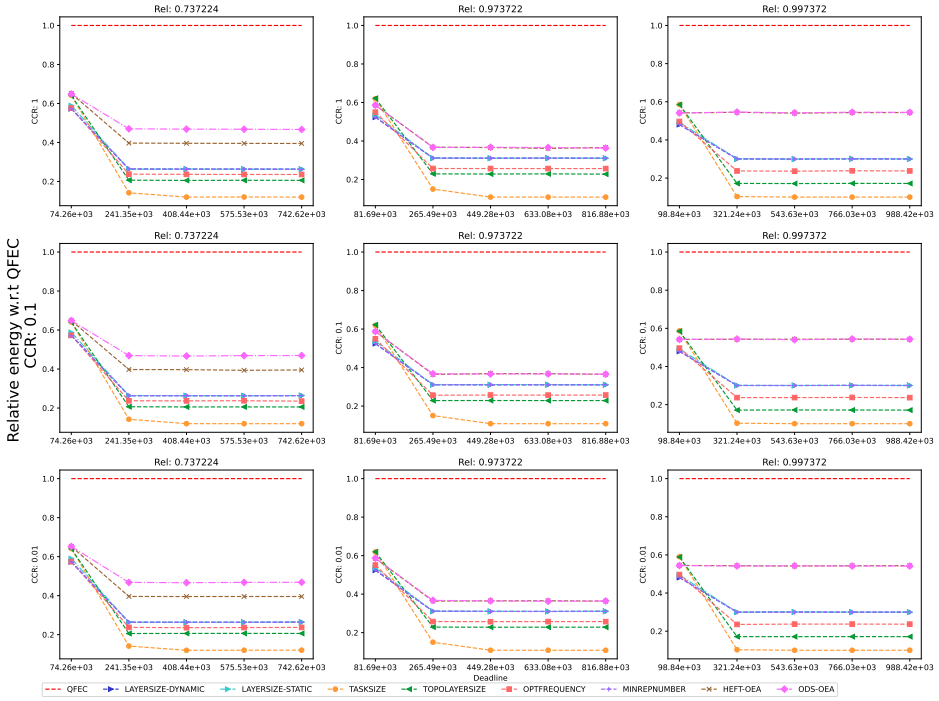

Figure 601: Performance of the different heuristics on the SRASearch.

E.5 $BC/WC = 0.5$

E.5.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

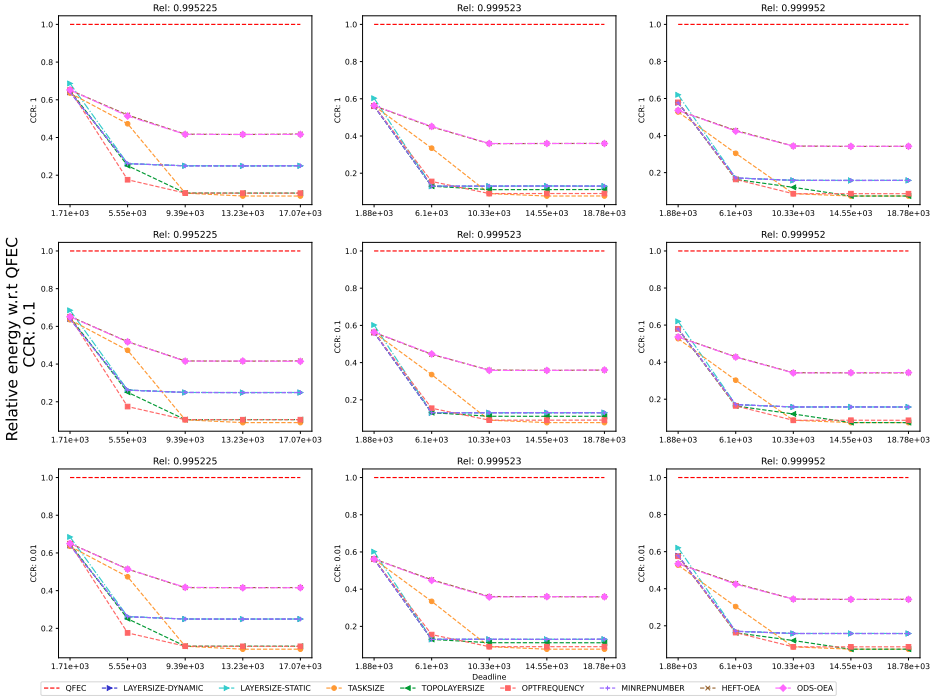

Figure 602: Performance of the different heuristics on the BWA workflow.

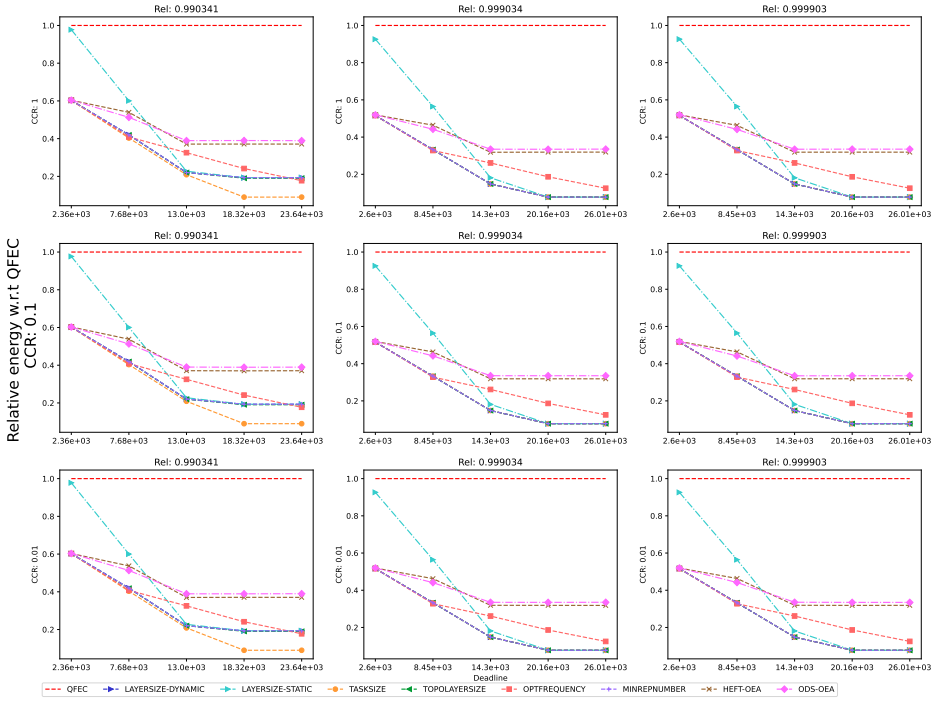

Figure 603: Performance of the different heuristics on the Cholesky workflow.

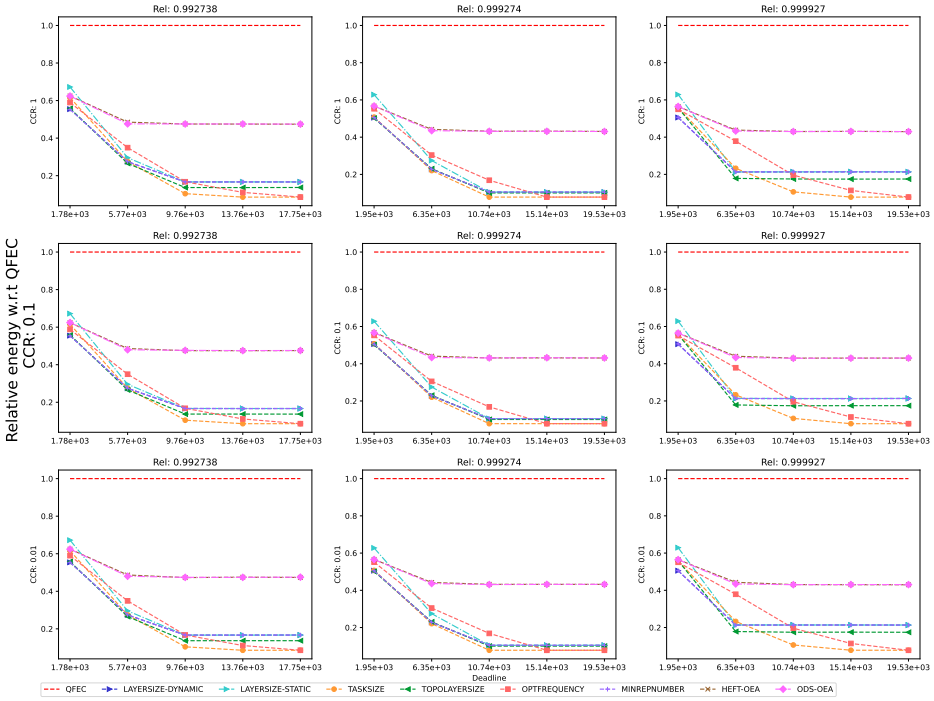

Figure 604: Performance of the different heuristics on the Cycles workflow.

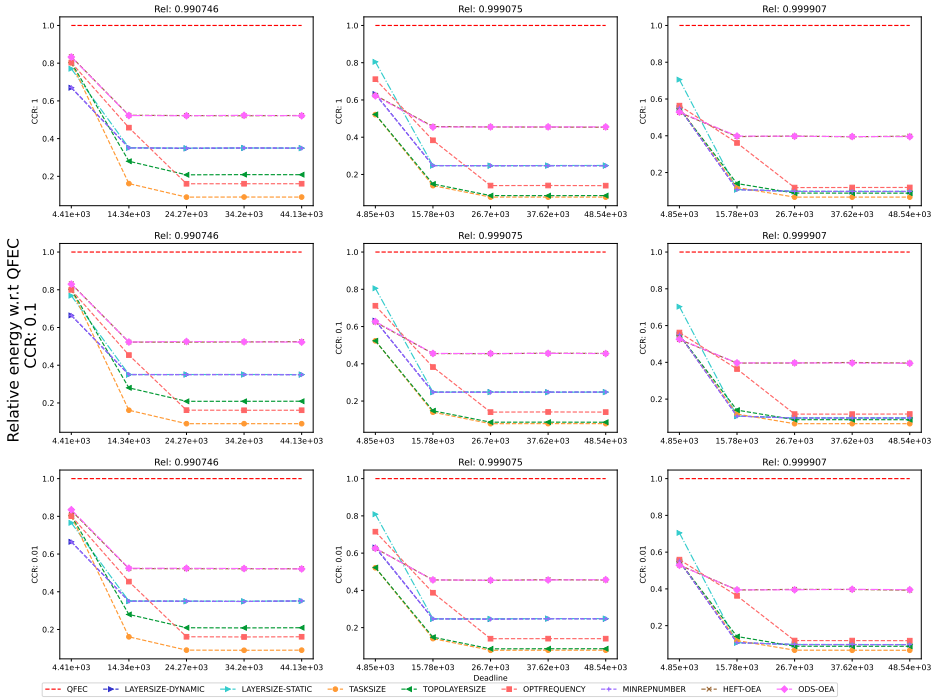

Figure 605: Performance of the different heuristics on the Epigenomics workflow.

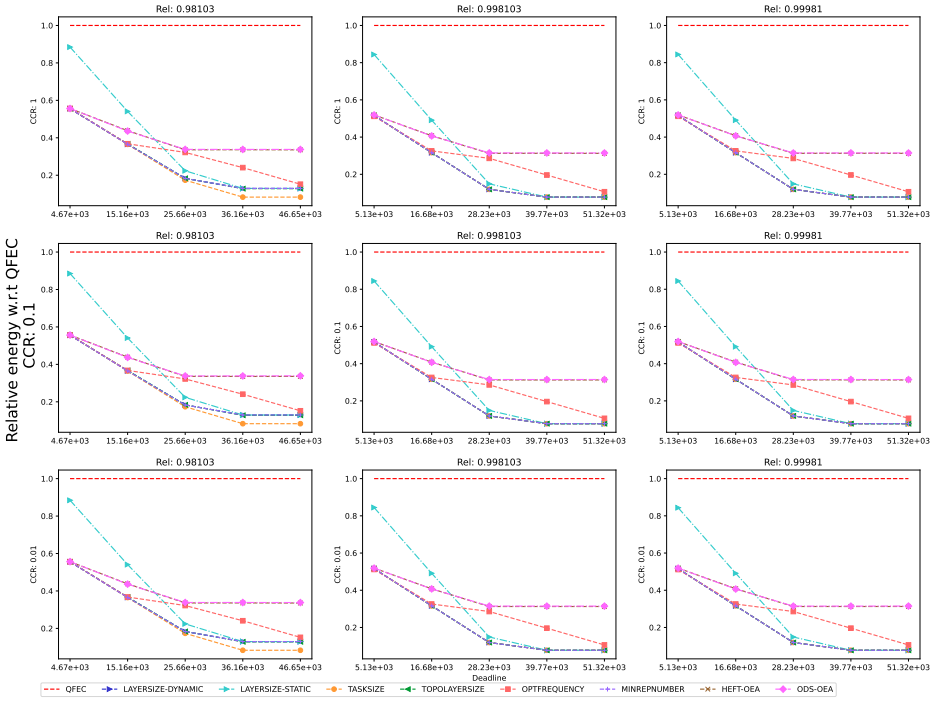

Figure 606: Performance of the different heuristics on the LU workflow.

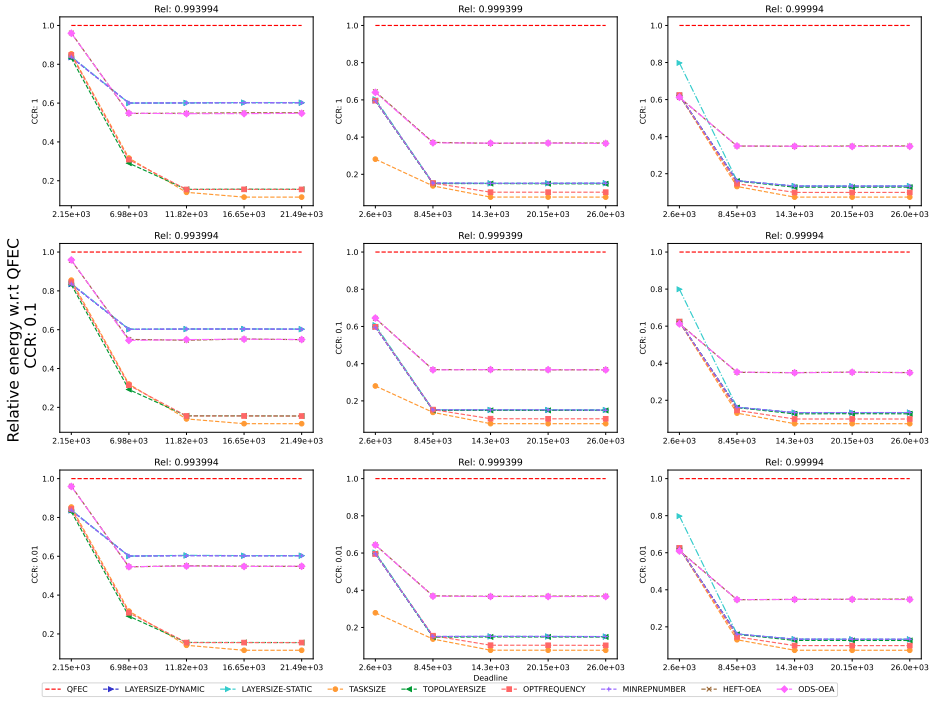

Figure 607: Performance of the different heuristics on the Montage workflow.

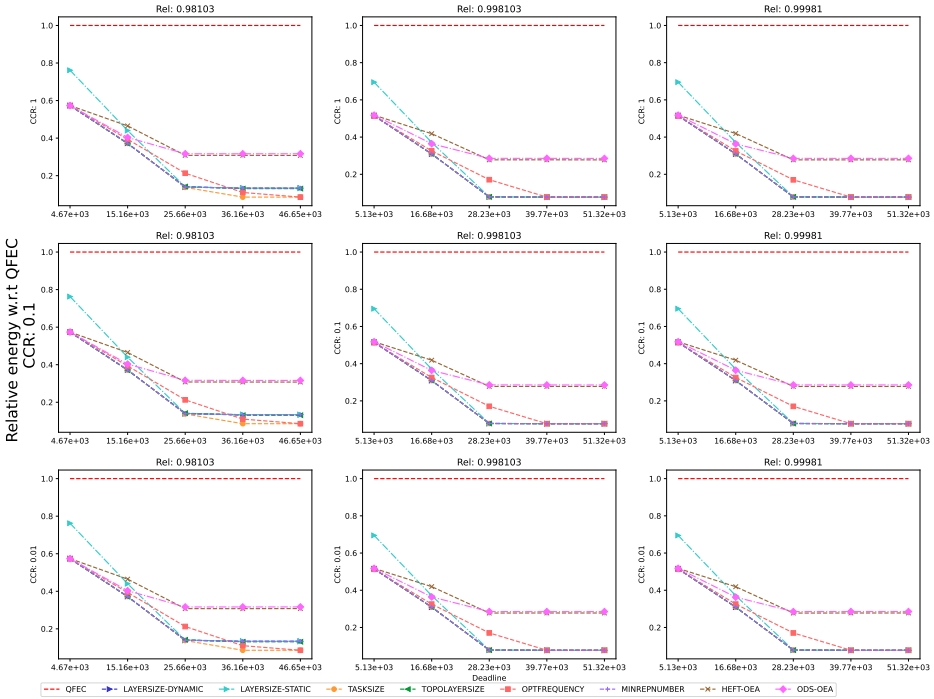

Figure 608: Performance of the different heuristics on the QR workflow.

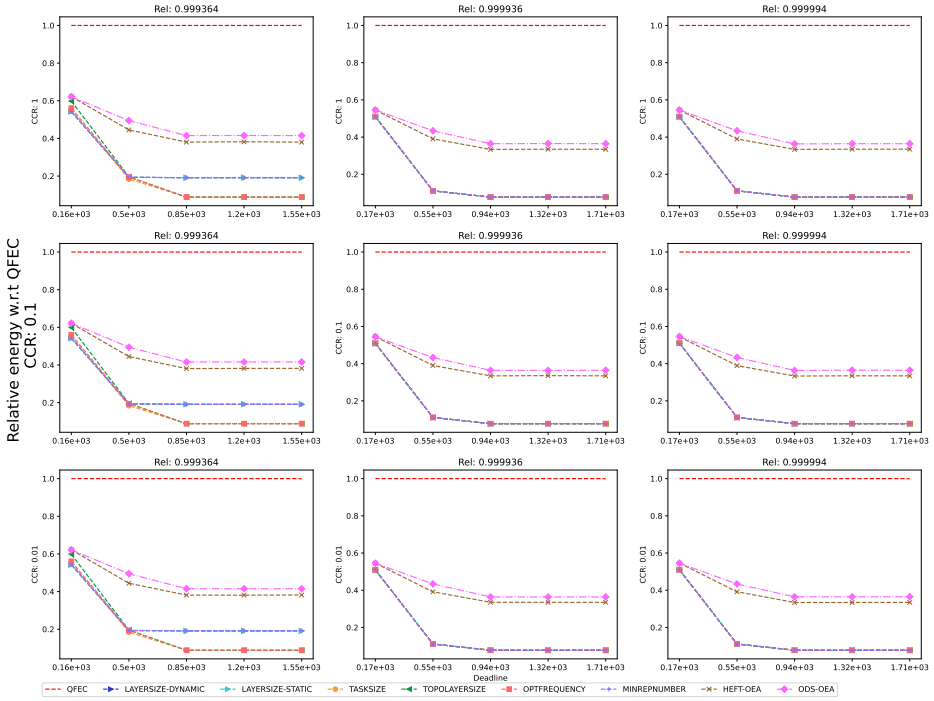

Figure 609: Performance of the different heuristics on the Seismology workflow.

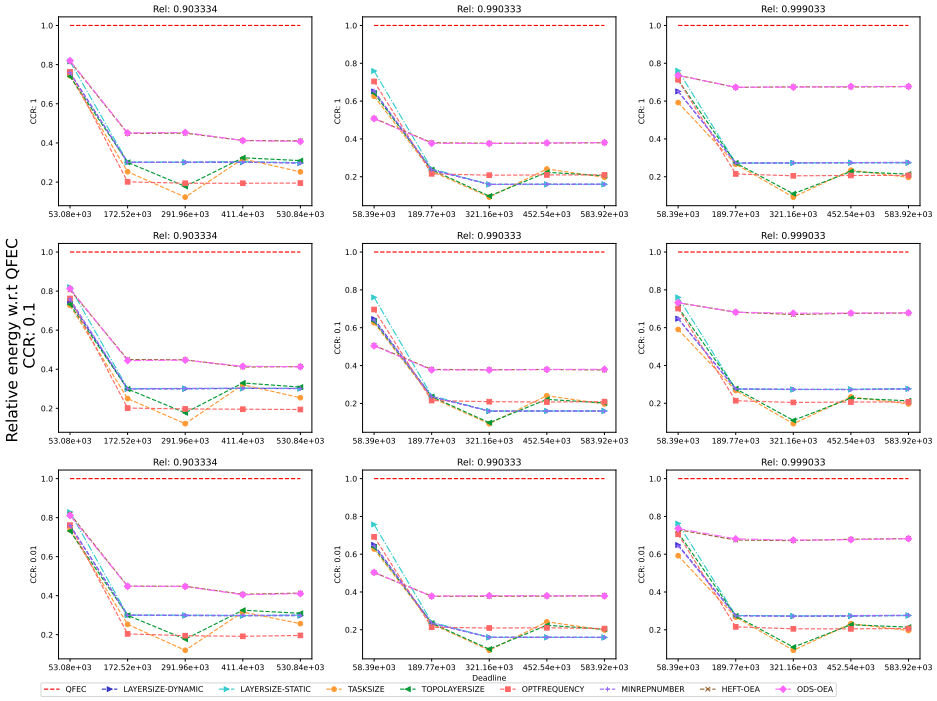

Figure 610: Performance of the different heuristics on the SoyKB workflow.

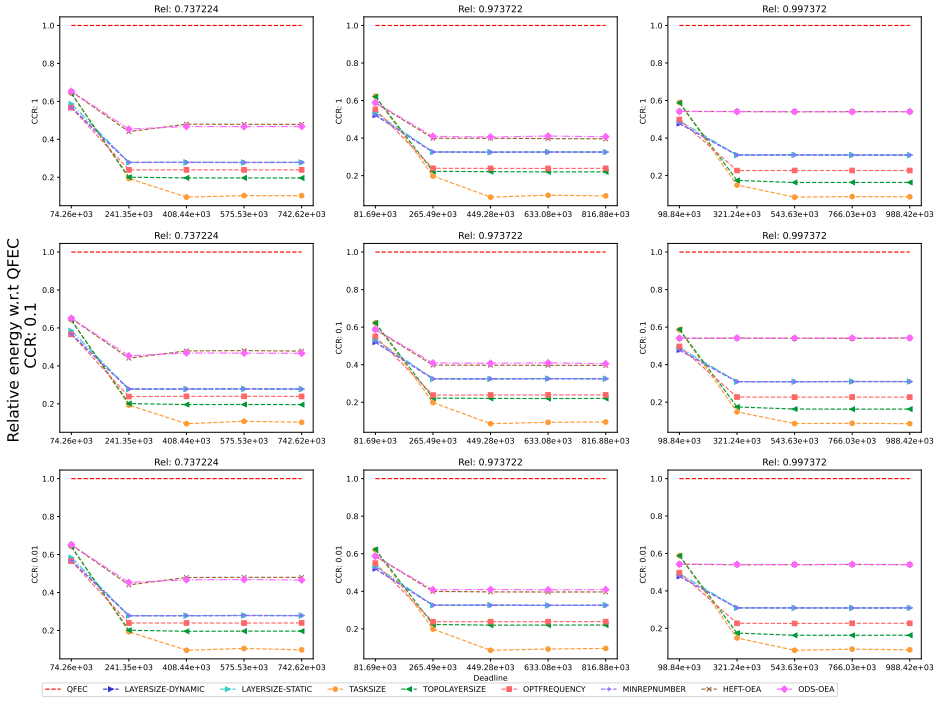

Figure 611: Performance of the different heuristics on the SRASearch.

E.5.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

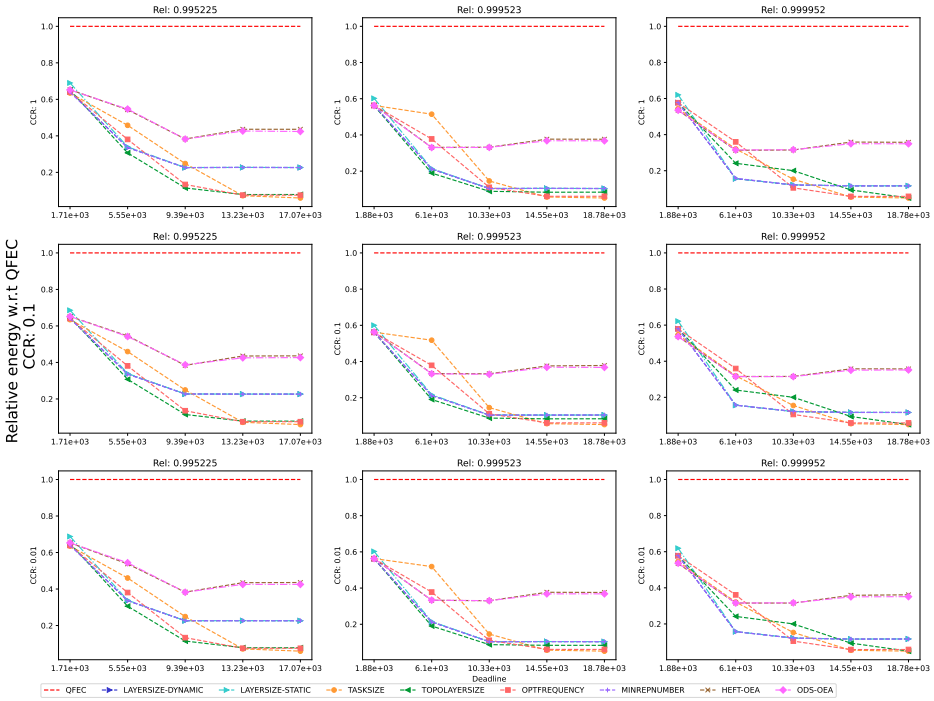

Figure 612: Performance of the different heuristics on the BWA workflow.

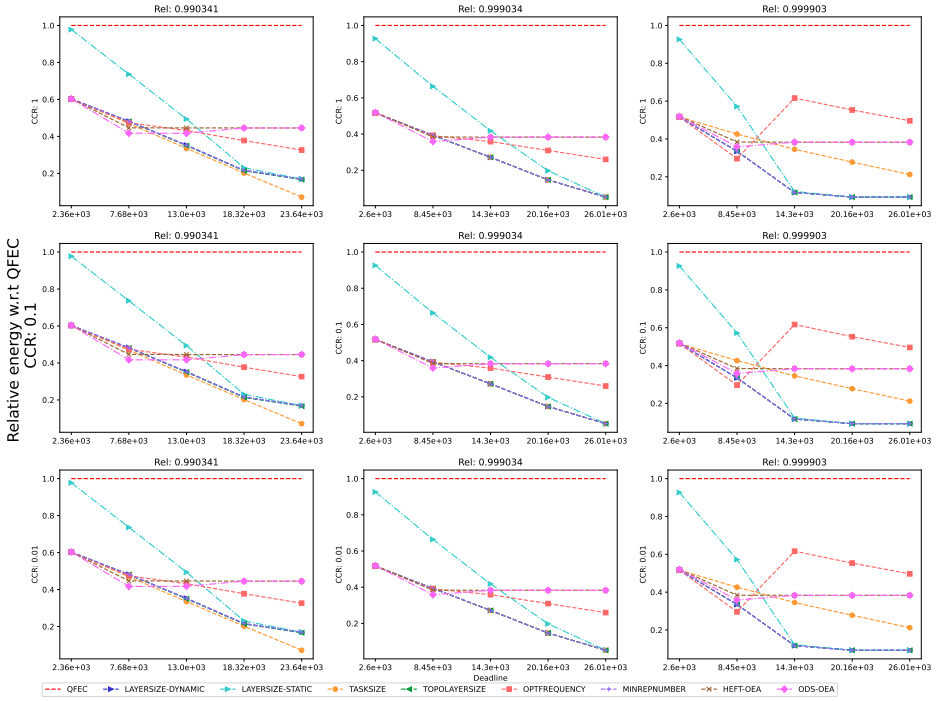

Figure 613: Performance of the different heuristics on the Cholesky workflow.

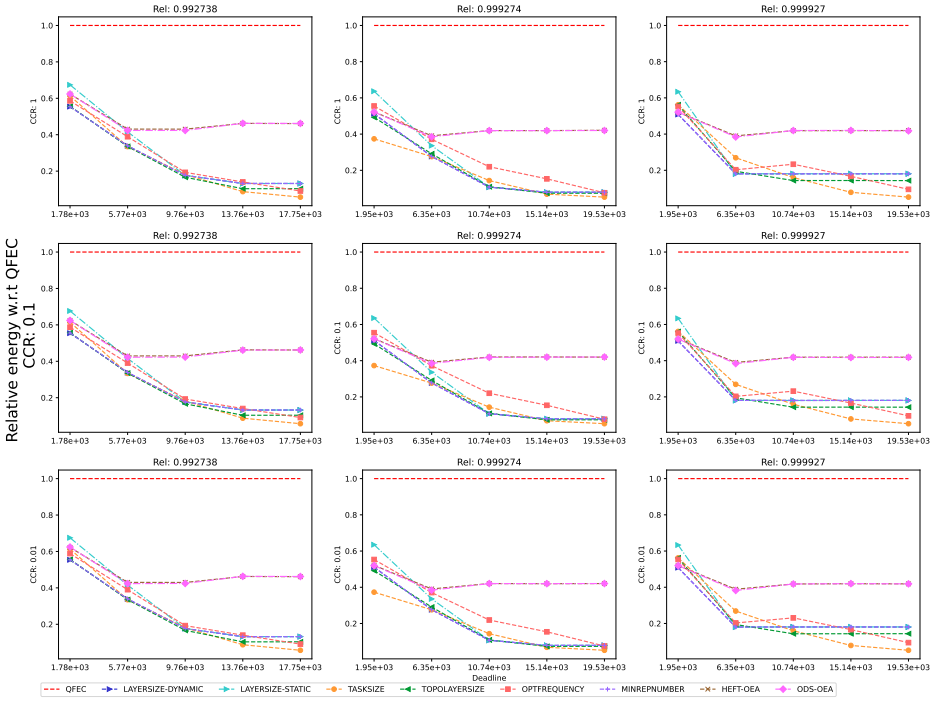

Figure 614: Performance of the different heuristics on the Cycles workflow.

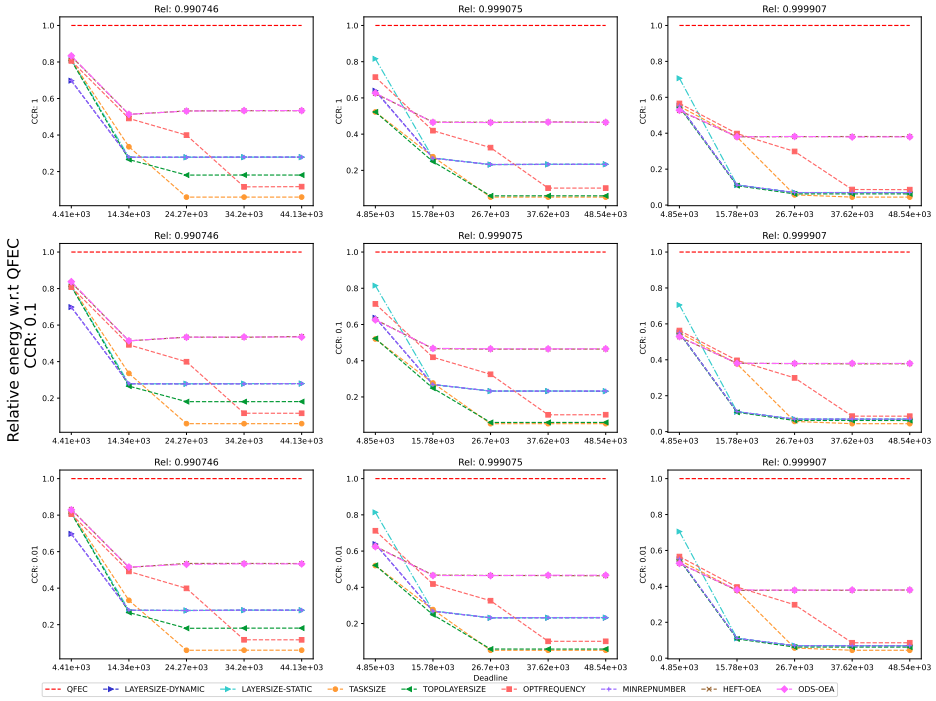

Figure 615: Performance of the different heuristics on the Epigenomics workflow.

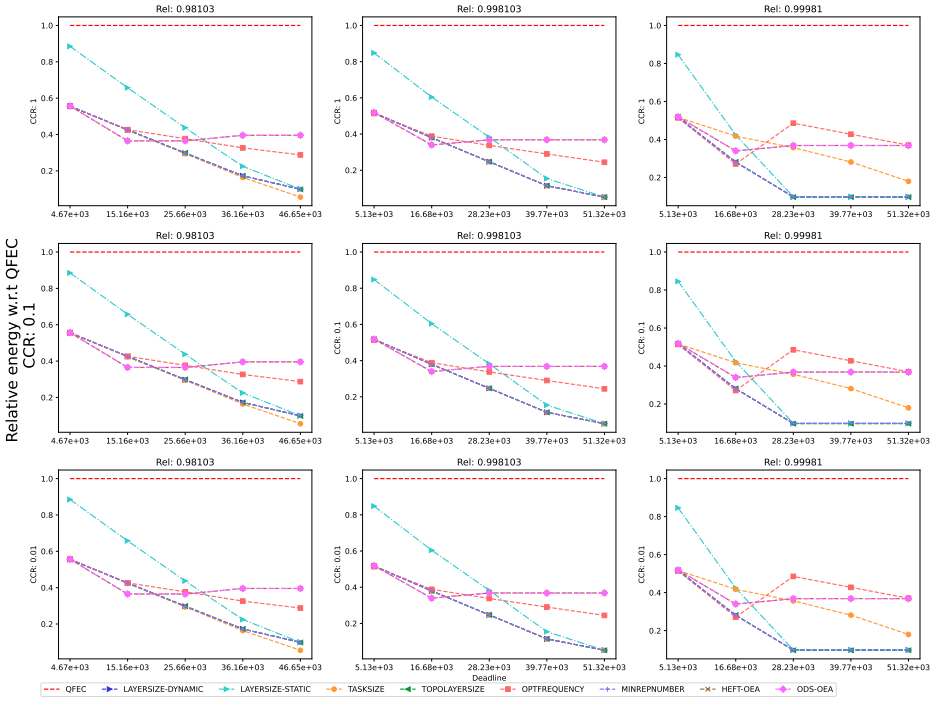

Figure 616: Performance of the different heuristics on the LU workflow.

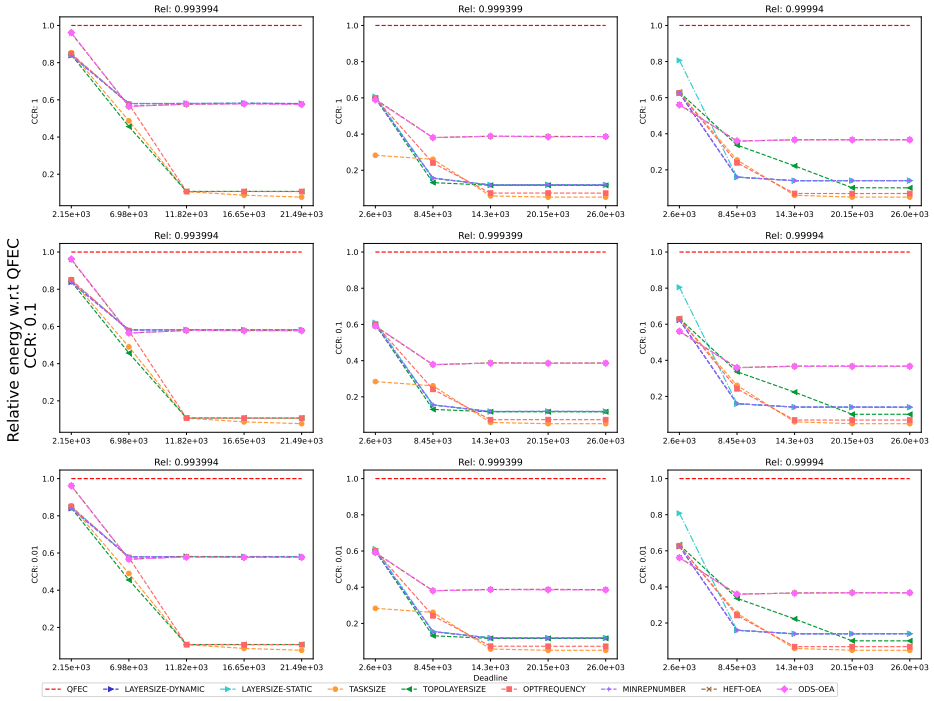

Figure 617: Performance of the different heuristics on the Montage workflow.

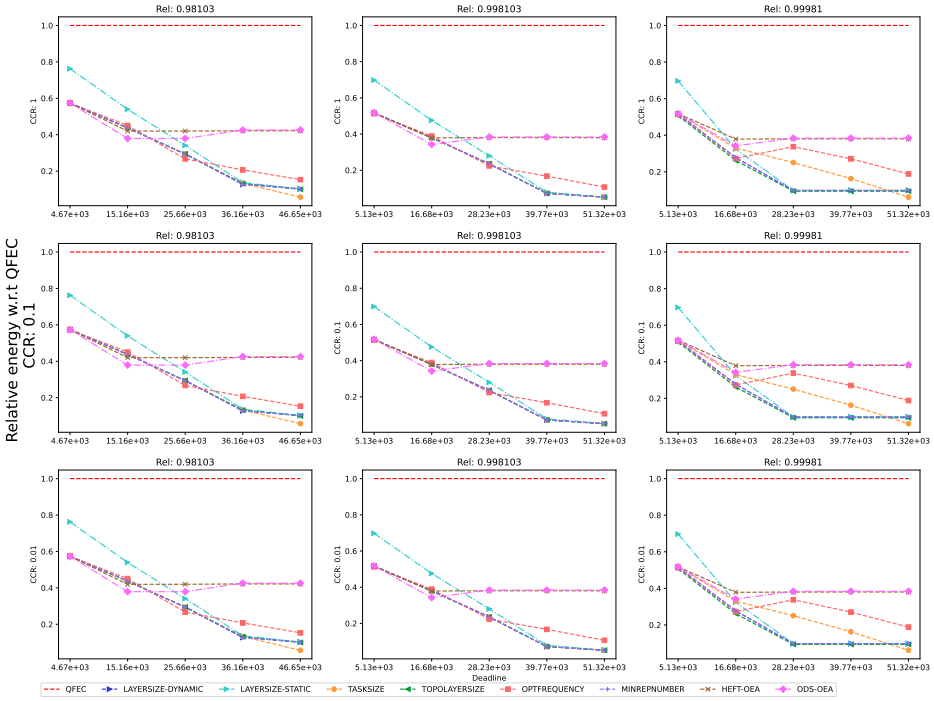

Figure 618: Performance of the different heuristics on the QR workflow.

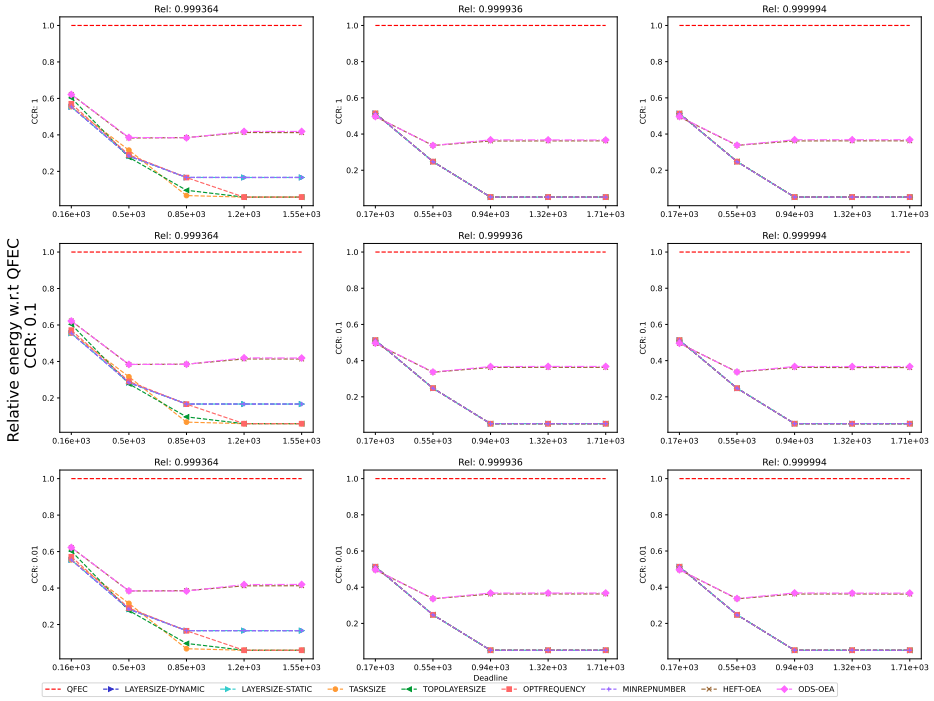

Figure 619: Performance of the different heuristics on the Seismology workflow.

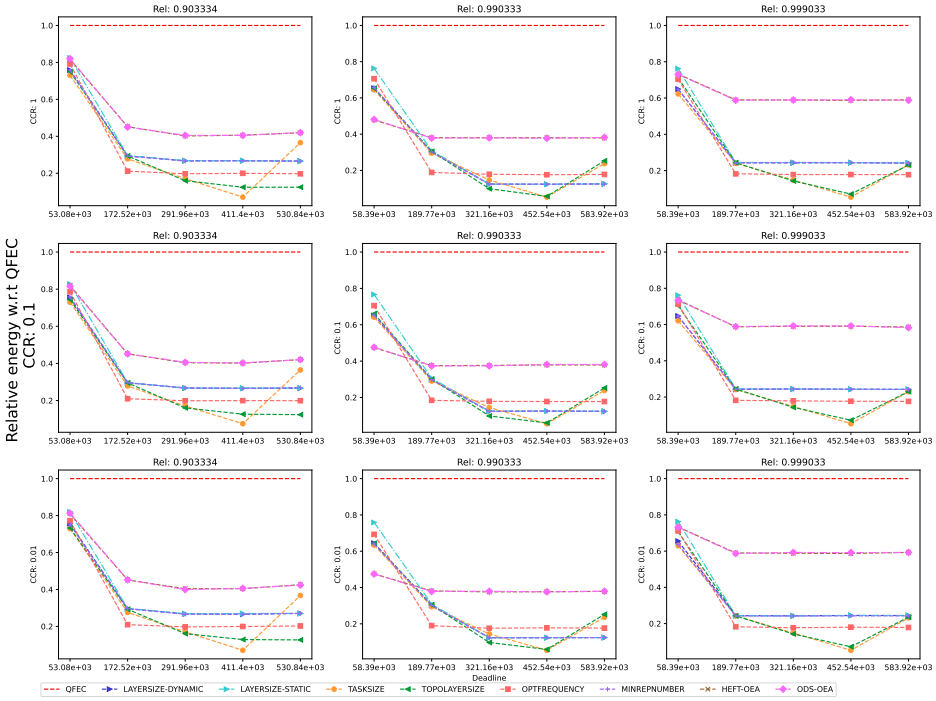

Figure 620: Performance of the different heuristics on the SoyKB workflow.

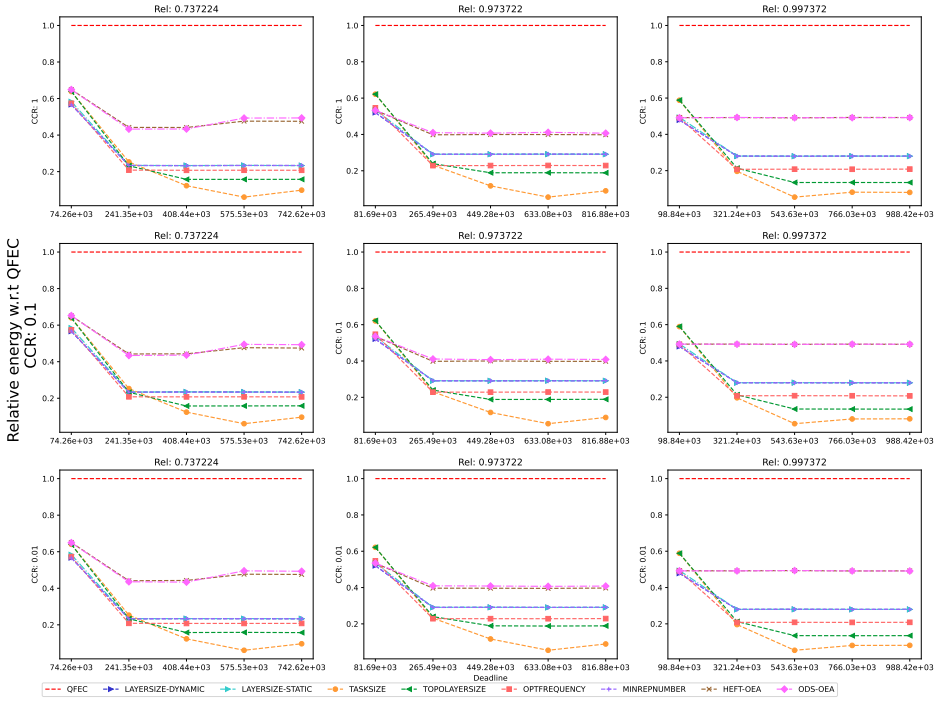

Figure 621: Performance of the different heuristics on the SRASearch.

E.5.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

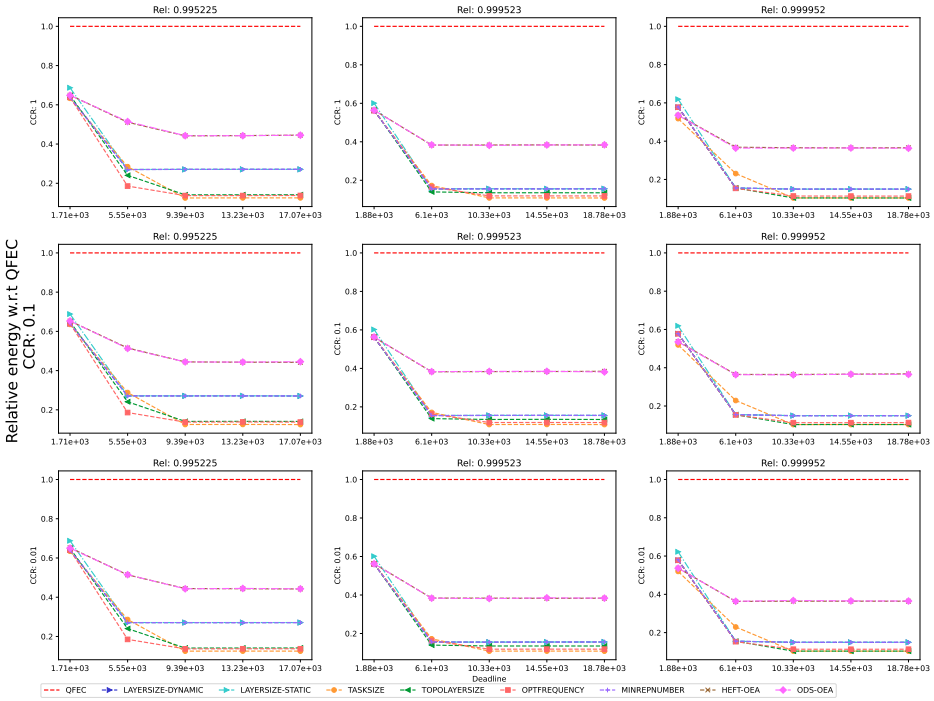

Figure 622: Performance of the different heuristics on the BWA workflow.

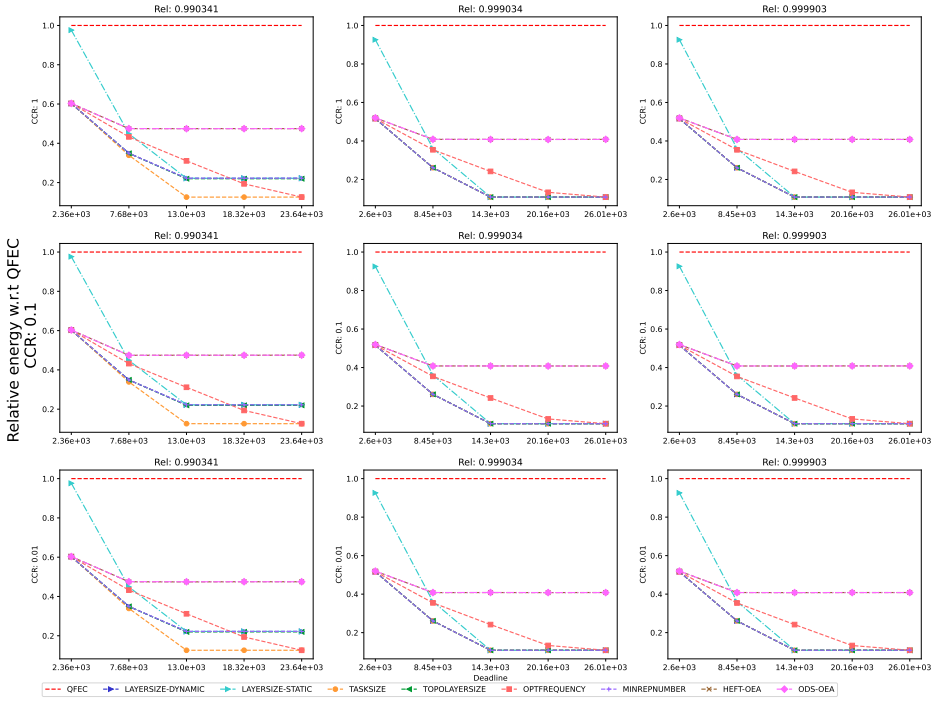

Figure 623: Performance of the different heuristics on the Cholesky workflow.

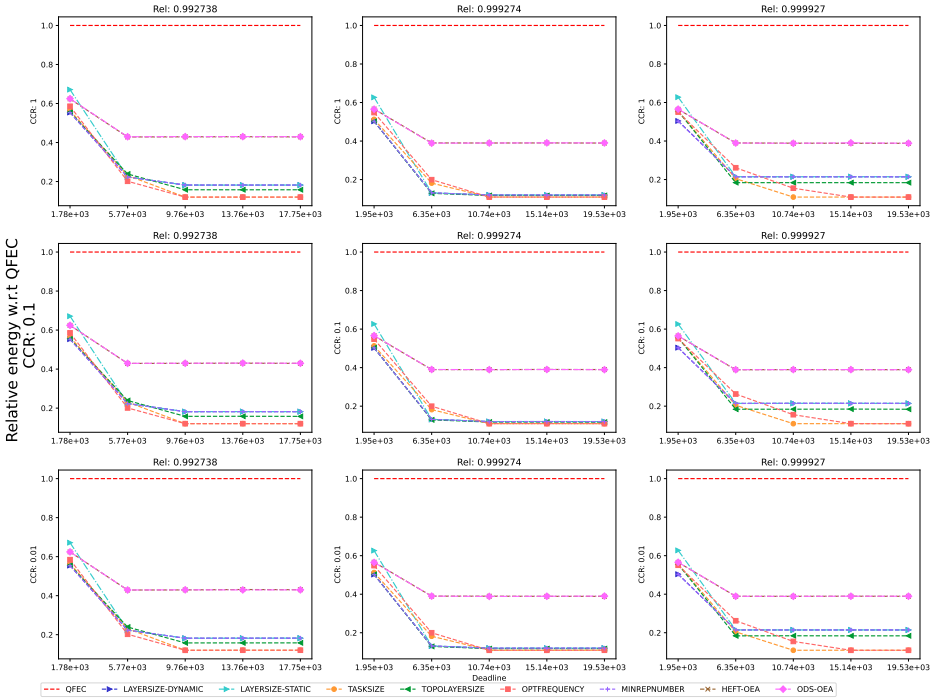

Figure 624: Performance of the different heuristics on the Cycles workflow.

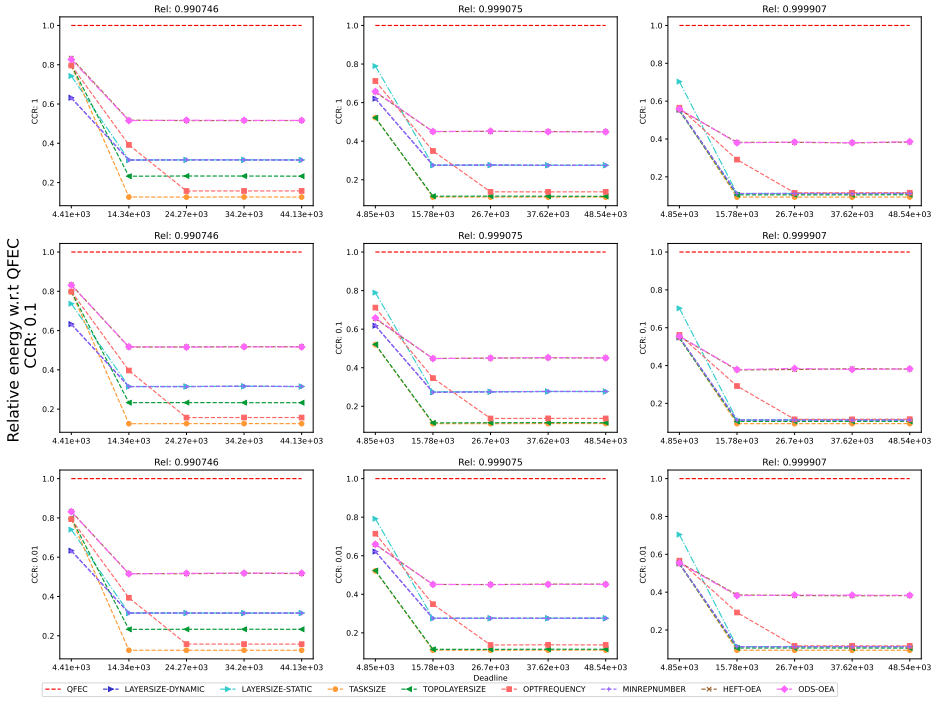

Figure 625: Performance of the different heuristics on the Epigenomics workflow.

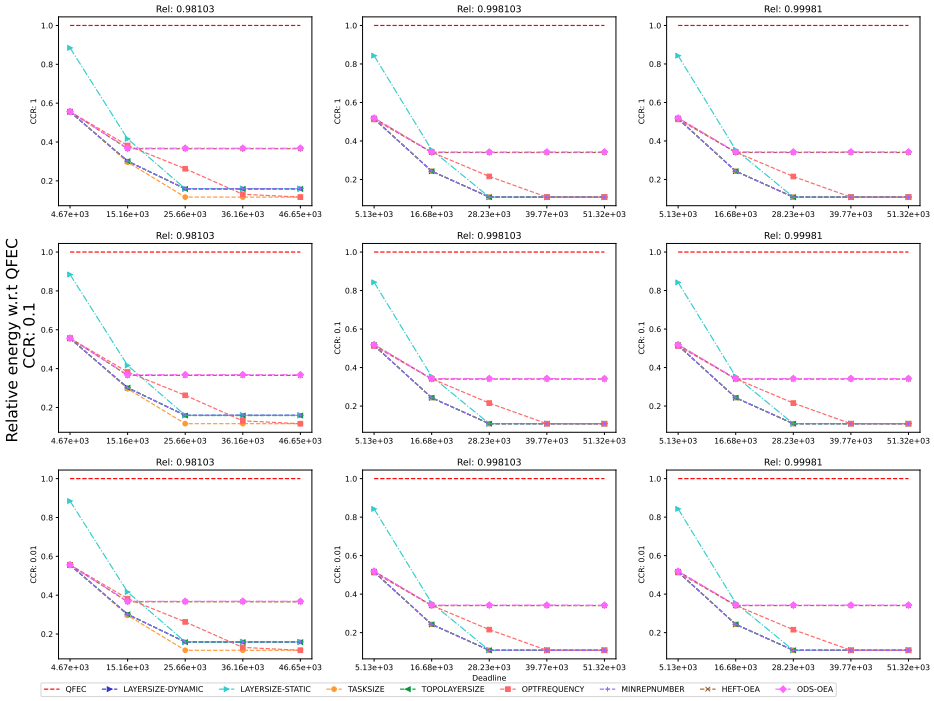

Figure 626: Performance of the different heuristics on the LU workflow.

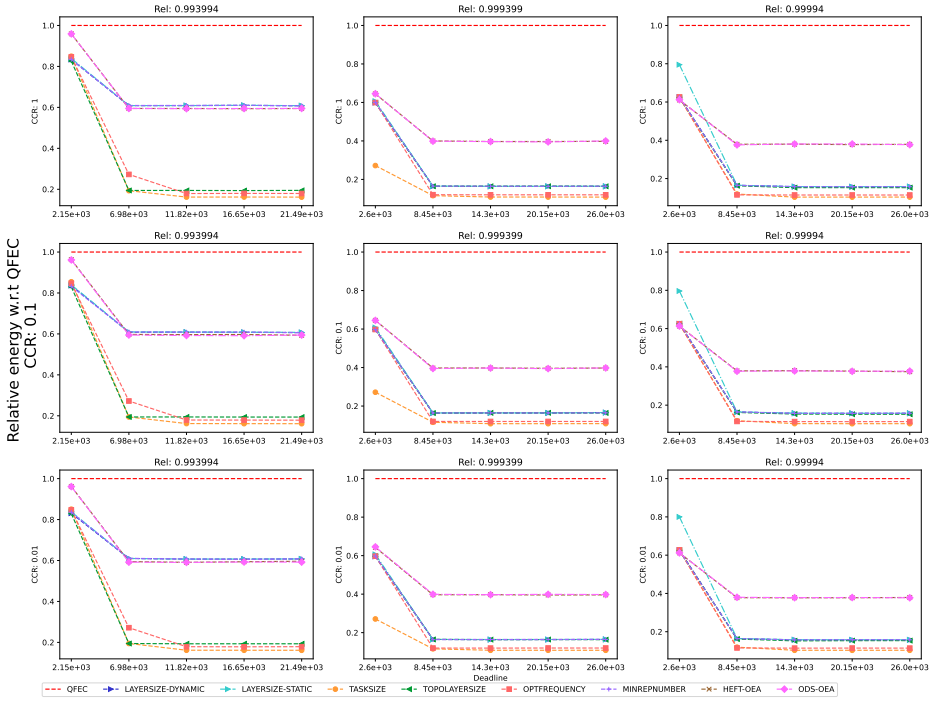

Figure 627: Performance of the different heuristics on the Montage workflow.

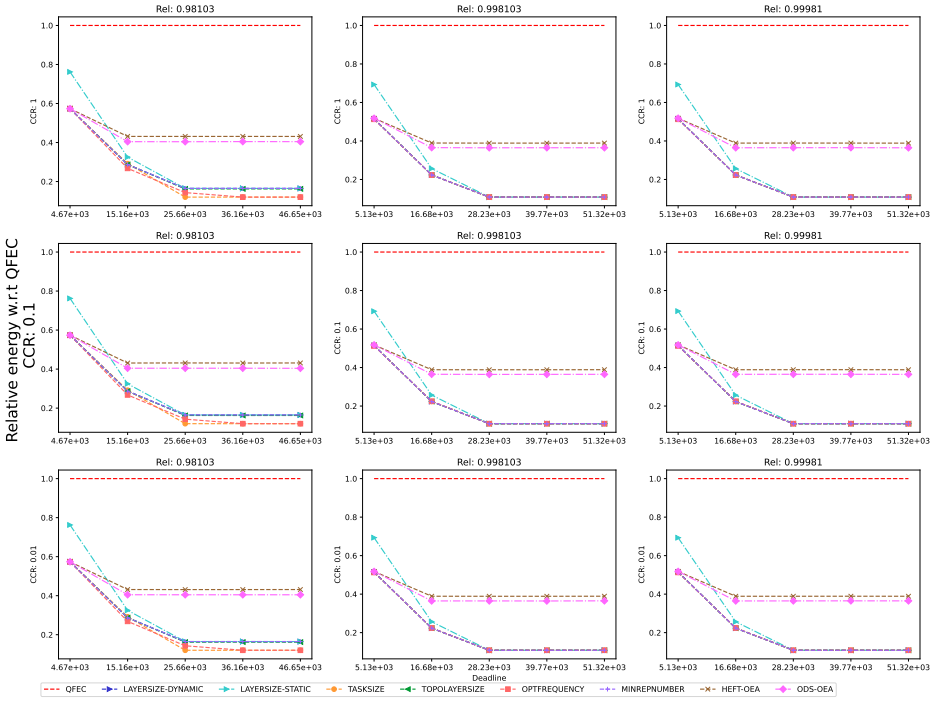

Figure 628: Performance of the different heuristics on the QR workflow.

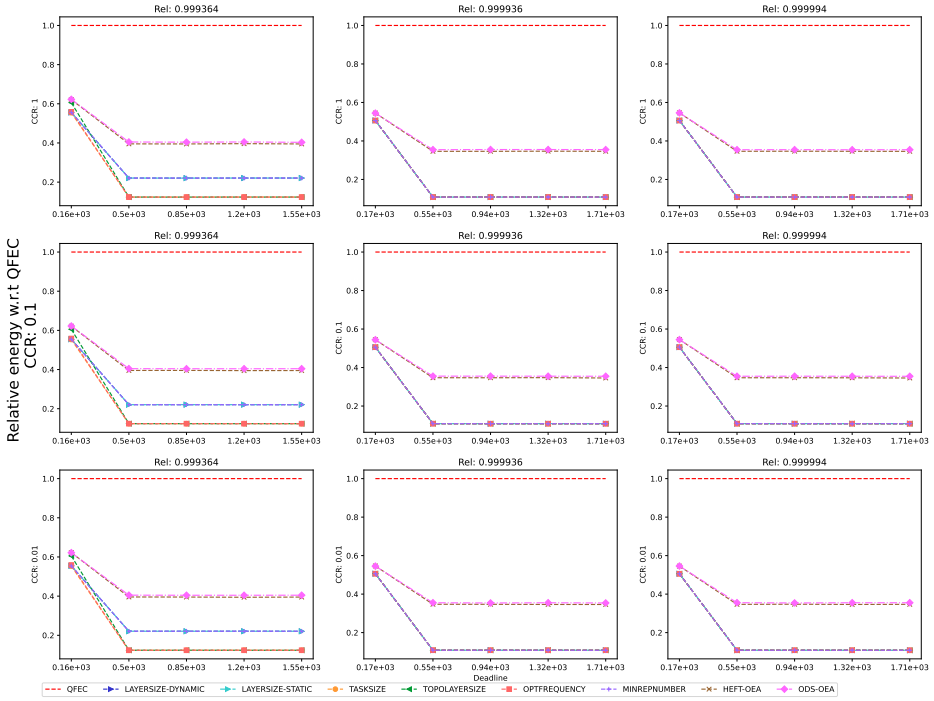

Figure 629: Performance of the different heuristics on the Seismology workflow.

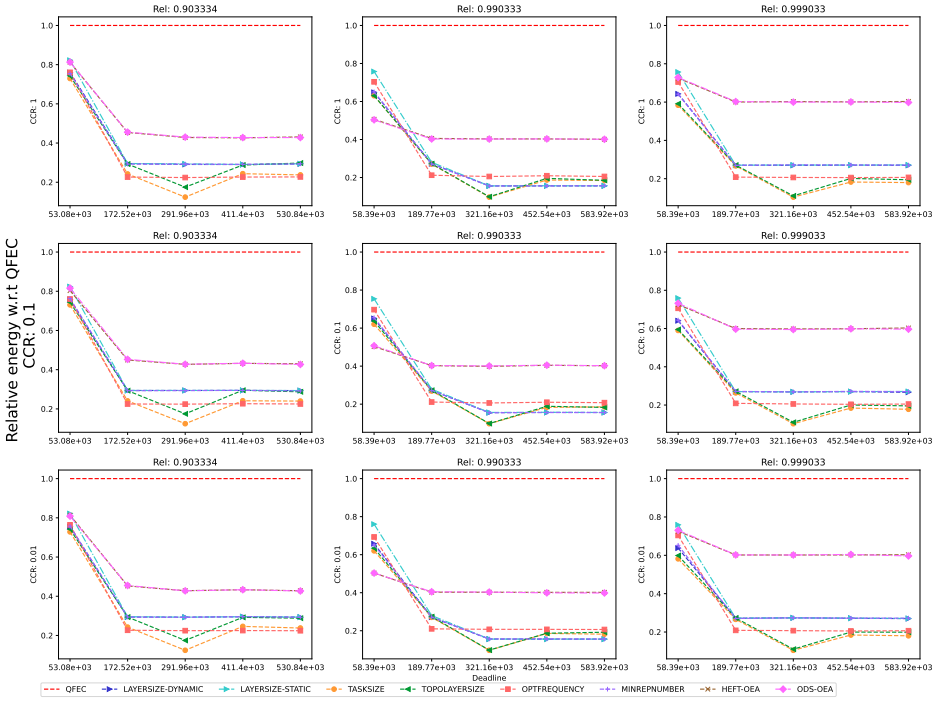

Figure 630: Performance of the different heuristics on the SoyKB workflow.

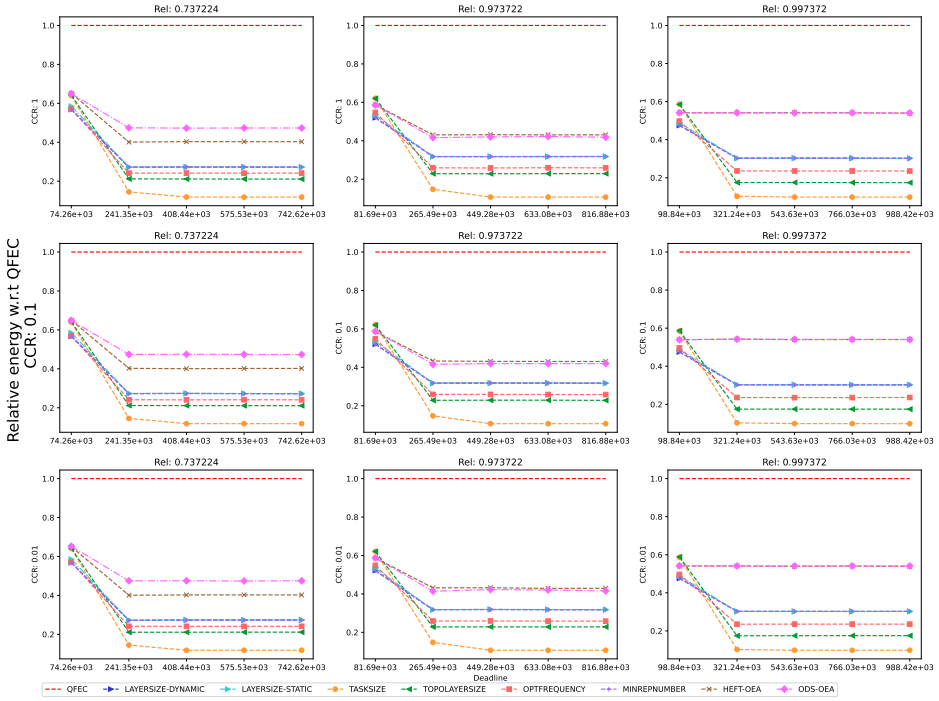

Figure 631: Performance of the different heuristics on the SRASearch.

E.5.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

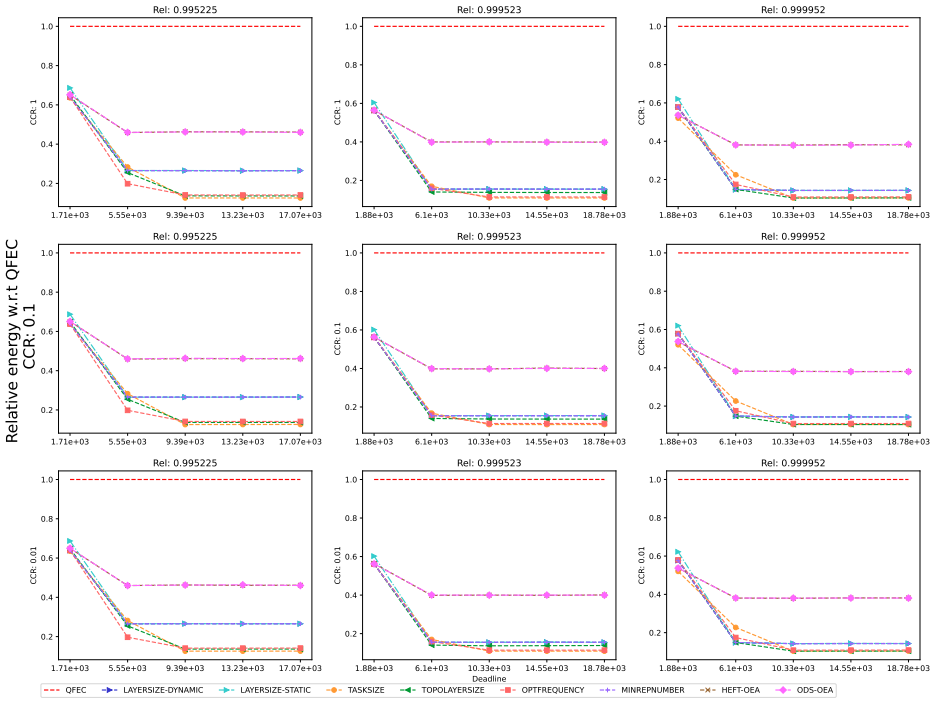

Figure 632: Performance of the different heuristics on the BWA workflow.

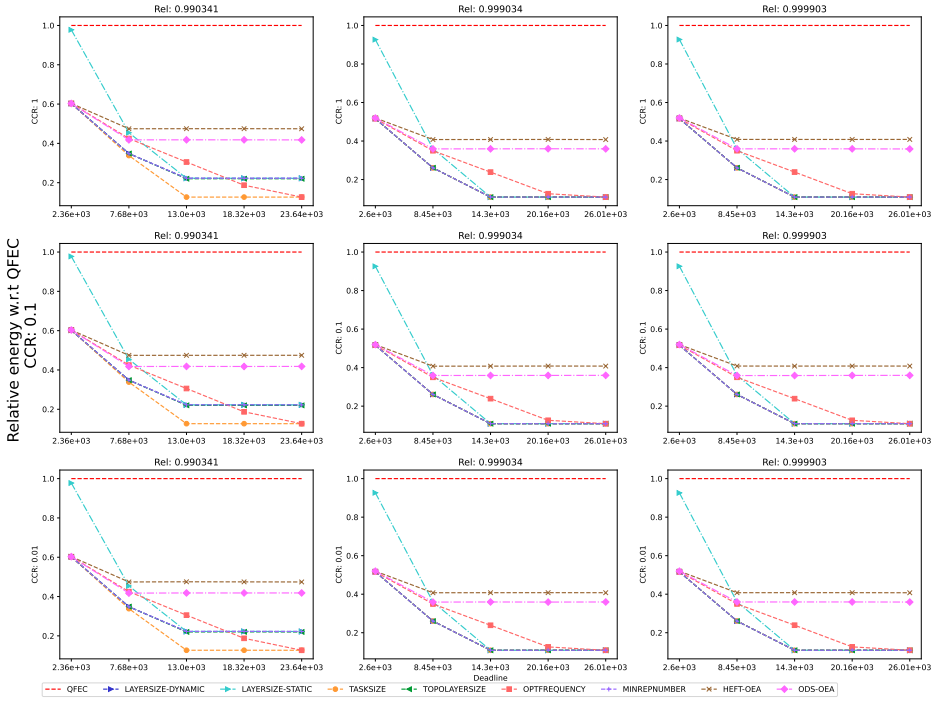

Figure 633: Performance of the different heuristics on the Cholesky workflow.

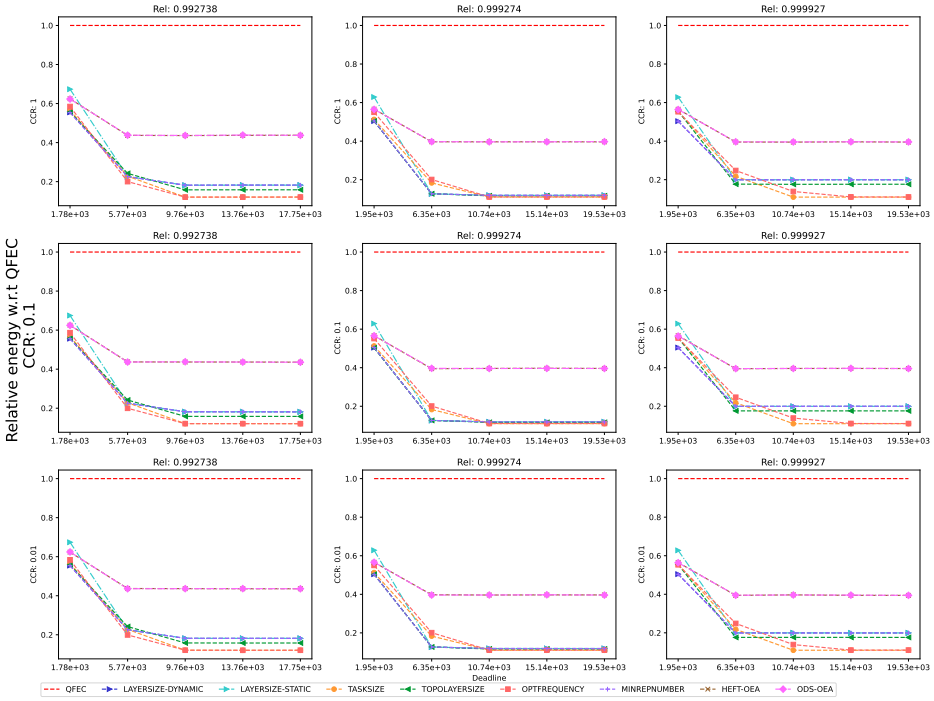

Figure 634: Performance of the different heuristics on the Cycles workflow.

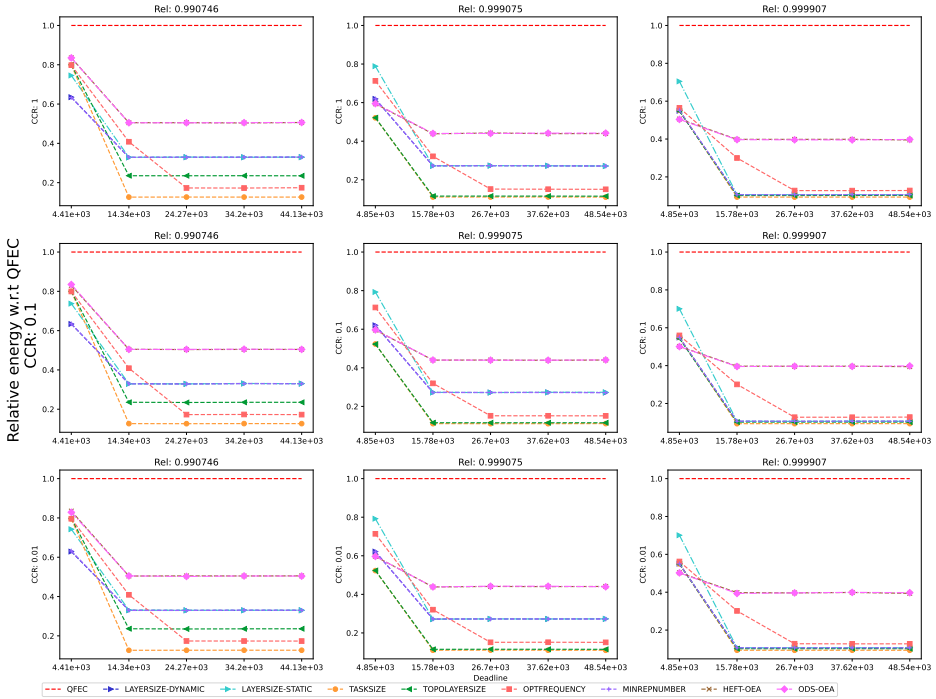

Figure 635: Performance of the different heuristics on the Epigenomics workflow.

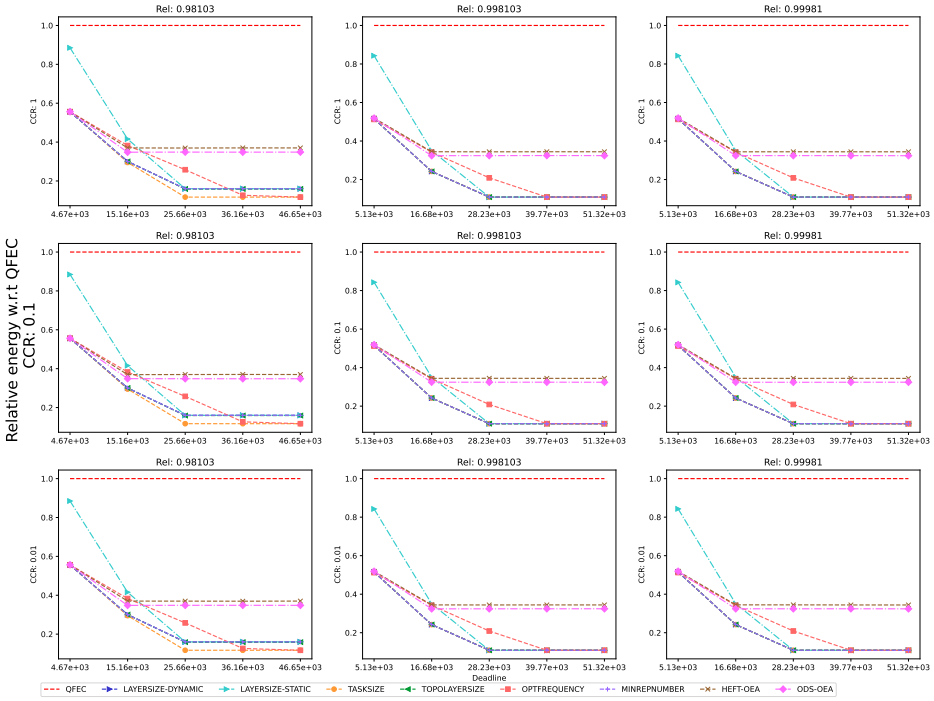

Figure 636: Performance of the different heuristics on the LU workflow.

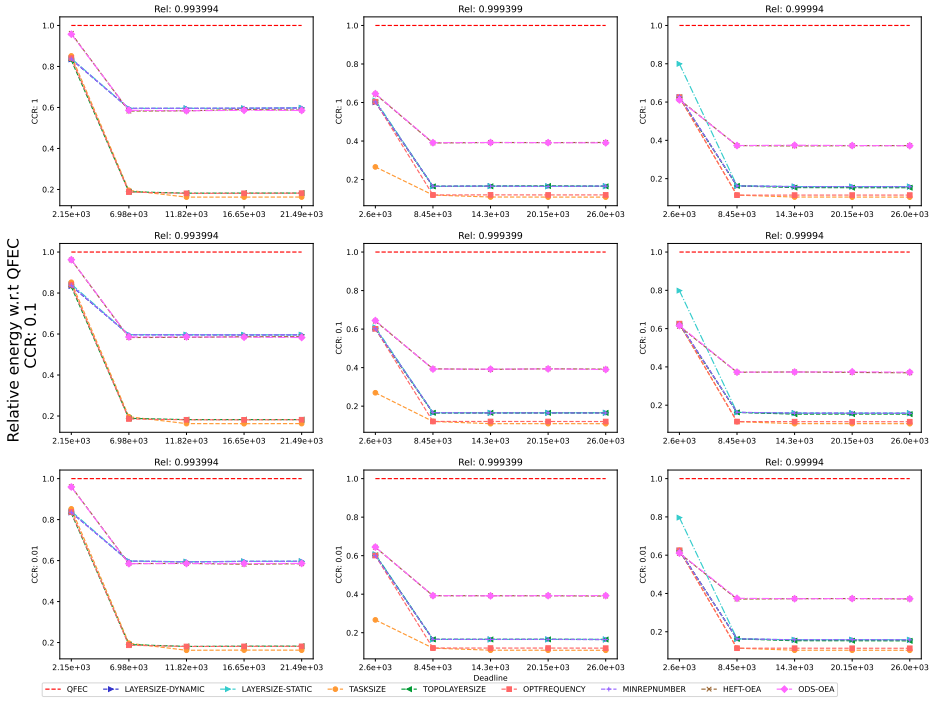

Figure 637: Performance of the different heuristics on the Montage workflow.

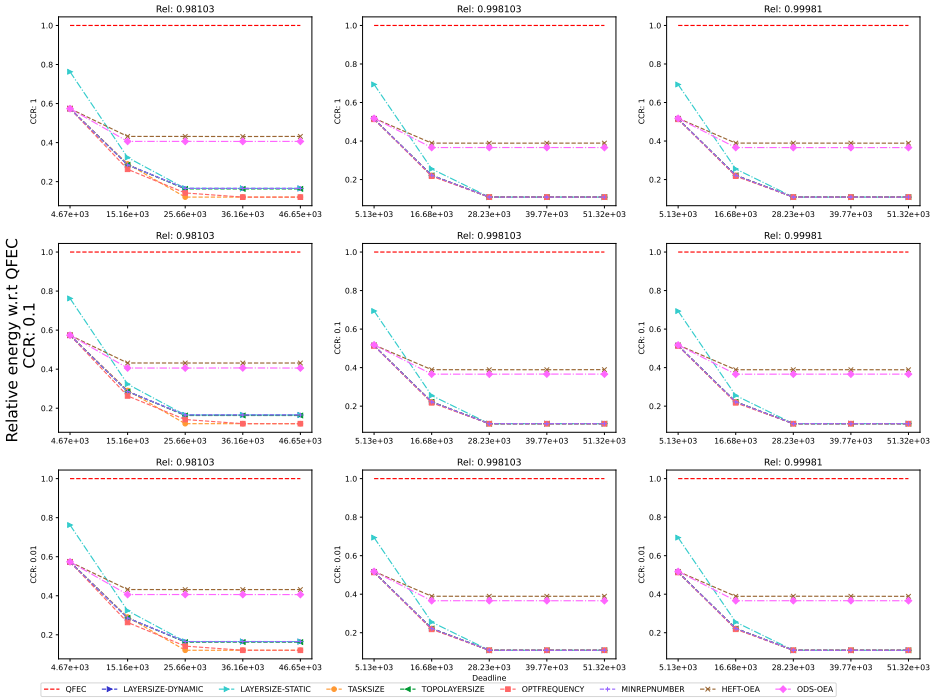

Figure 638: Performance of the different heuristics on the QR workflow.

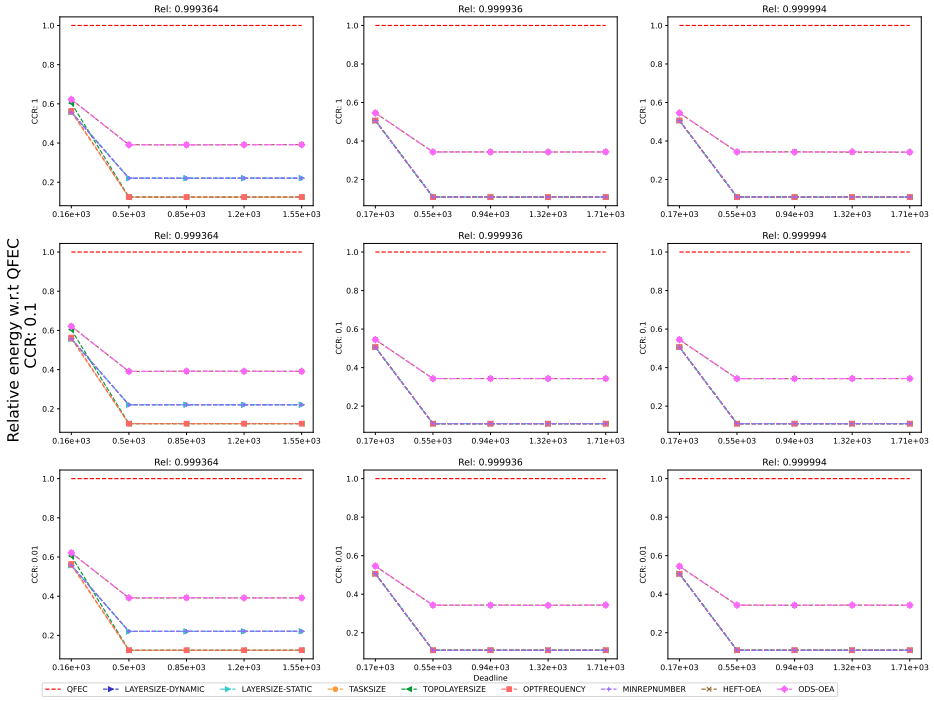

Figure 639: Performance of the different heuristics on the Seismology workflow.

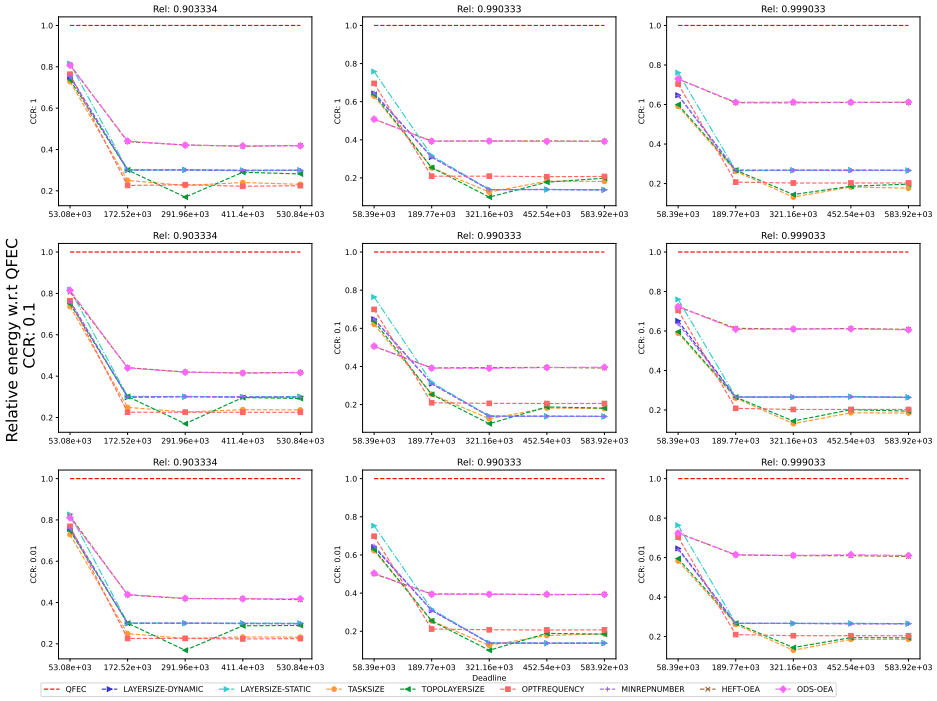

Figure 640: Performance of the different heuristics on the SoyKB workflow.

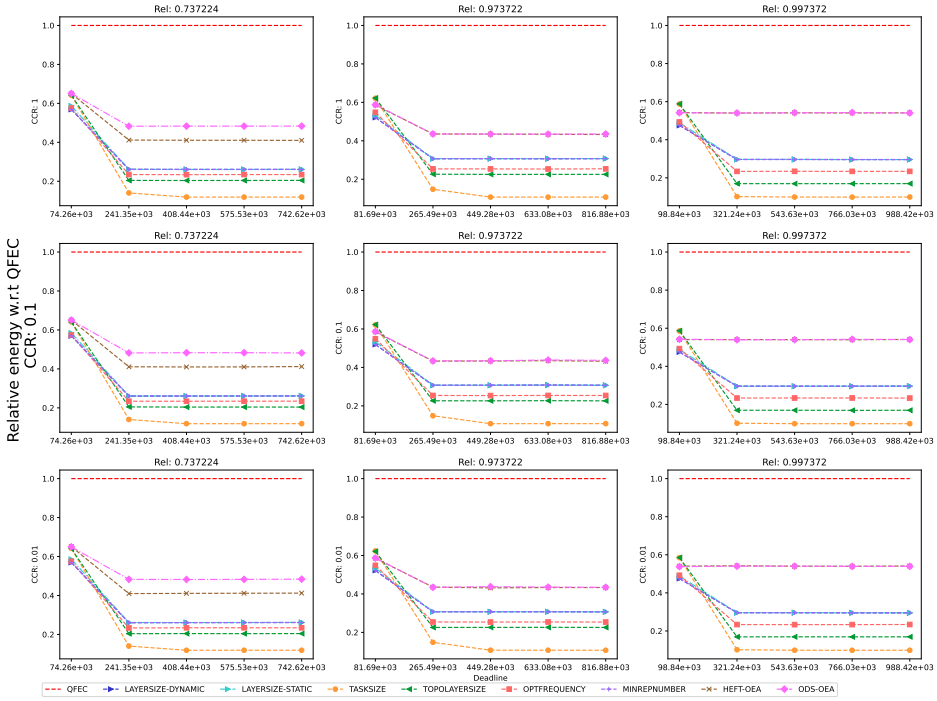

Figure 641: Performance of the different heuristics on the SRASearch.

E.6 $BC/WC = 0.6$

E.6.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

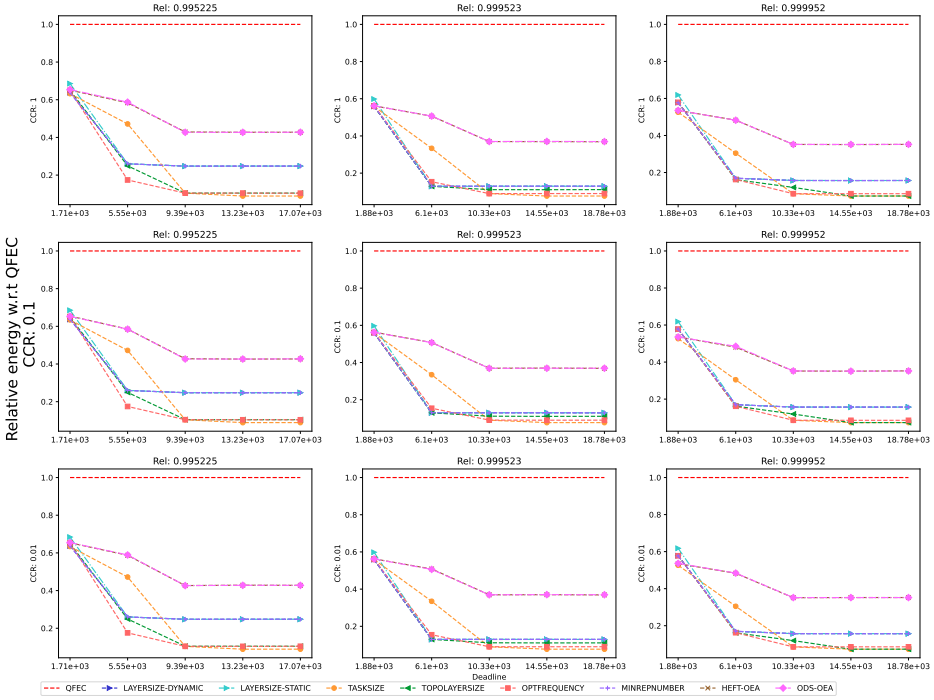

Figure 642: Performance of the different heuristics on the BWA workflow.

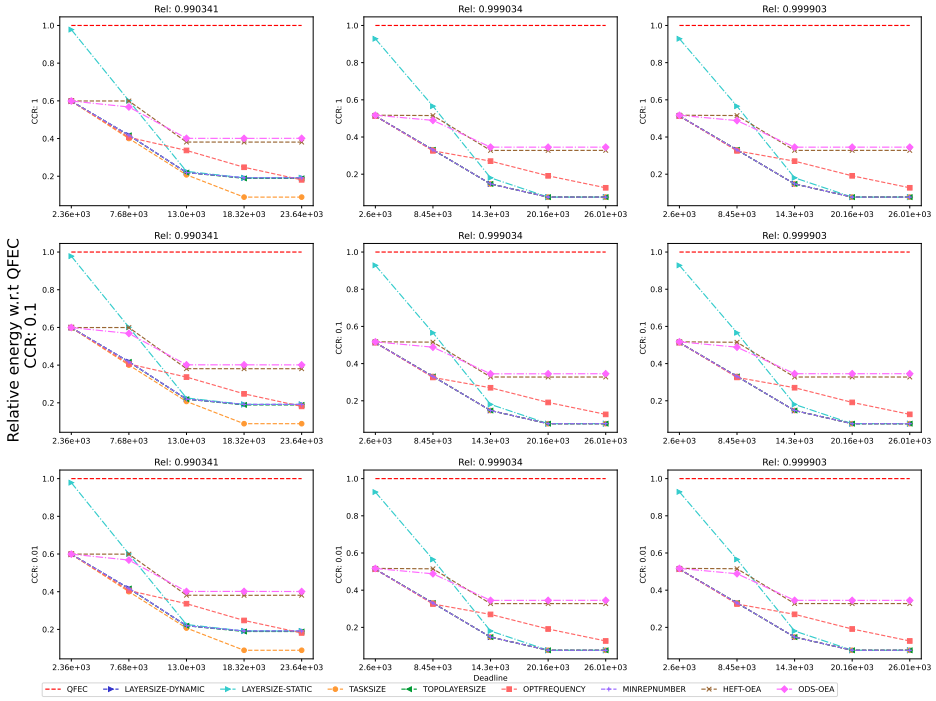

Figure 643: Performance of the different heuristics on the Cholesky workflow.

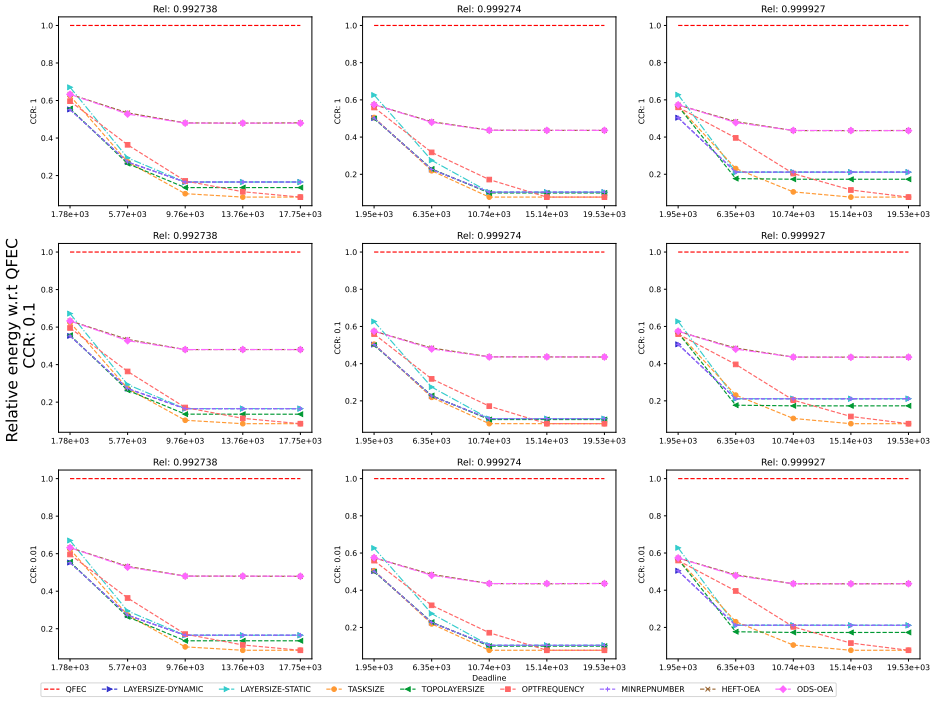

Figure 644: Performance of the different heuristics on the Cycles workflow.

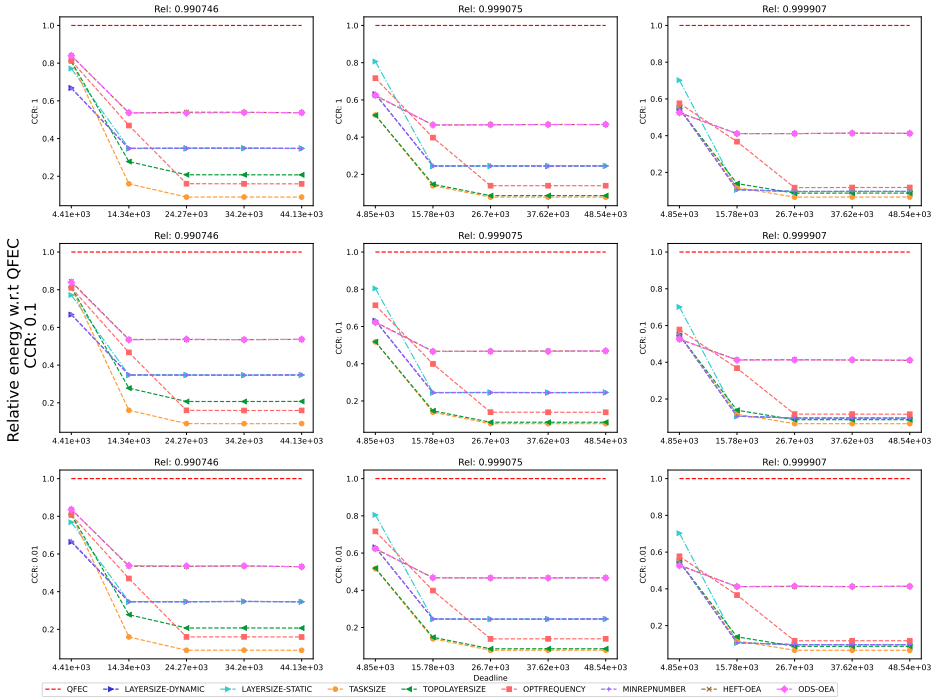

Figure 645: Performance of the different heuristics on the Epigenomics workflow.

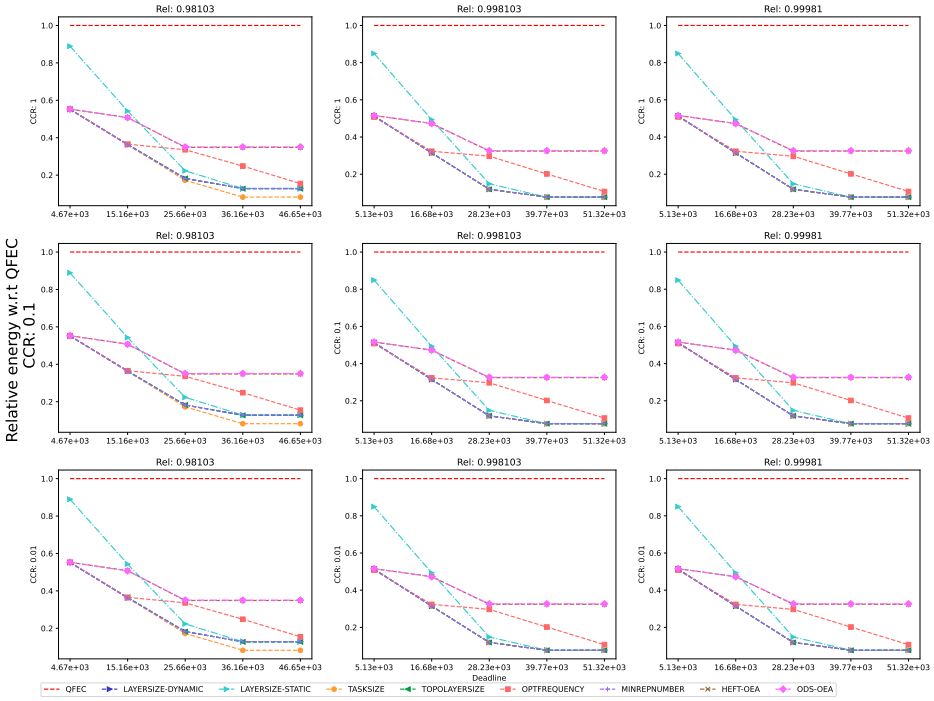

Figure 646: Performance of the different heuristics on the LU workflow.

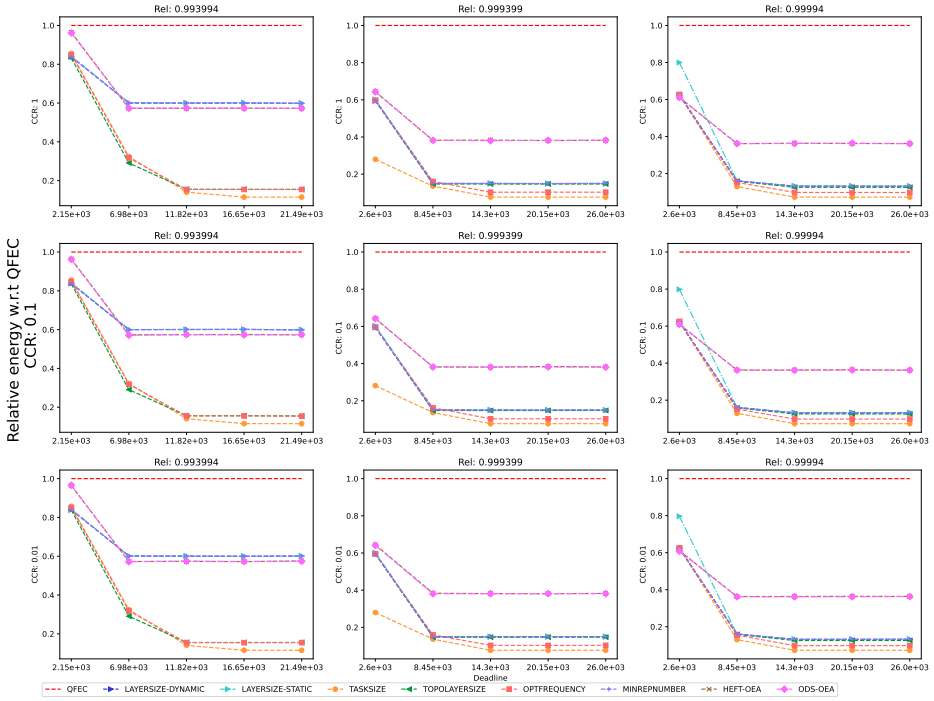

Figure 647: Performance of the different heuristics on the Montage workflow.

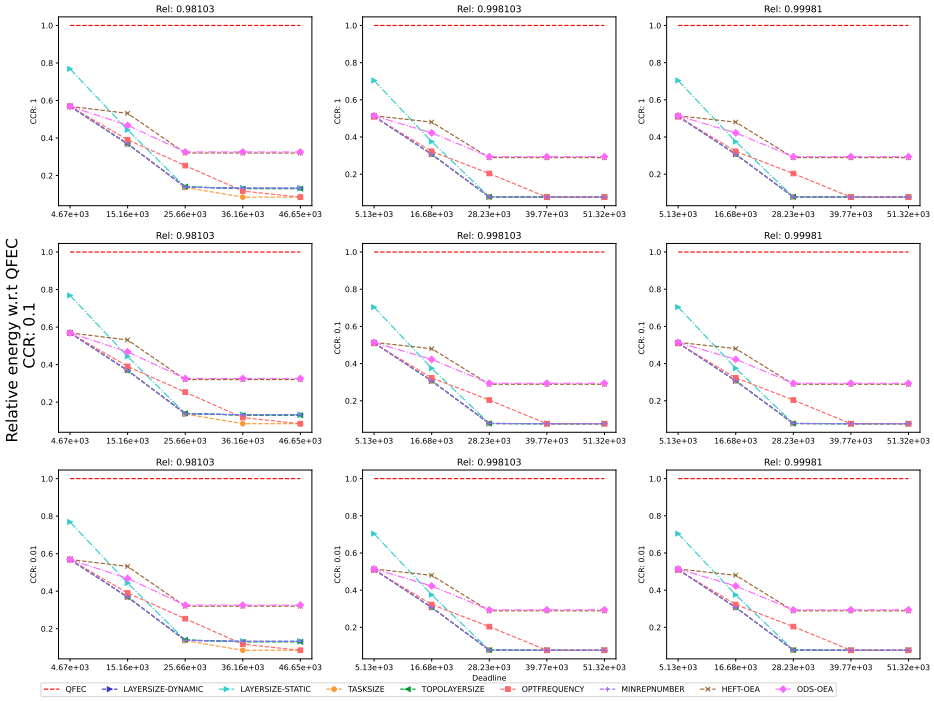

Figure 648: Performance of the different heuristics on the QR workflow.

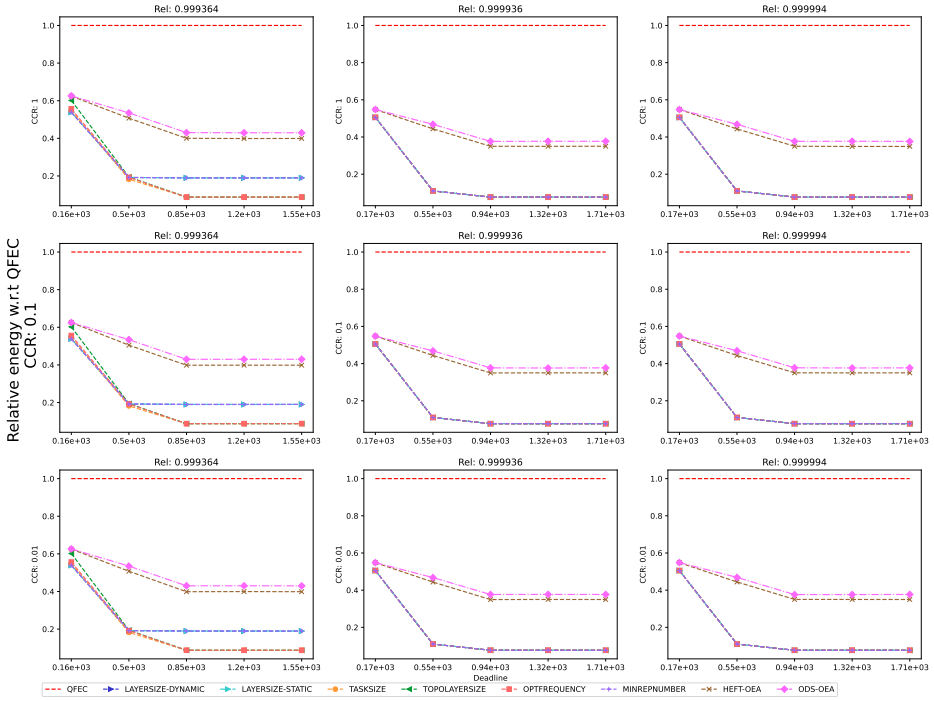

Figure 649: Performance of the different heuristics on the Seismology workflow.

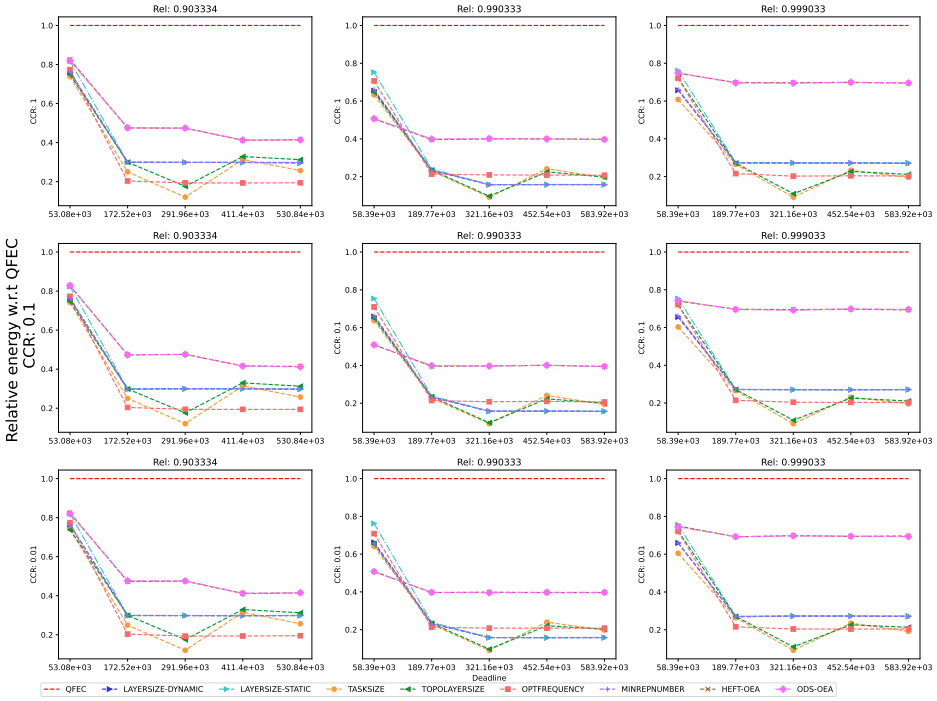

Figure 650: Performance of the different heuristics on the SoyKB workflow.

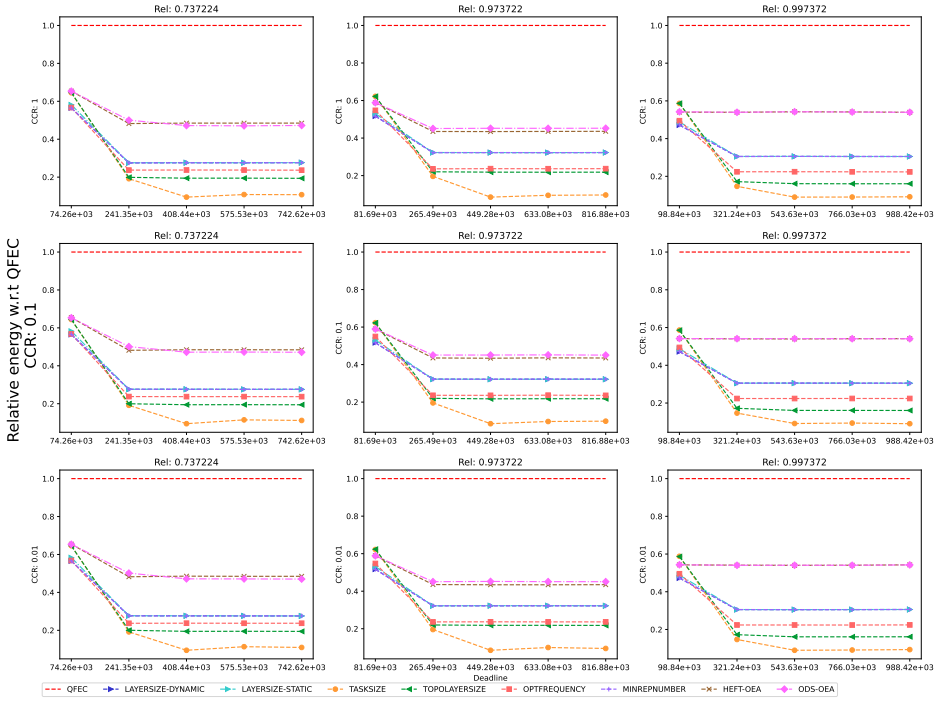

Figure 651: Performance of the different heuristics on the SRASearch.

E.6.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

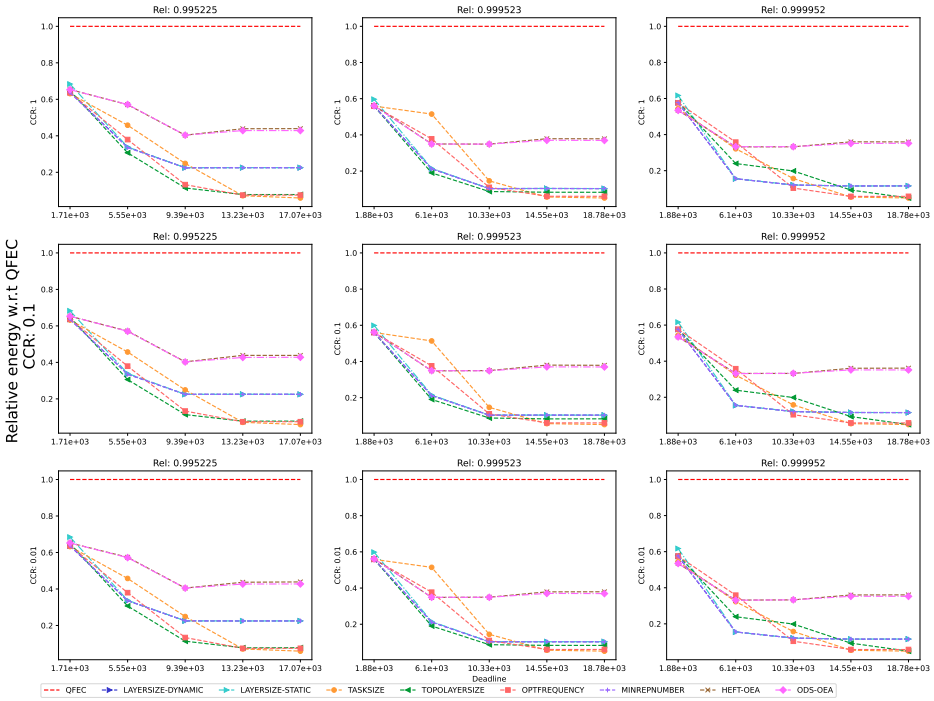

Figure 652: Performance of the different heuristics on the BWA workflow.

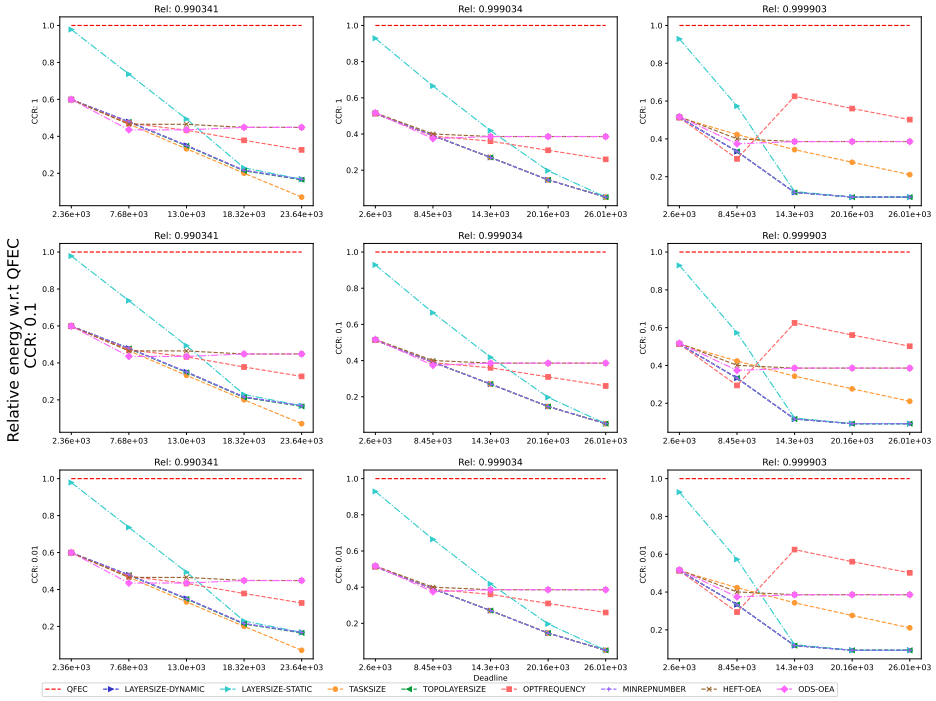

Figure 653: Performance of the different heuristics on the Cholesky workflow.

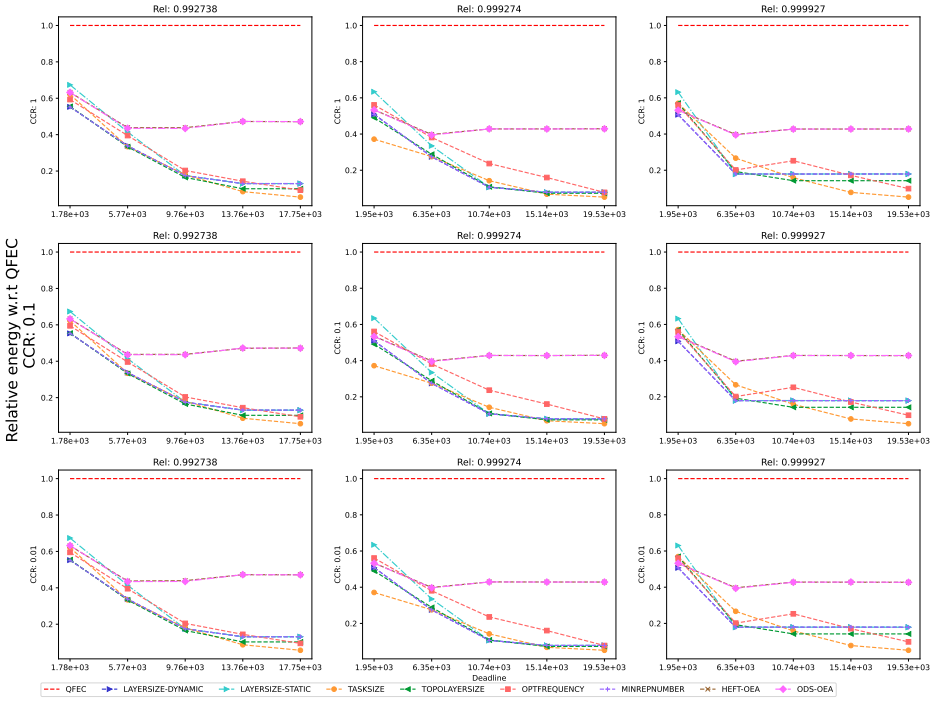

Figure 654: Performance of the different heuristics on the Cycles workflow.

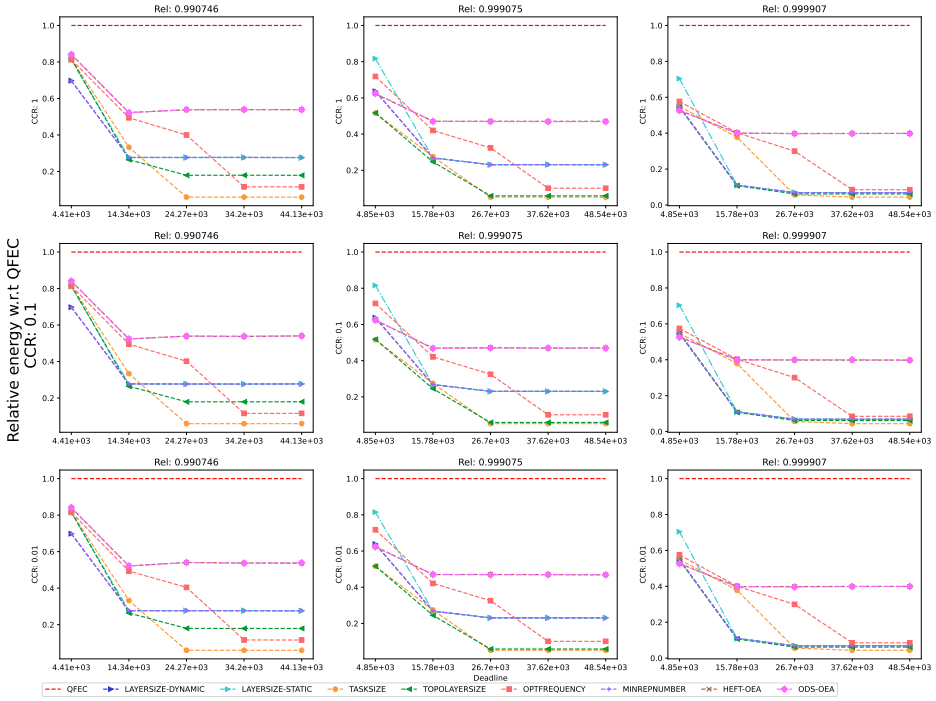

Figure 655: Performance of the different heuristics on the Epigenomics workflow.

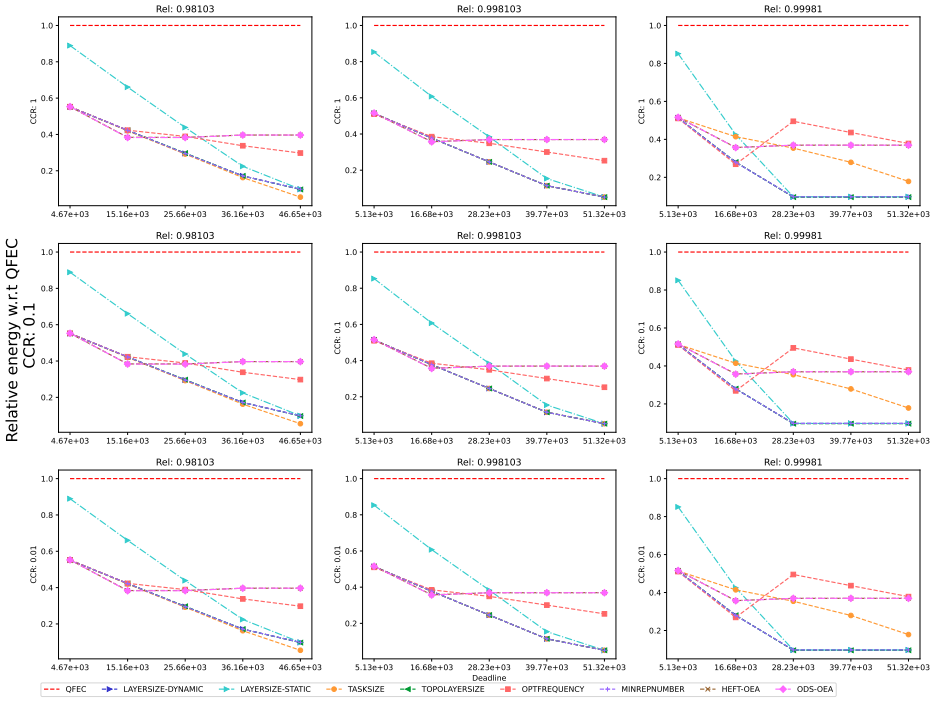

Figure 656: Performance of the different heuristics on the LU workflow.

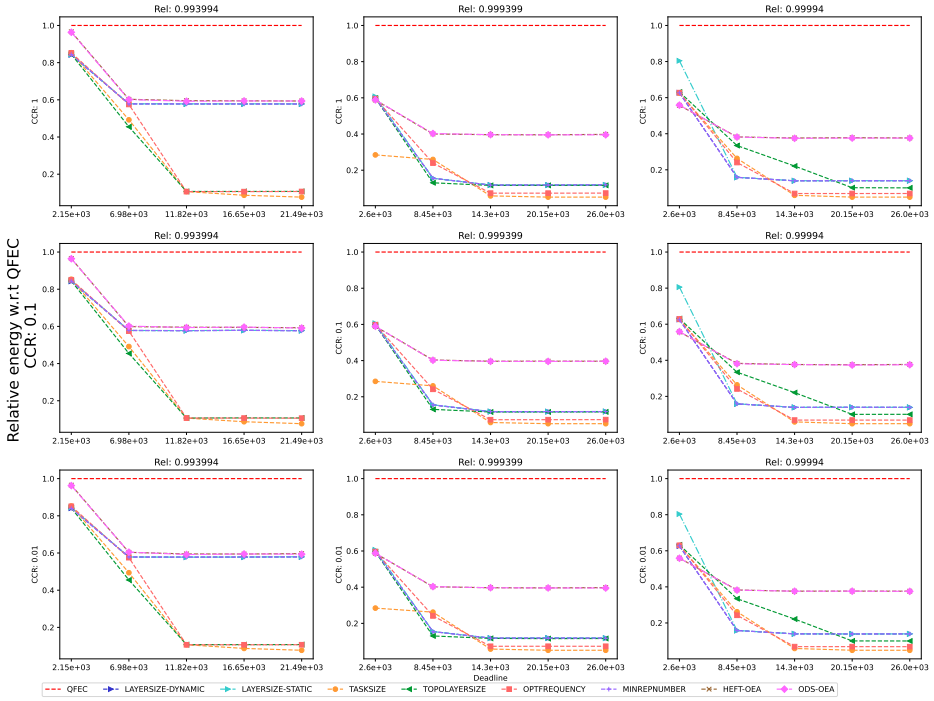

Figure 657: Performance of the different heuristics on the Montage workflow.

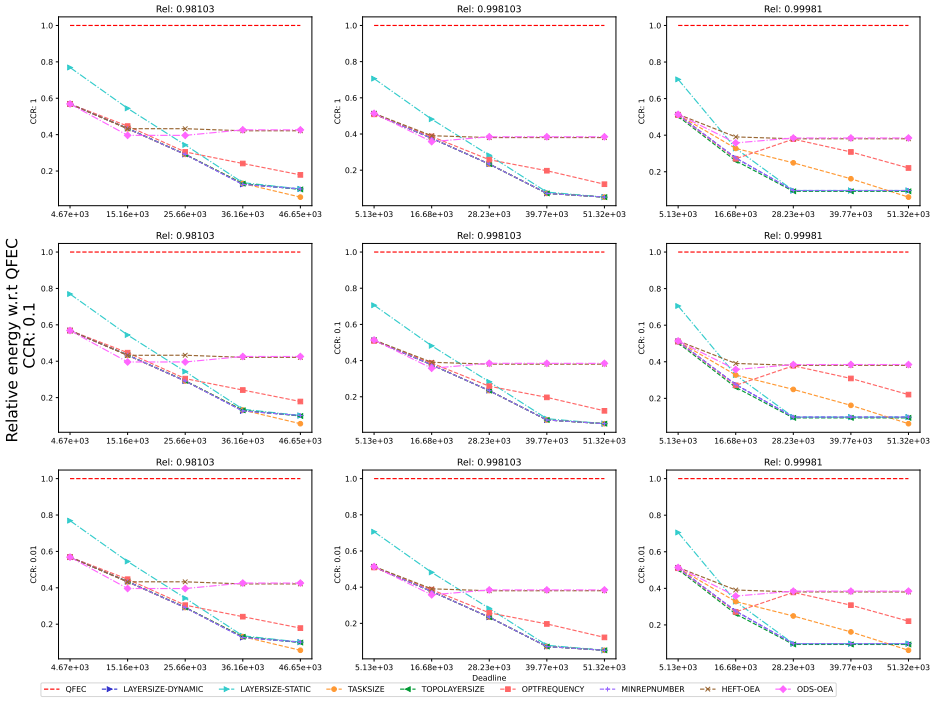

Figure 658: Performance of the different heuristics on the QR workflow.

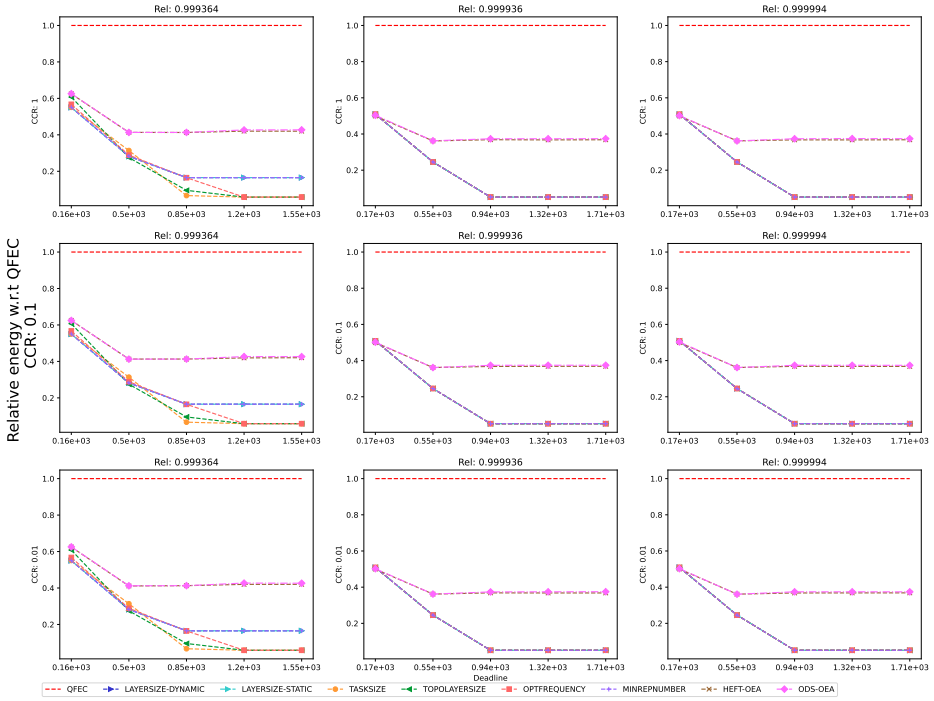

Figure 659: Performance of the different heuristics on the Seismology workflow.

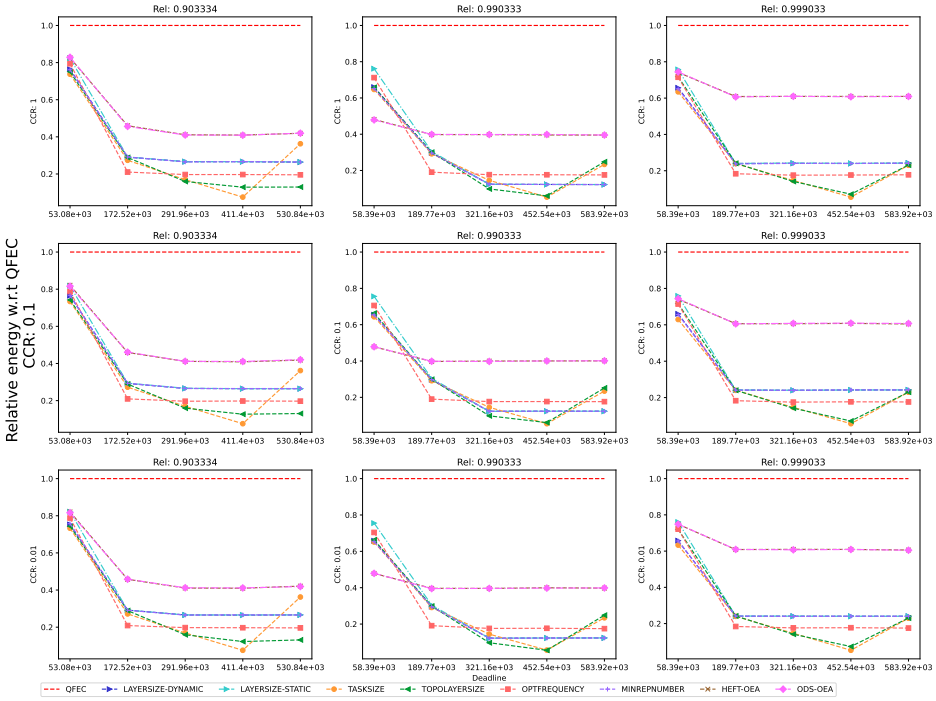

Figure 660: Performance of the different heuristics on the SoyKB workflow.

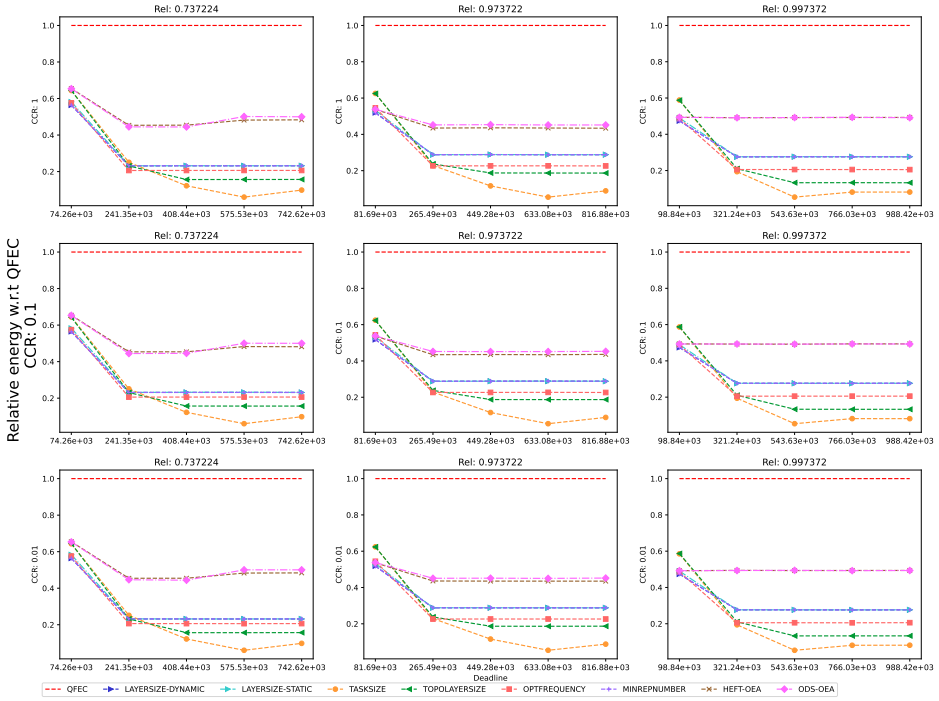

Figure 661: Performance of the different heuristics on the SRASearch.

E.6.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

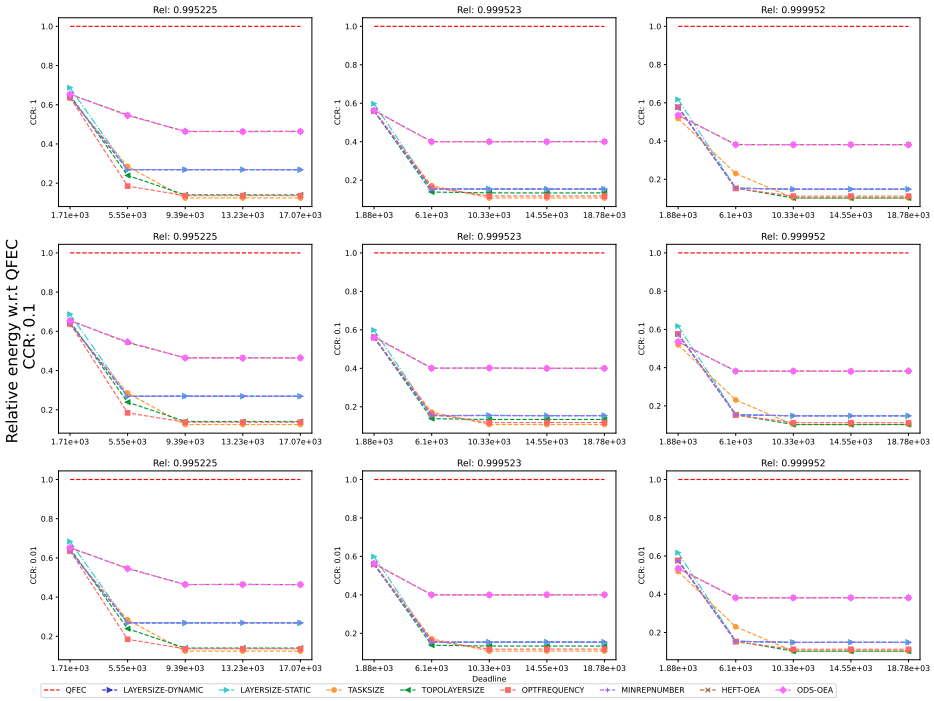

Figure 662: Performance of the different heuristics on the BWA workflow.

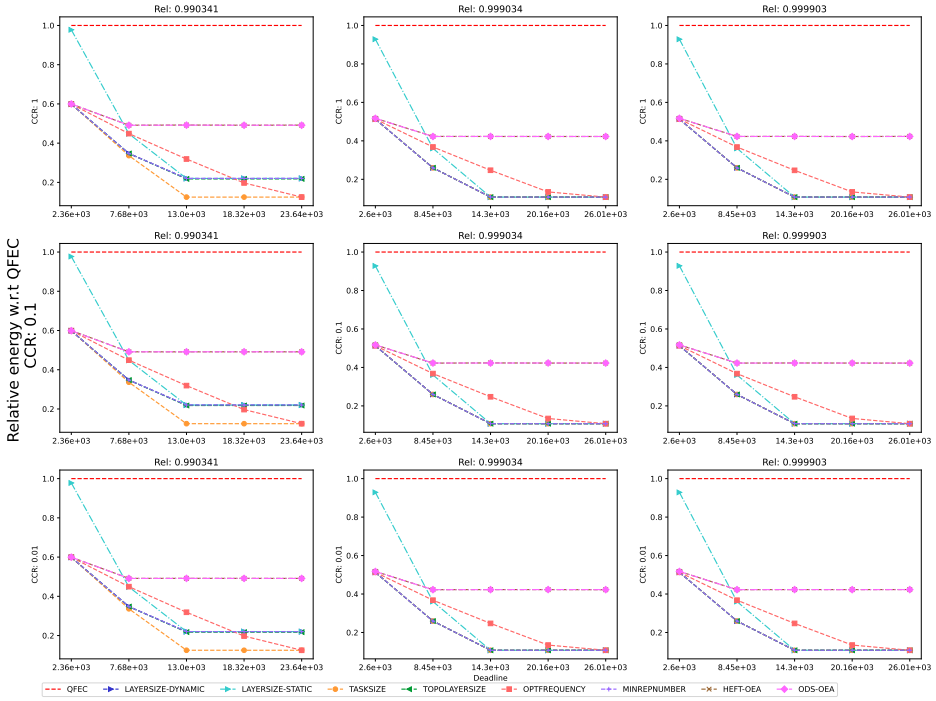

Figure 663: Performance of the different heuristics on the Cholesky workflow.

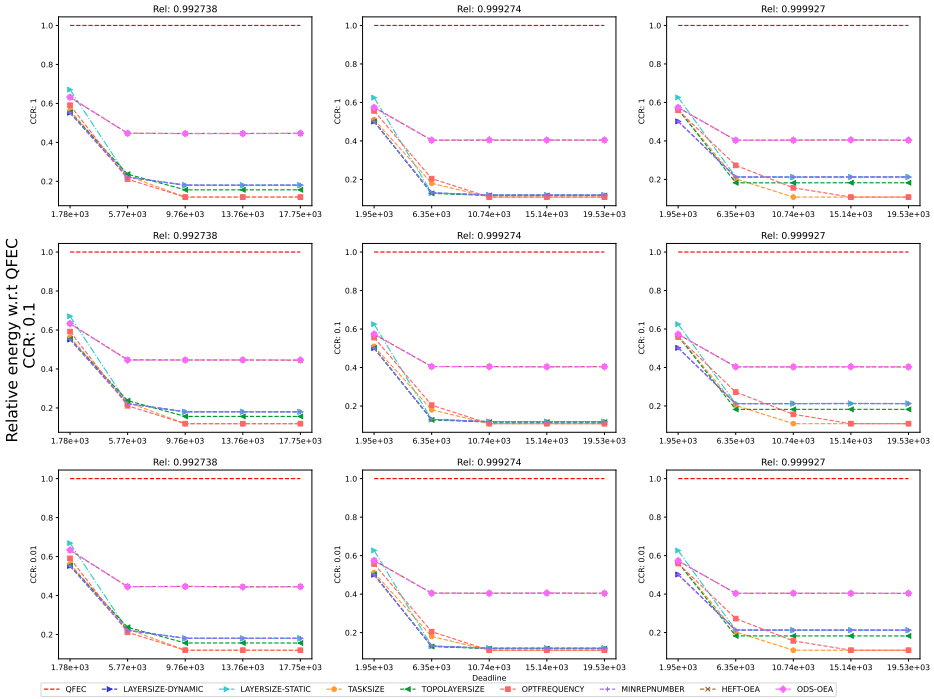

Figure 664: Performance of the different heuristics on the Cycles workflow.

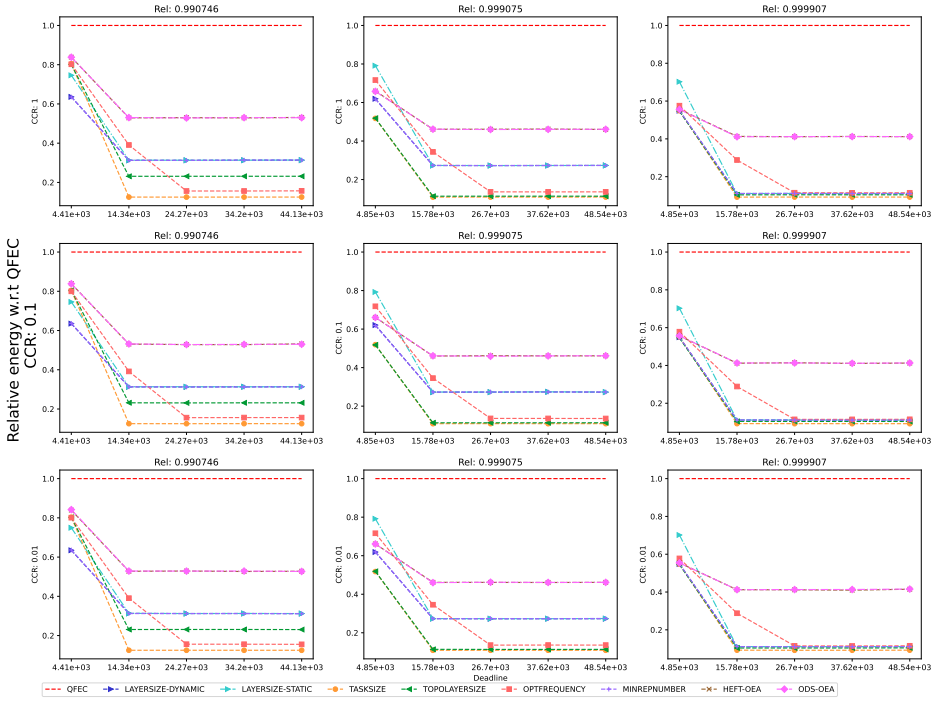

Figure 665: Performance of the different heuristics on the Epigenomics workflow.

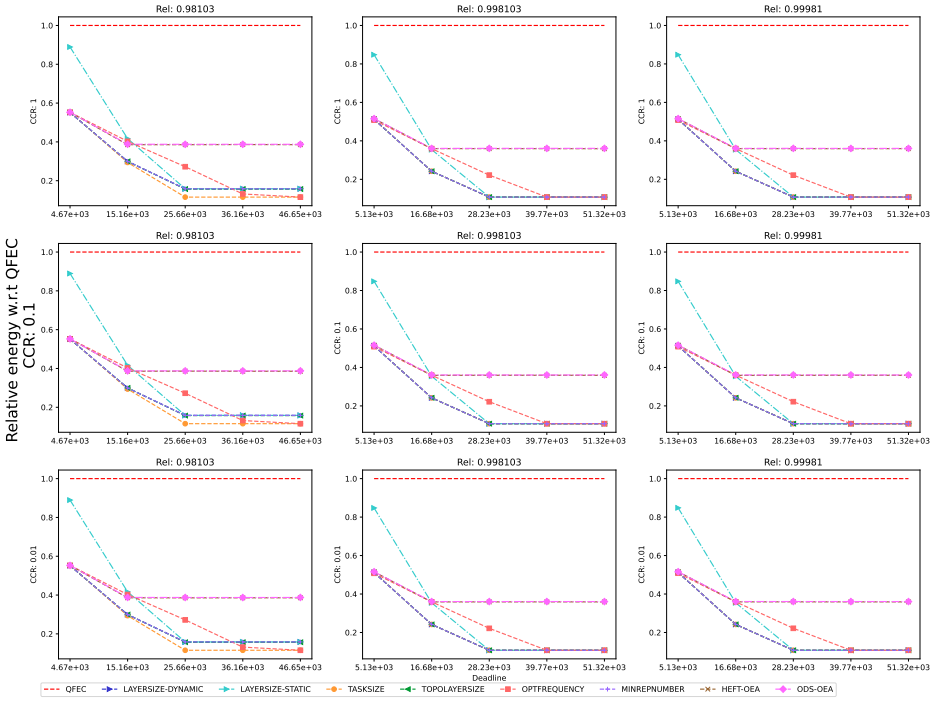

Figure 666: Performance of the different heuristics on the LU workflow.

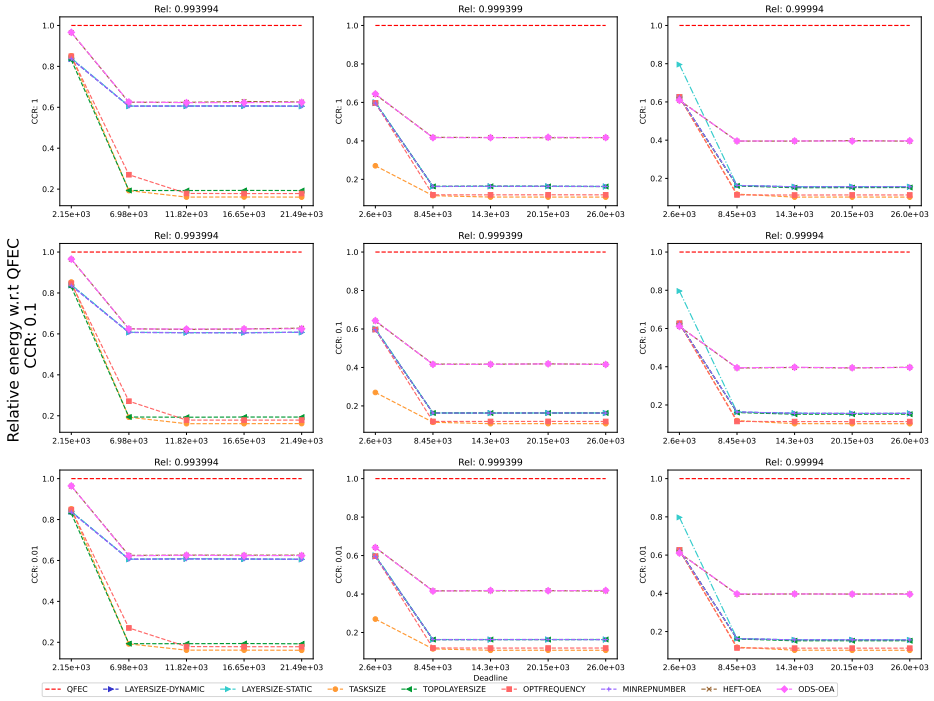

Figure 667: Performance of the different heuristics on the Montage workflow.

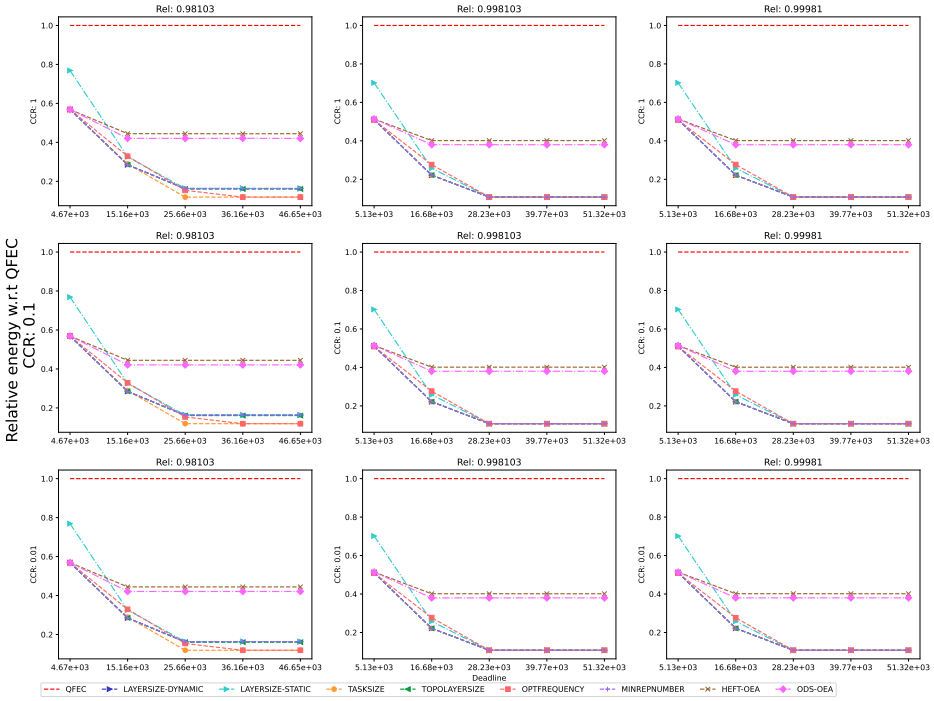

Figure 668: Performance of the different heuristics on the QR workflow.

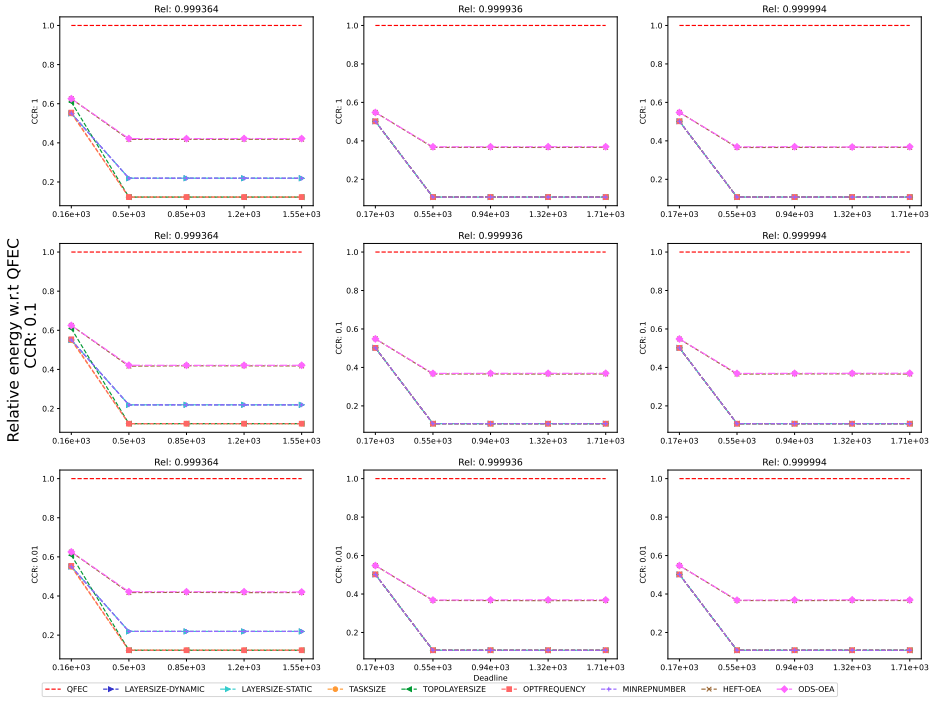

Figure 669: Performance of the different heuristics on the Seismology workflow.

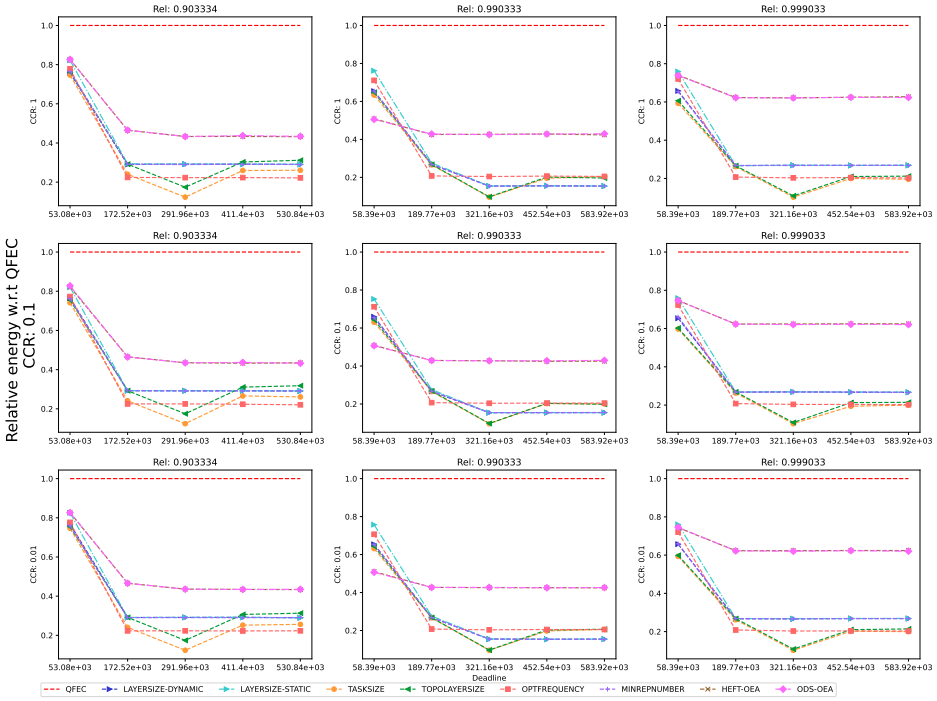

Figure 670: Performance of the different heuristics on the SoyKB workflow.

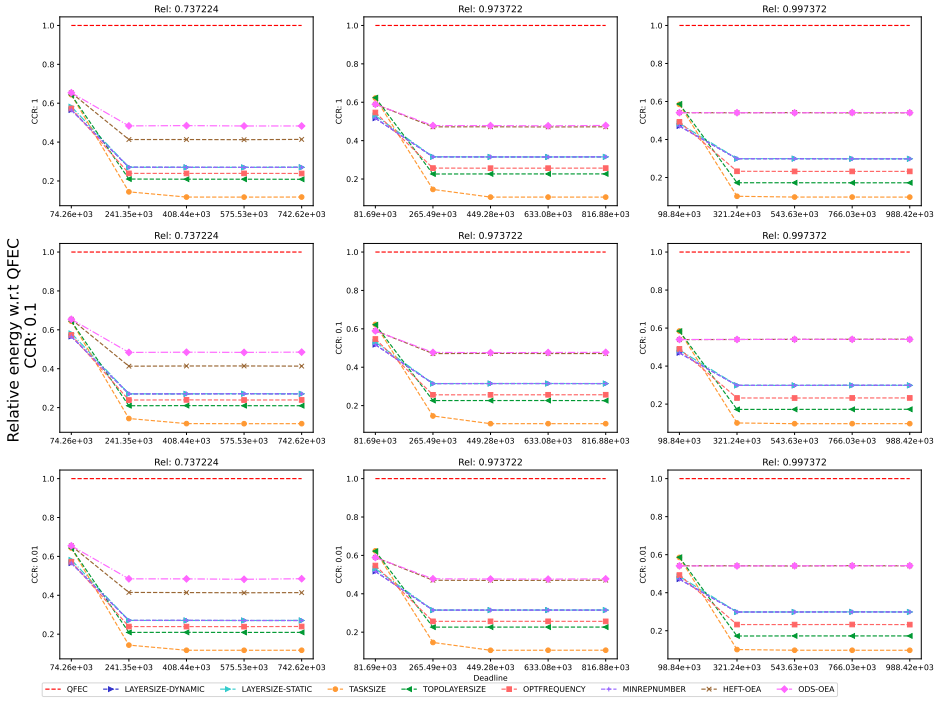

Figure 671: Performance of the different heuristics on the SRASearch.

E.6.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

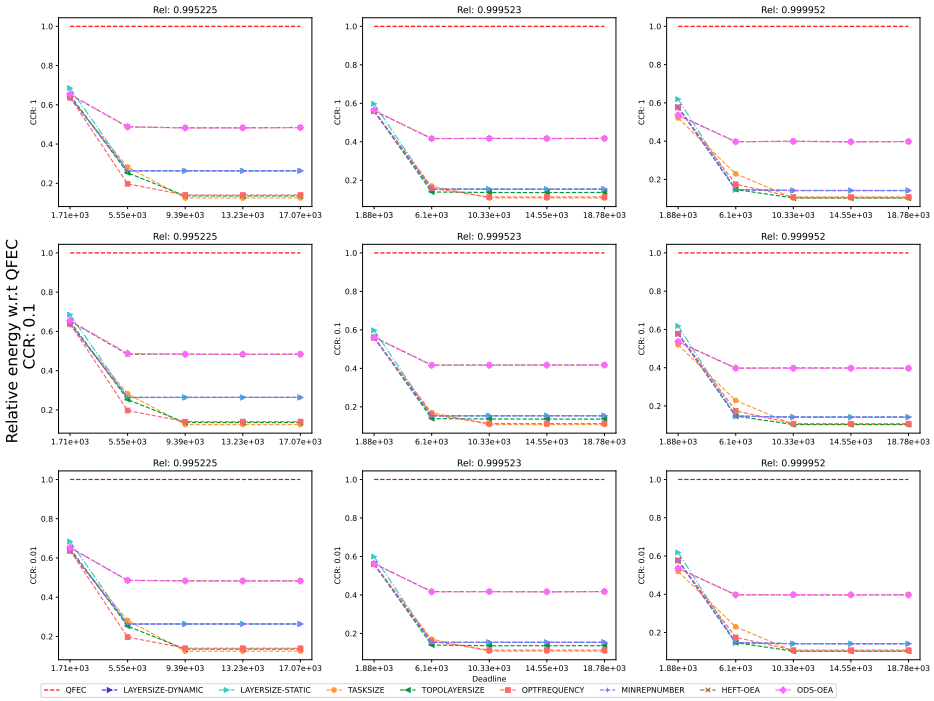

Figure 672: Performance of the different heuristics on the BWA workflow.

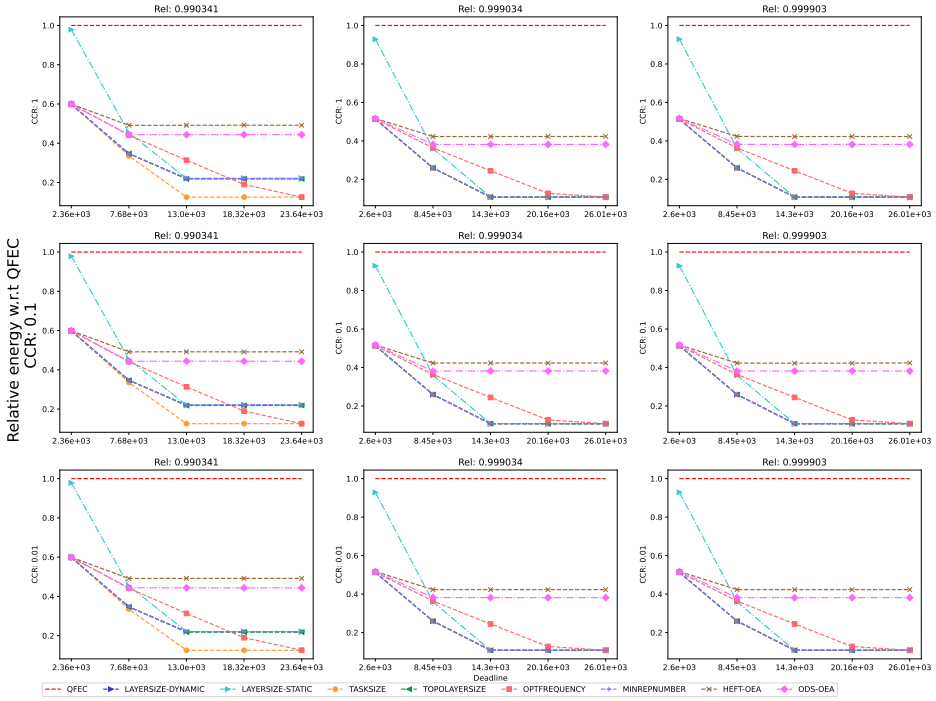

Figure 673: Performance of the different heuristics on the Cholesky workflow.

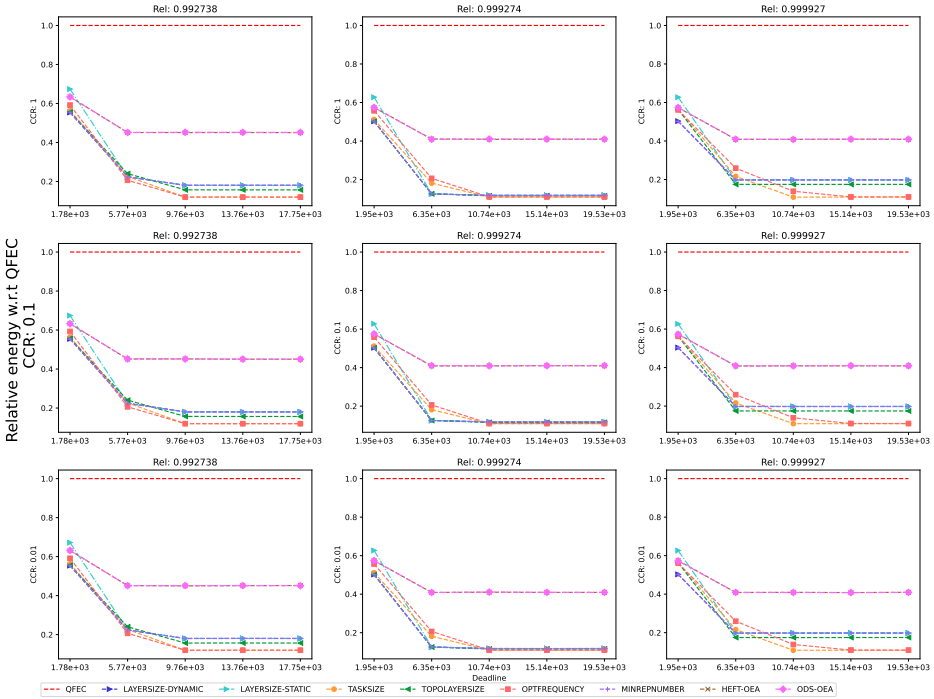

Figure 674: Performance of the different heuristics on the Cycles workflow.

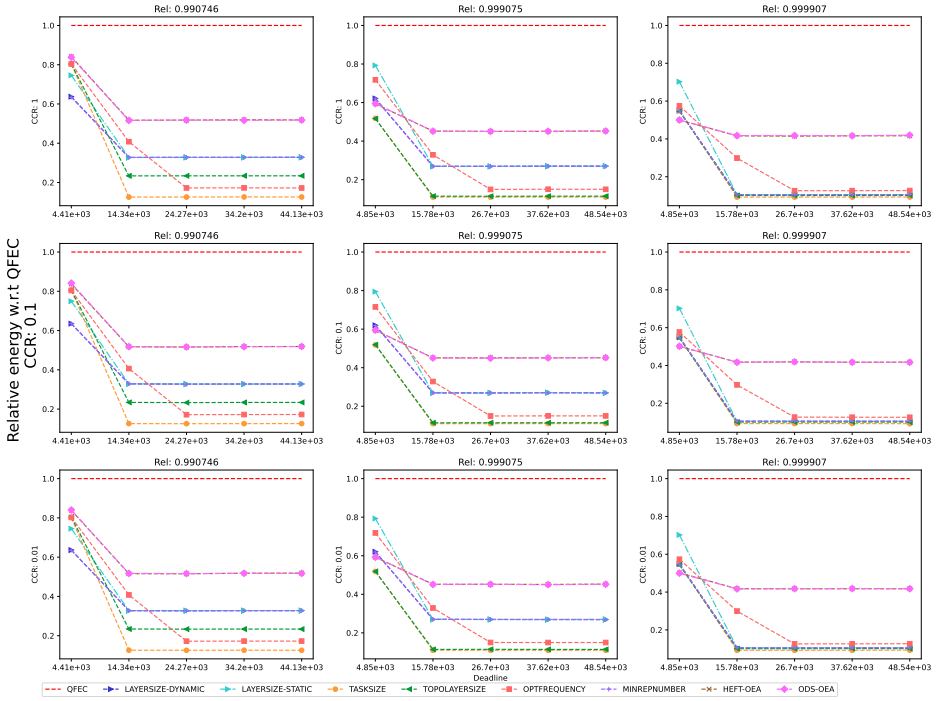

Figure 675: Performance of the different heuristics on the Epigenomics workflow.

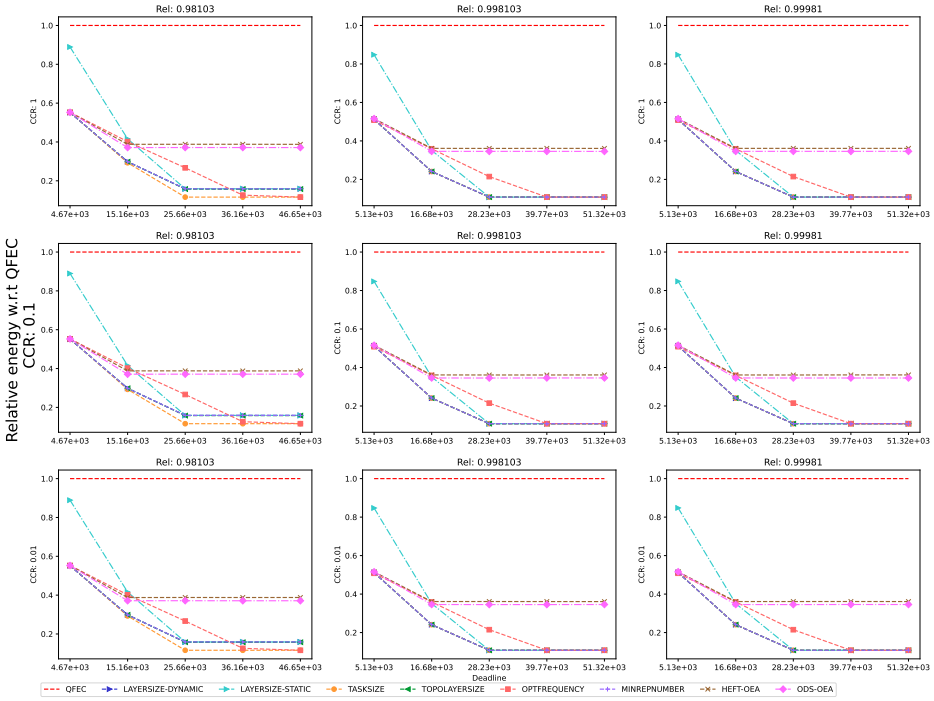

Figure 676: Performance of the different heuristics on the LU workflow.

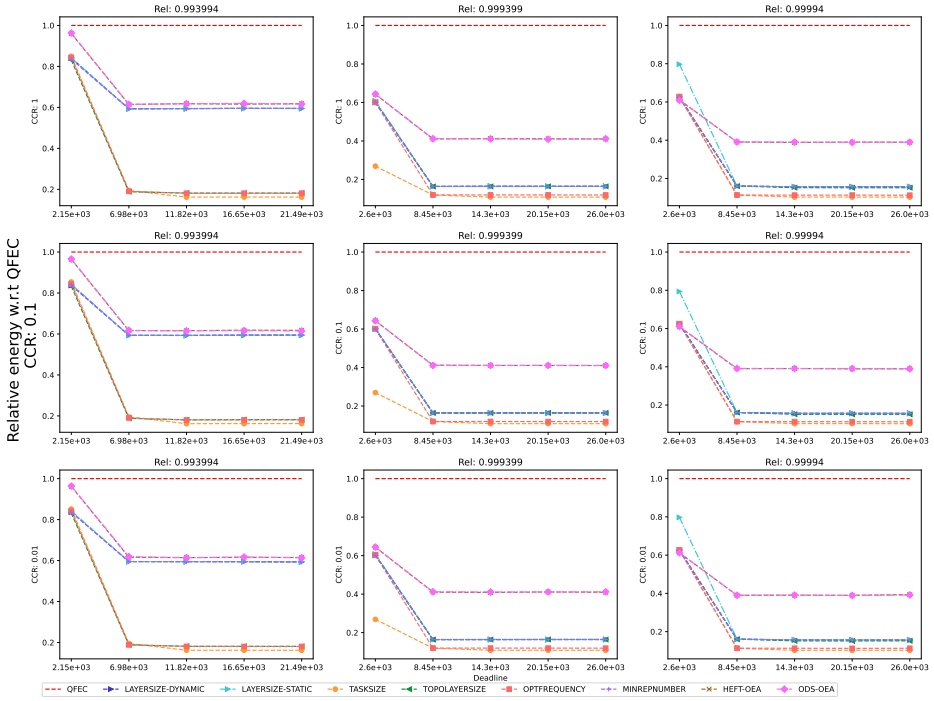

Figure 677: Performance of the different heuristics on the Montage workflow.

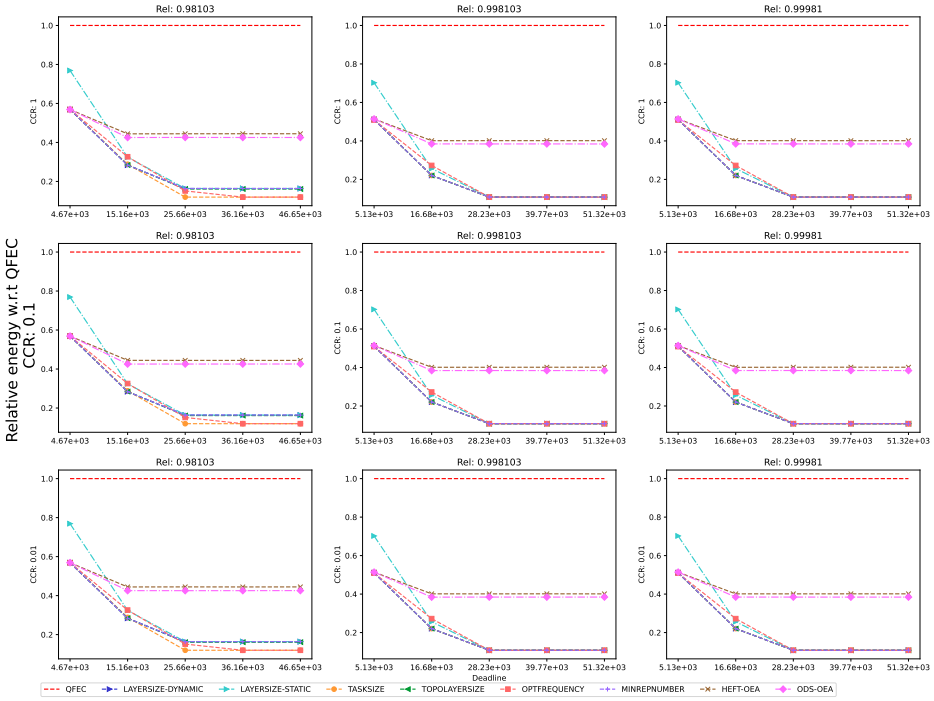

Figure 678: Performance of the different heuristics on the QR workflow.

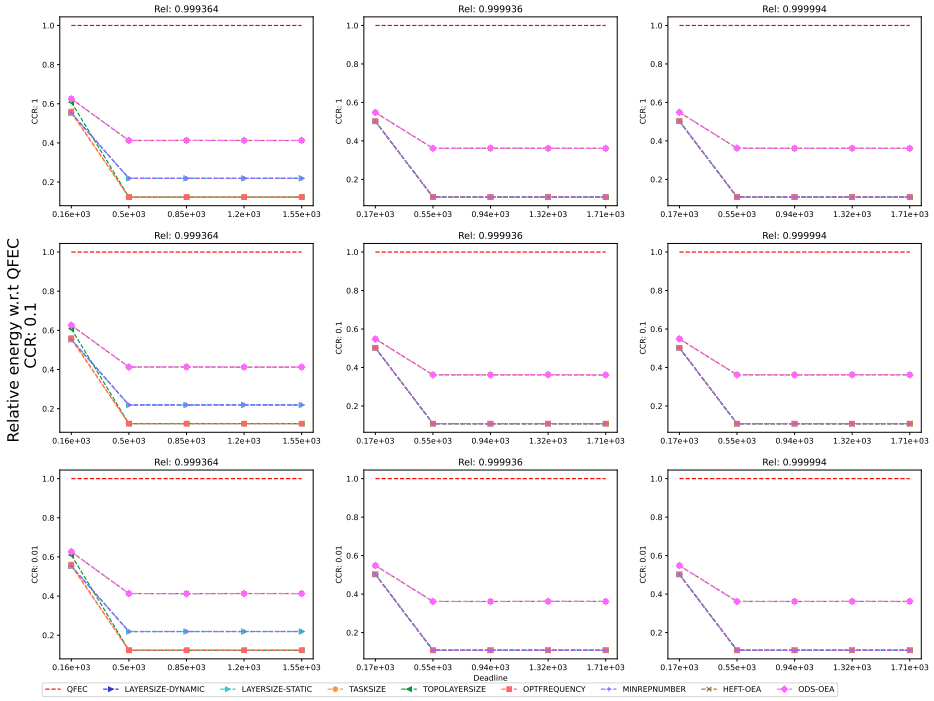

Figure 679: Performance of the different heuristics on the Seismology workflow.

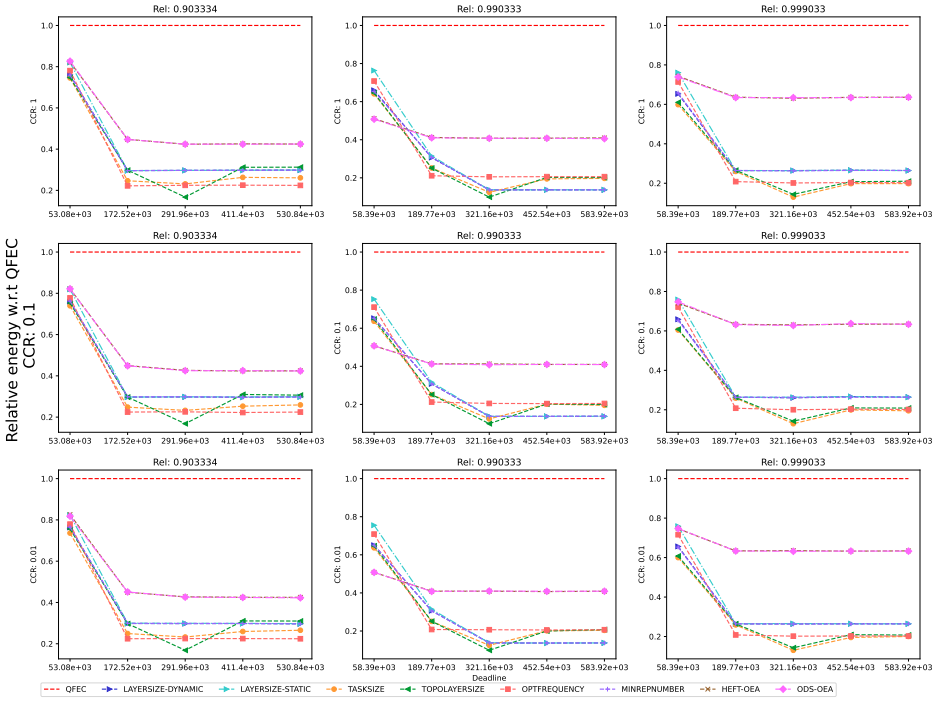

Figure 680: Performance of the different heuristics on the SoyKB workflow.

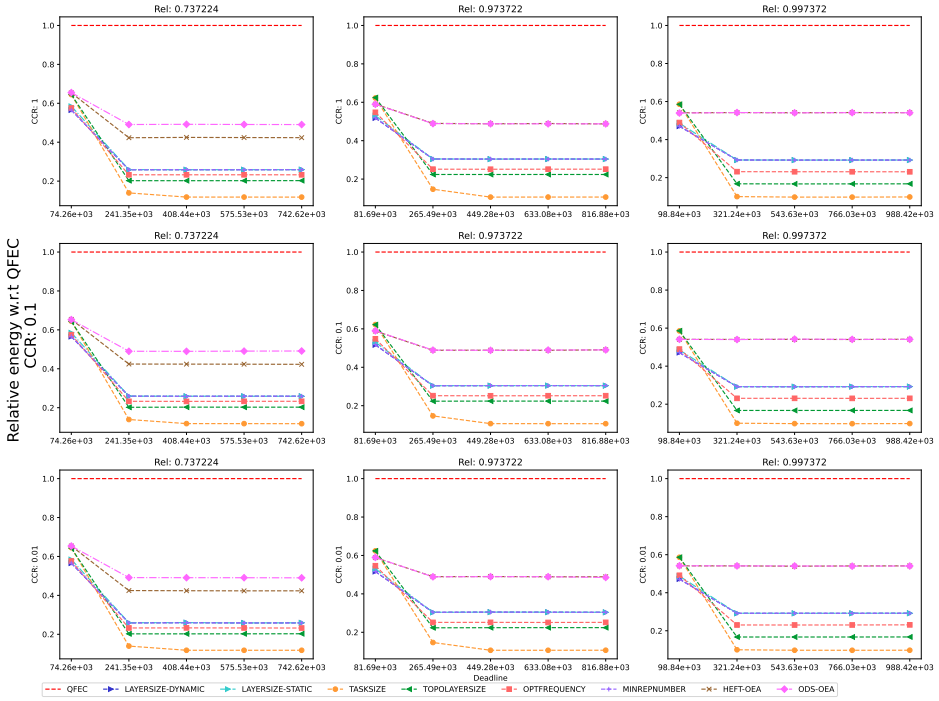

Figure 681: Performance of the different heuristics on the SRASearch.

E.7 $BC/WC = 0.7$

E.7.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

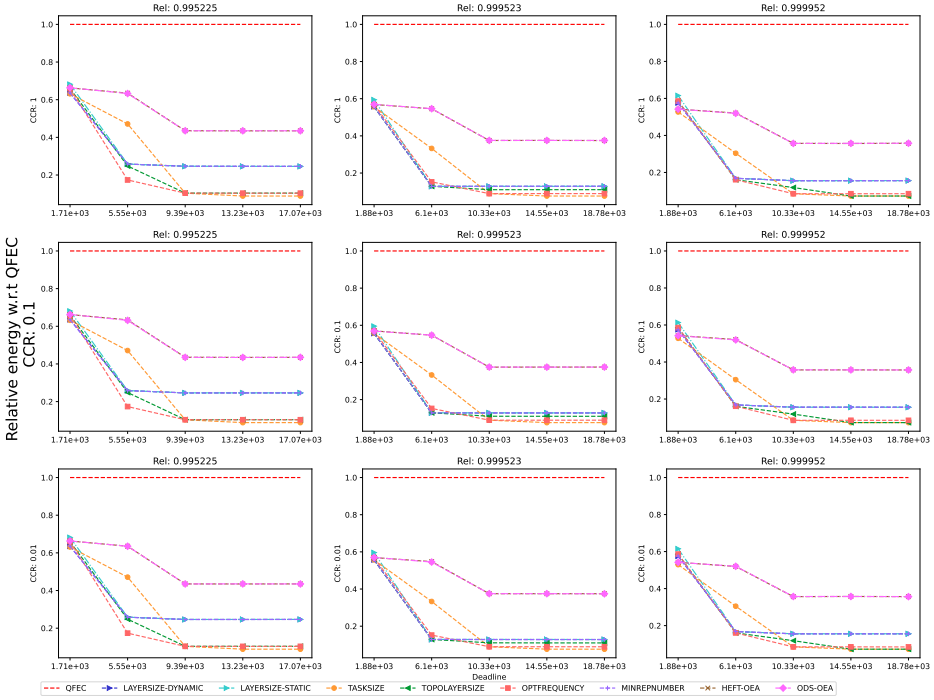

Figure 682: Performance of the different heuristics on the BWA workflow.

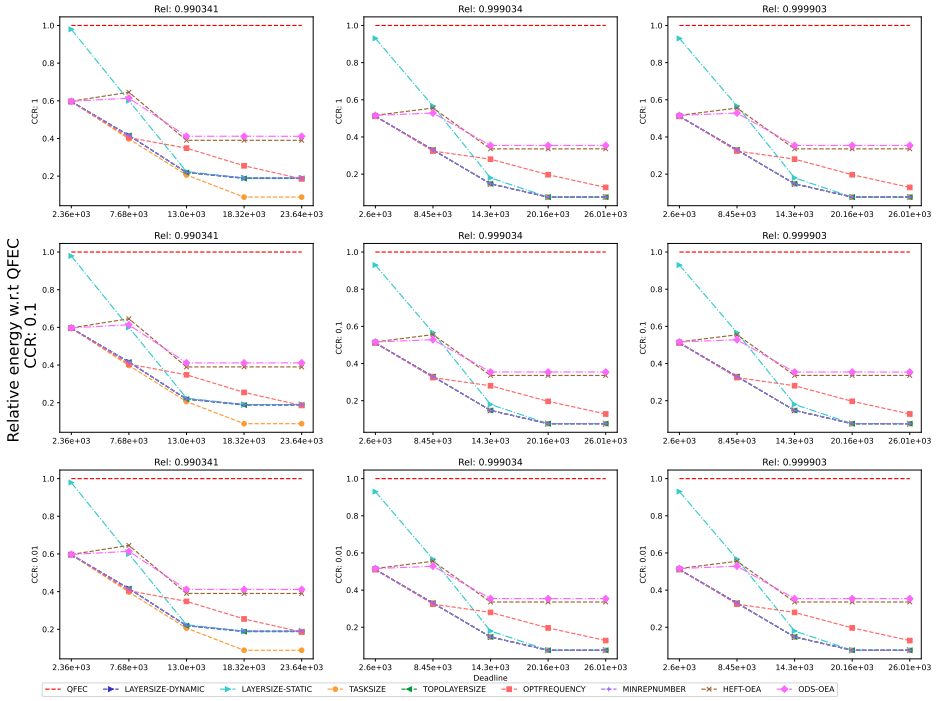

Figure 683: Performance of the different heuristics on the Cholesky workflow.

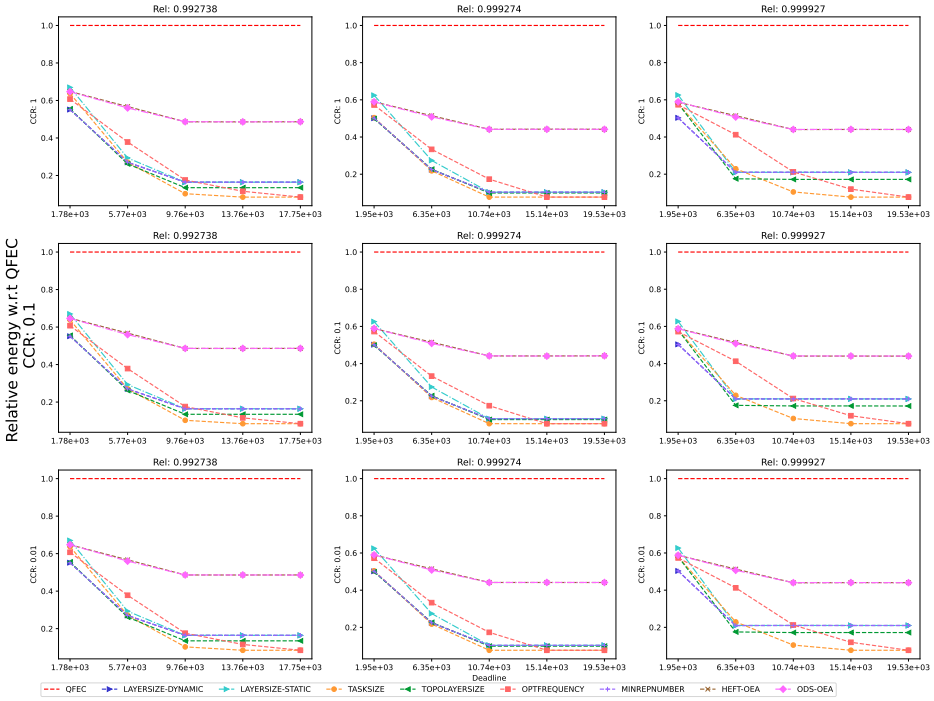

Figure 684: Performance of the different heuristics on the Cycles workflow.

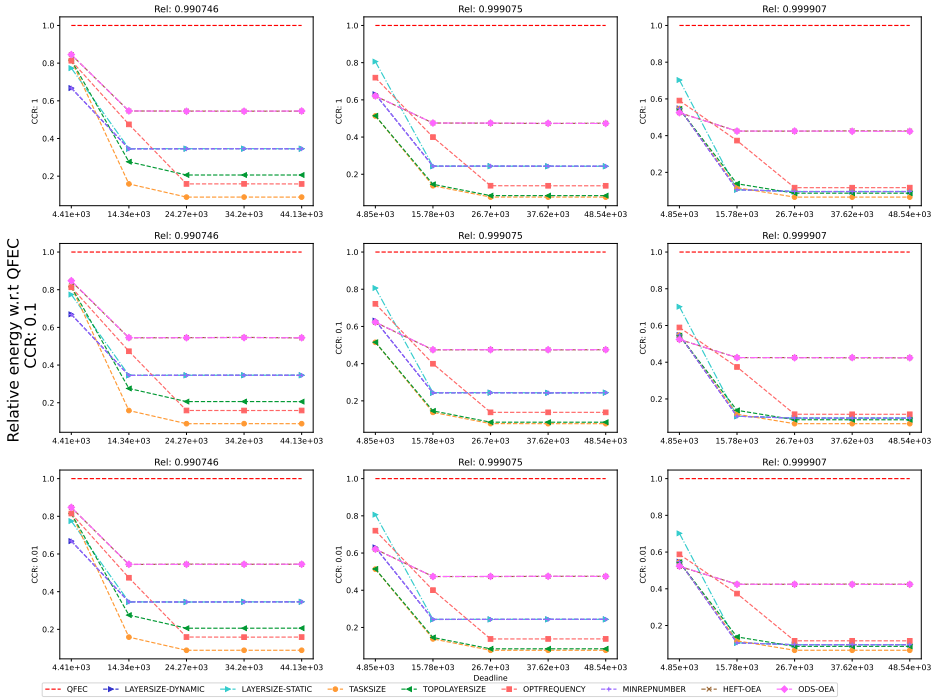

Figure 685: Performance of the different heuristics on the Epigenomics workflow.

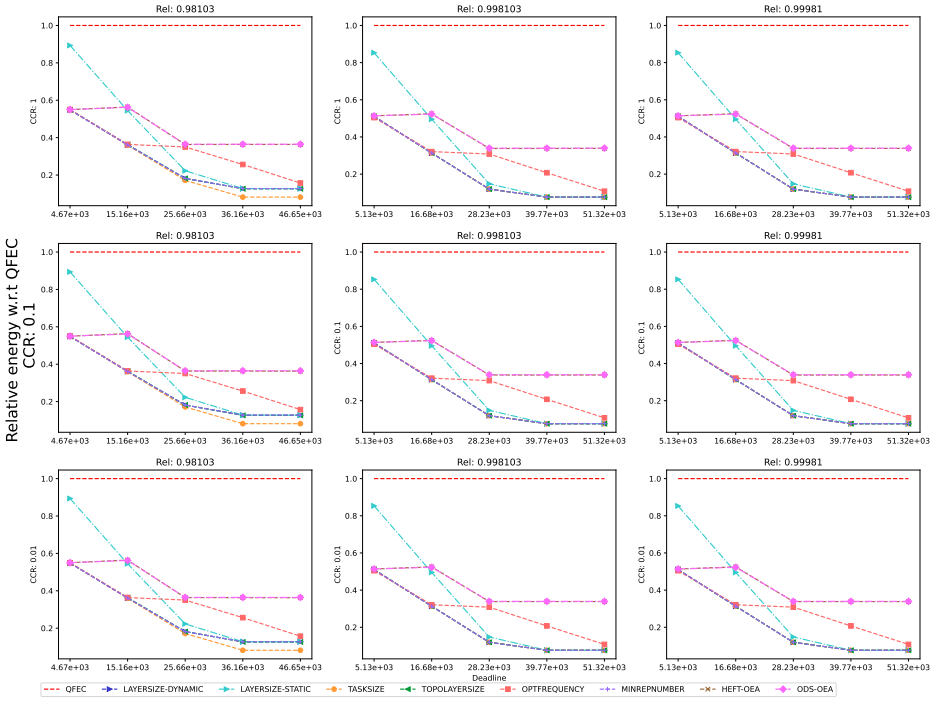

Figure 686: Performance of the different heuristics on the LU workflow.

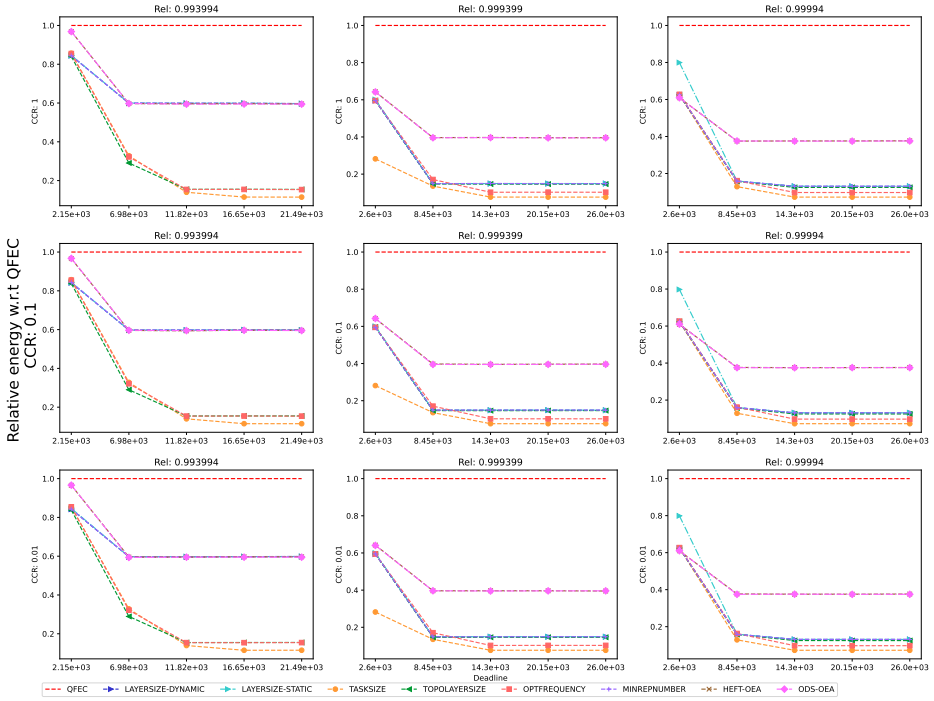

Figure 687: Performance of the different heuristics on the Montage workflow.

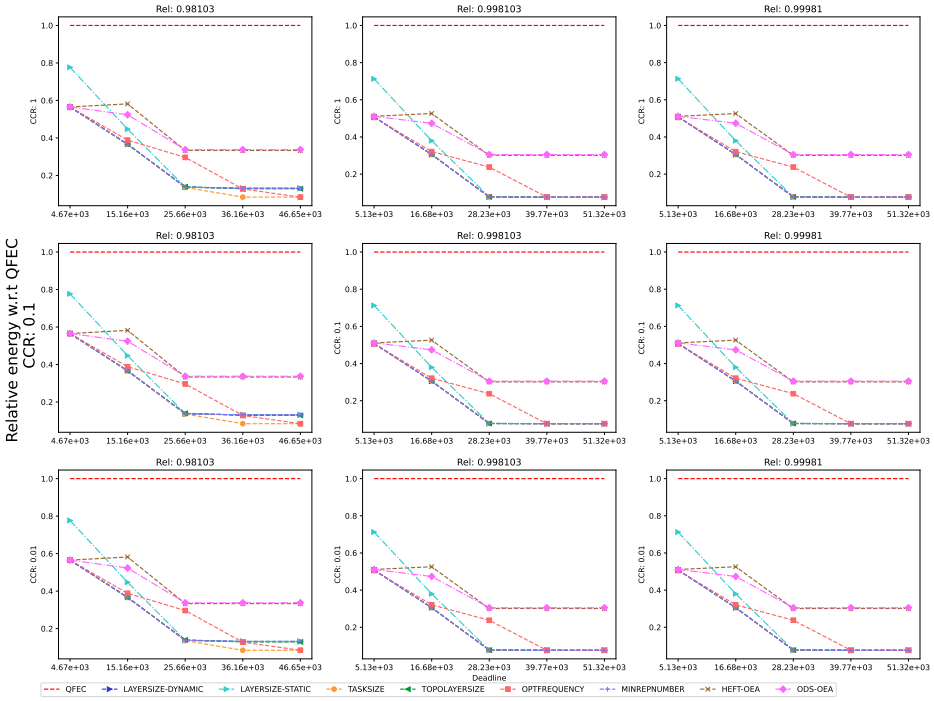

Figure 688: Performance of the different heuristics on the QR workflow.

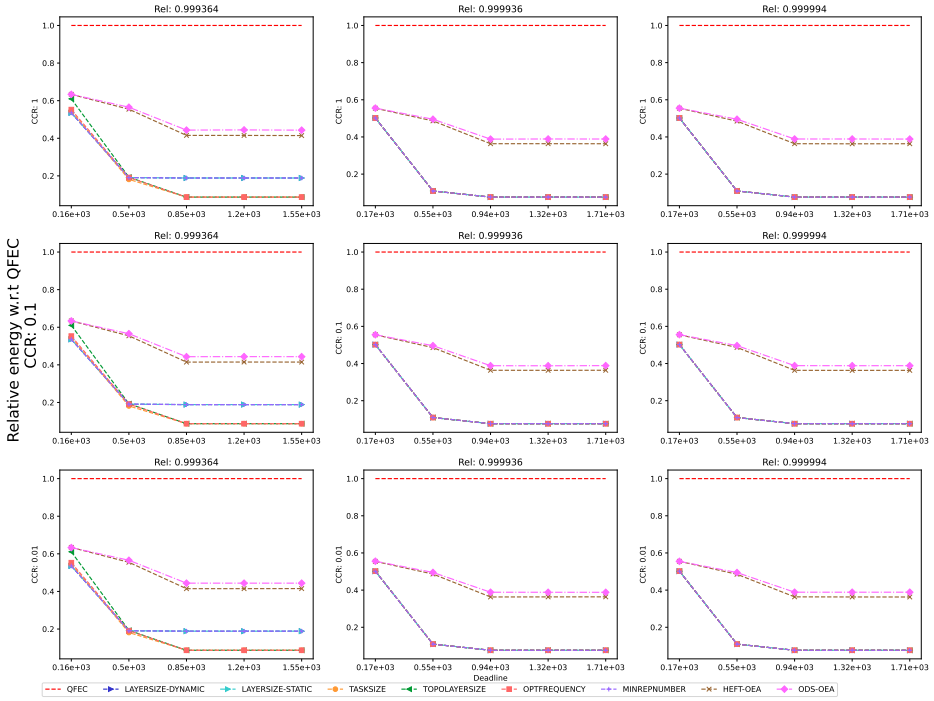

Figure 689: Performance of the different heuristics on the Seismology workflow.

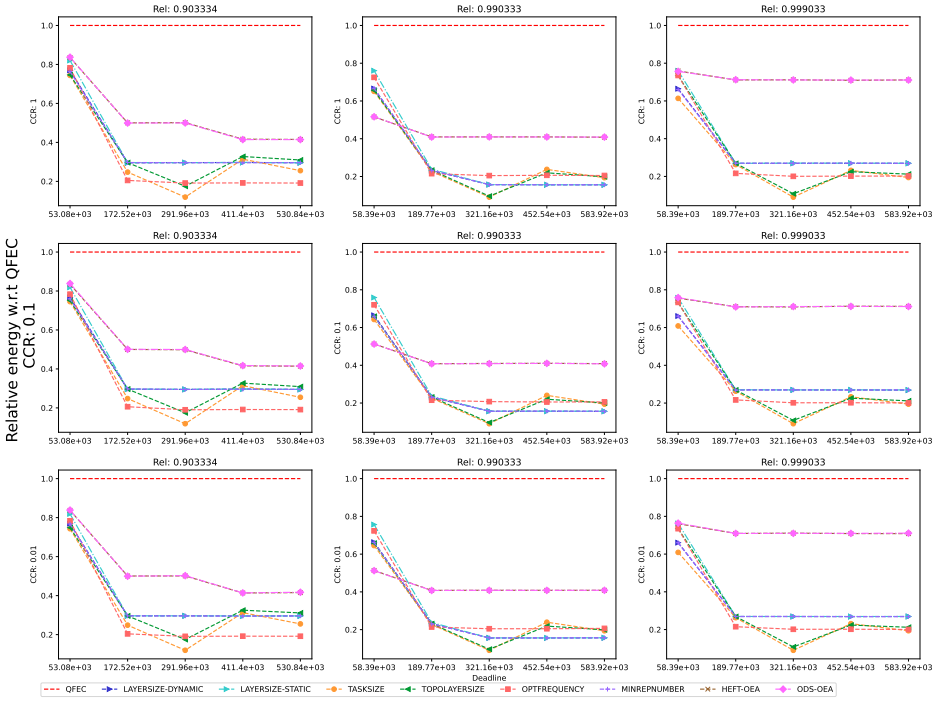

Figure 690: Performance of the different heuristics on the SoyKB workflow.

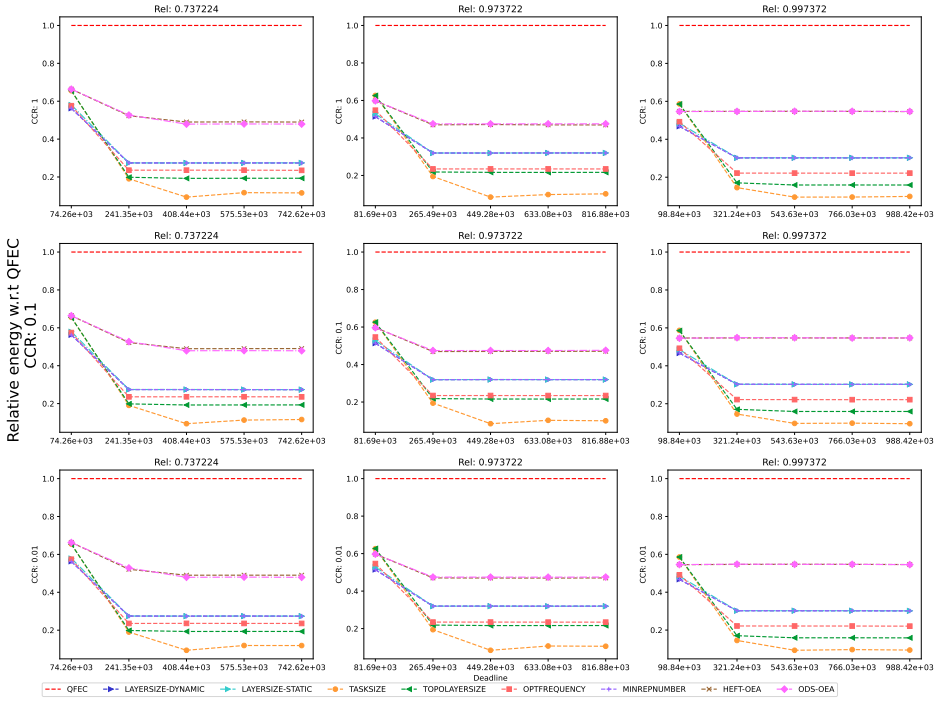

Figure 691: Performance of the different heuristics on the SRASearch.

E.7.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

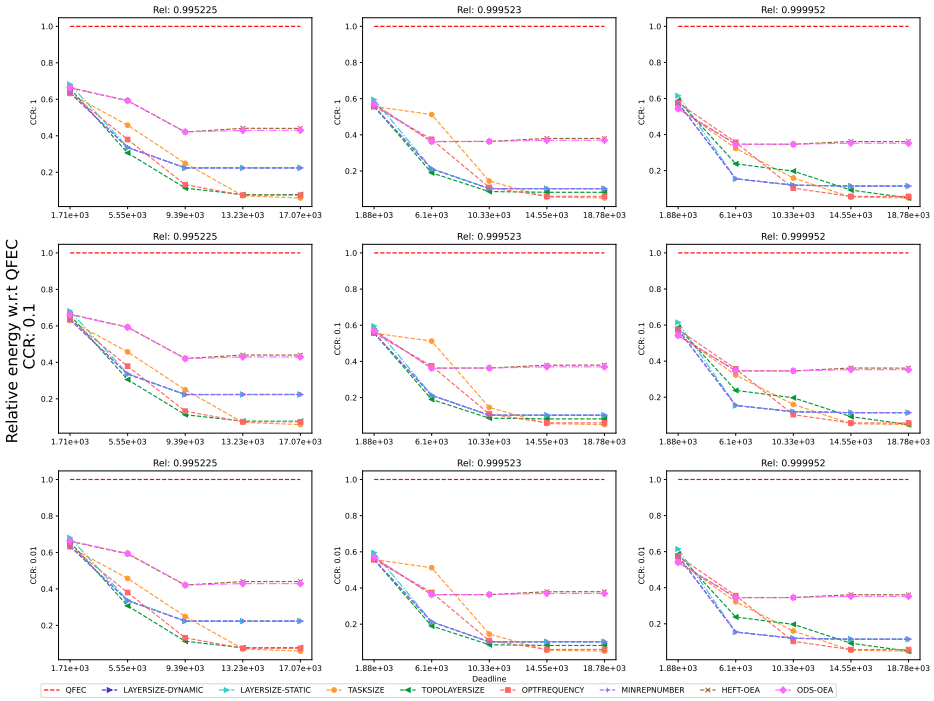

Figure 692: Performance of the different heuristics on the BWA workflow.

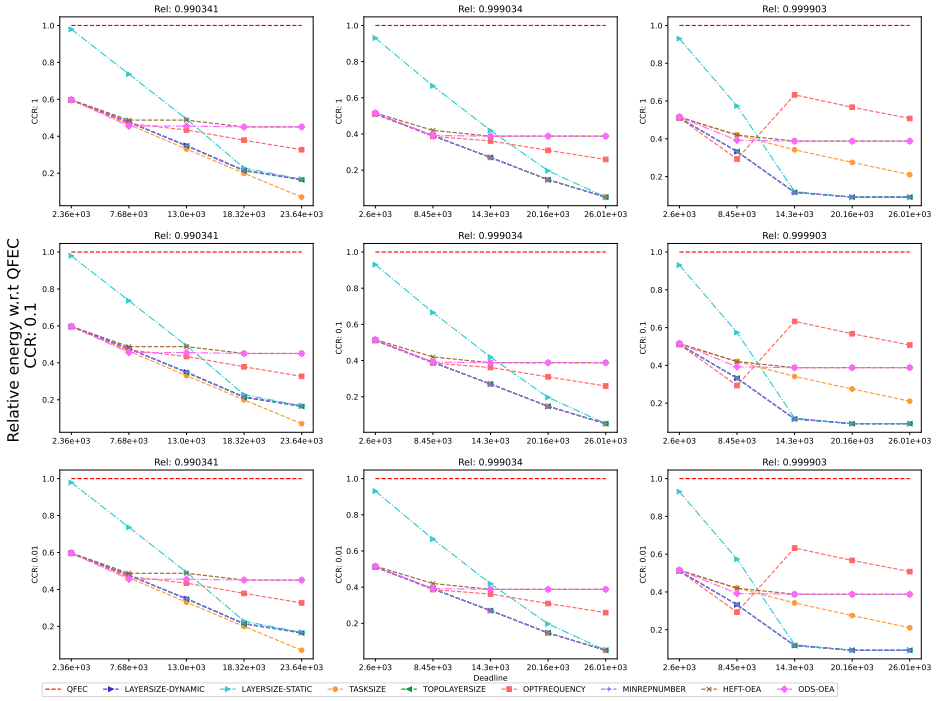

Figure 693: Performance of the different heuristics on the Cholesky workflow.

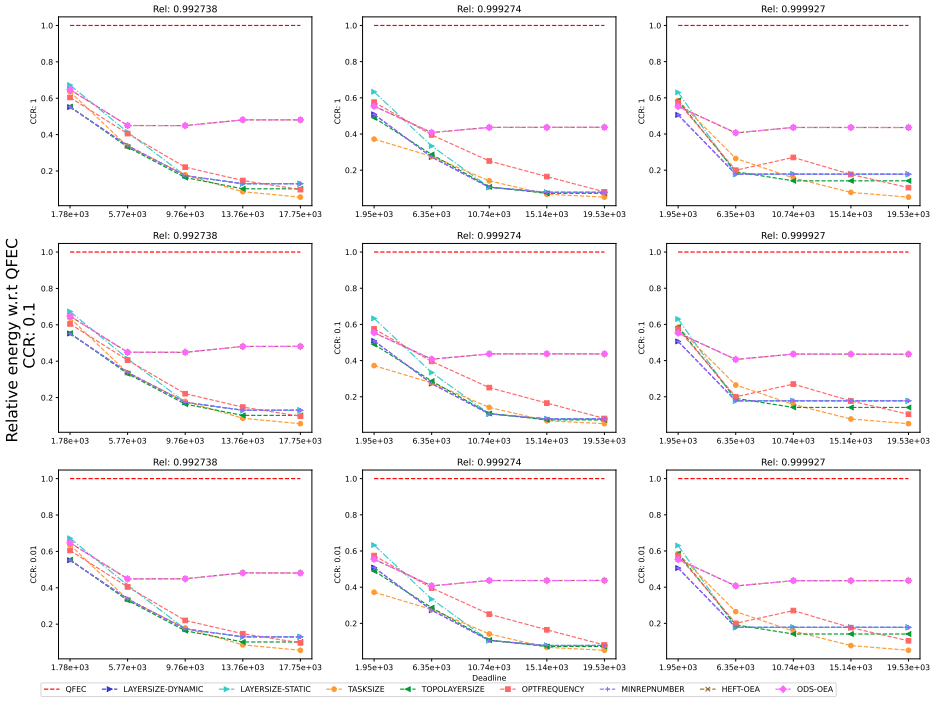

Figure 694: Performance of the different heuristics on the Cycles workflow.

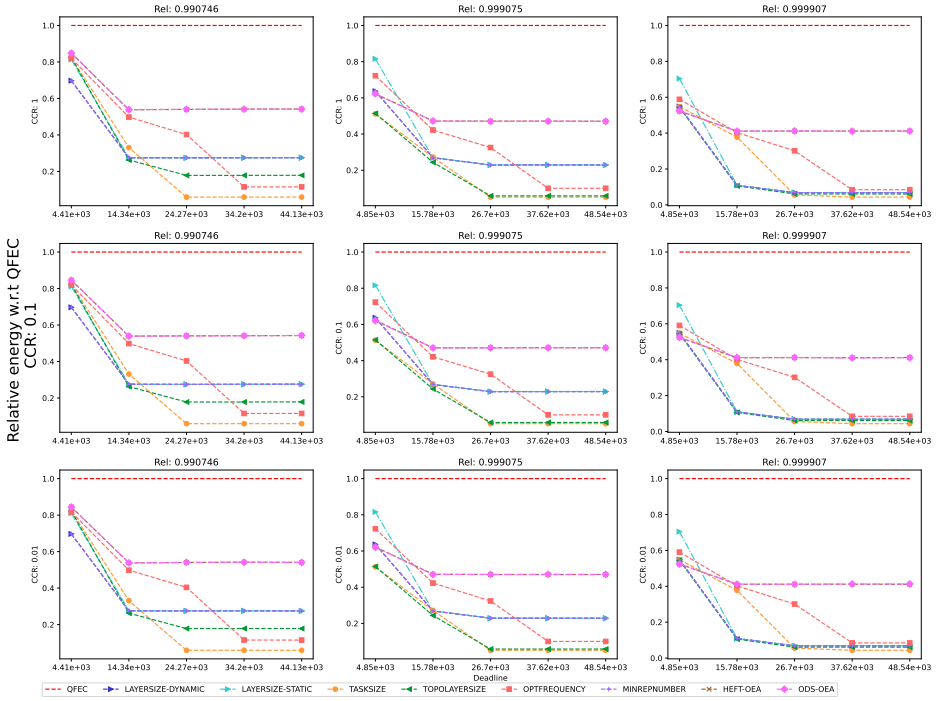

Figure 695: Performance of the different heuristics on the Epigenomics workflow.

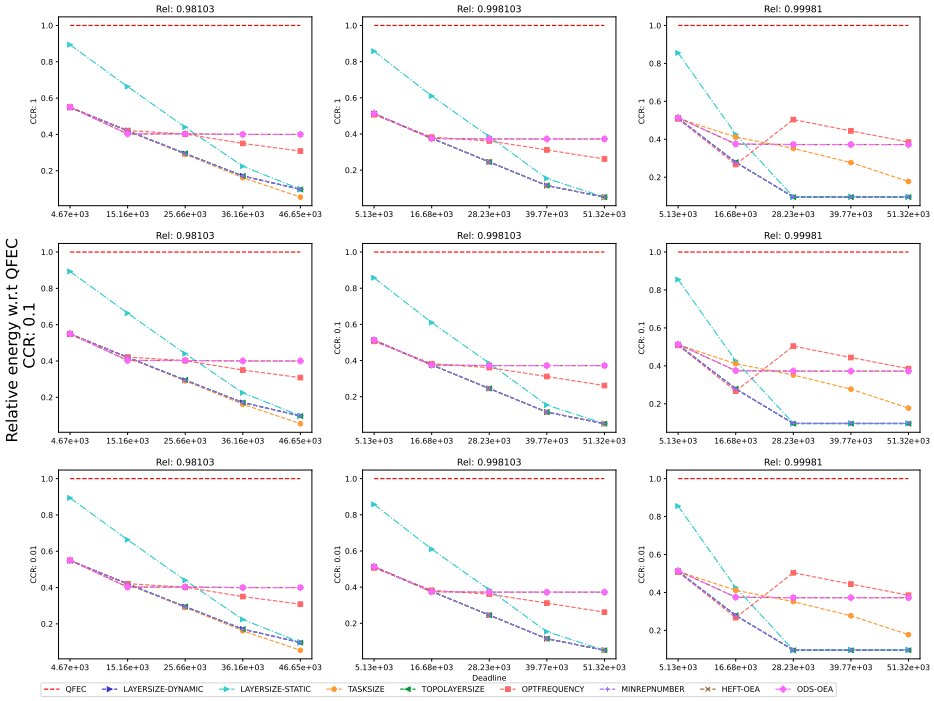

Figure 696: Performance of the different heuristics on the LU workflow.

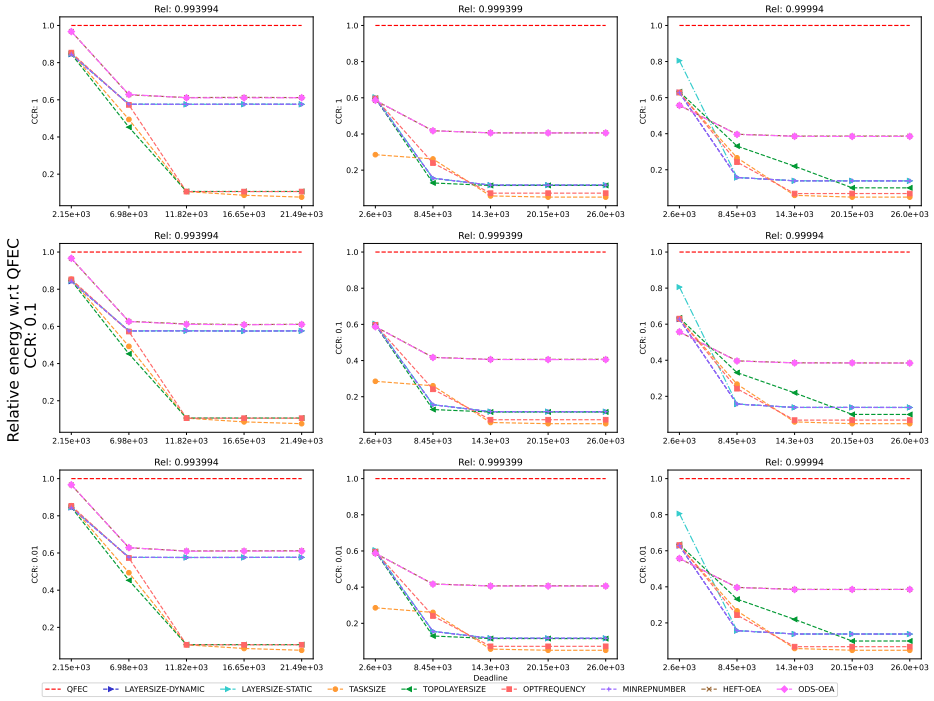

Figure 697: Performance of the different heuristics on the Montage workflow.

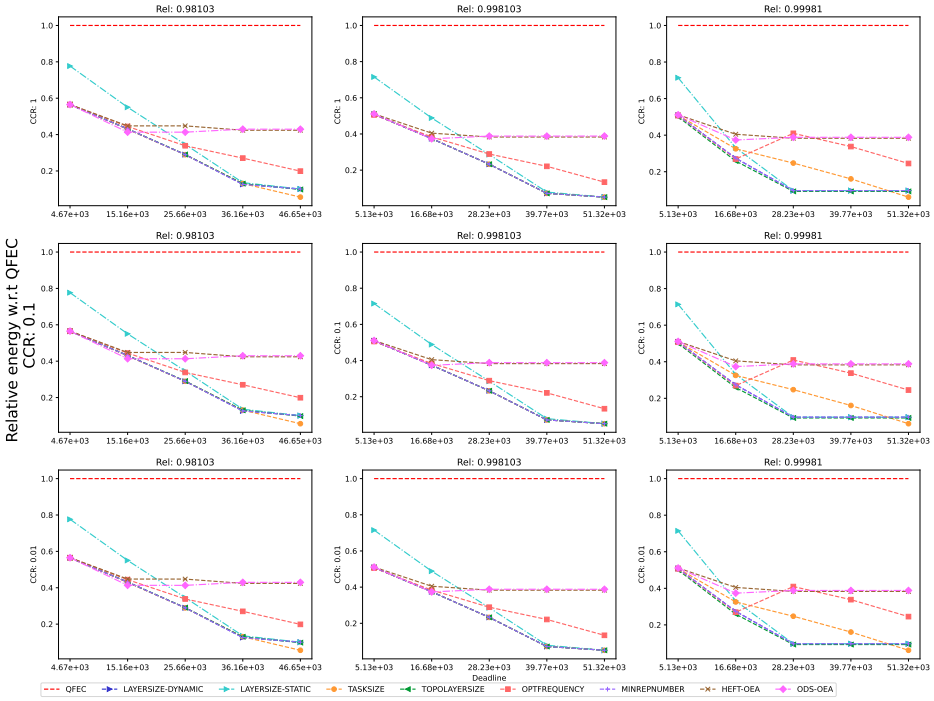

Figure 698: Performance of the different heuristics on the QR workflow.

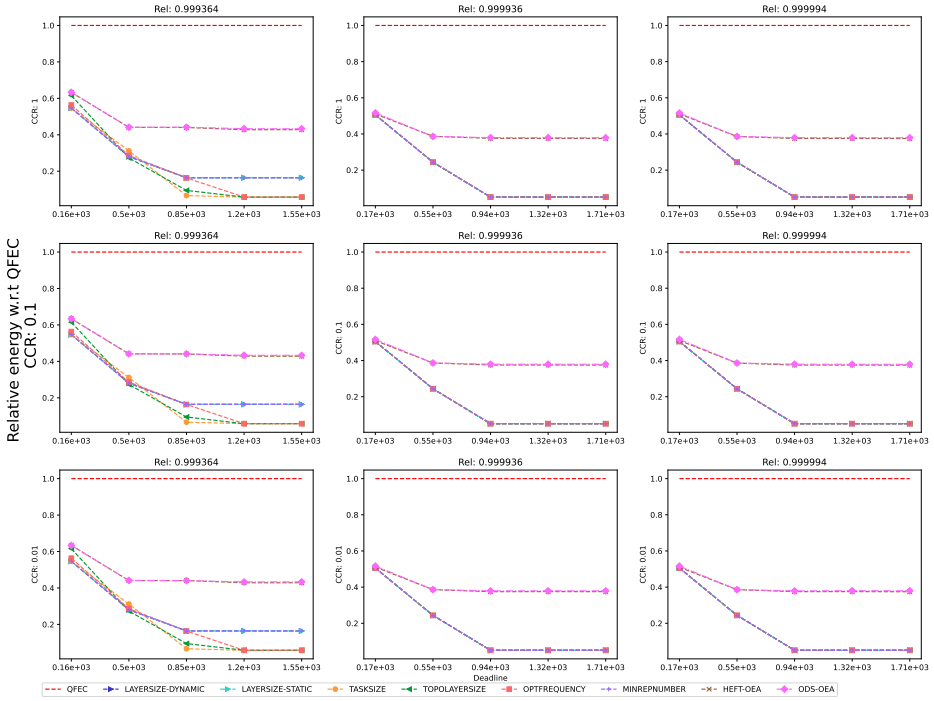

Figure 699: Performance of the different heuristics on the Seismology workflow.

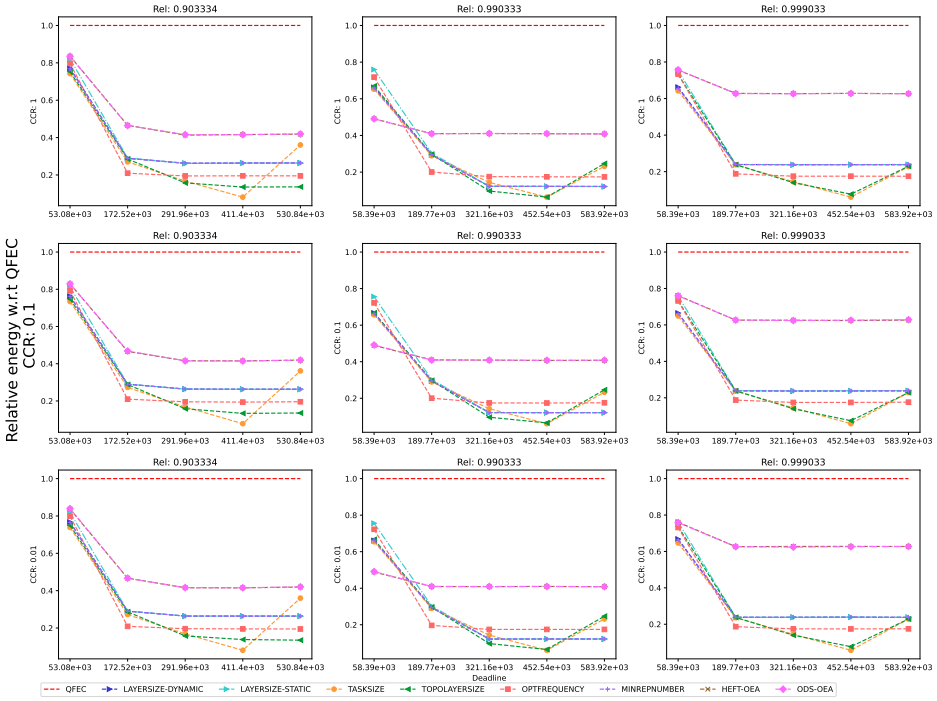

Figure 700: Performance of the different heuristics on the SoyKB workflow.

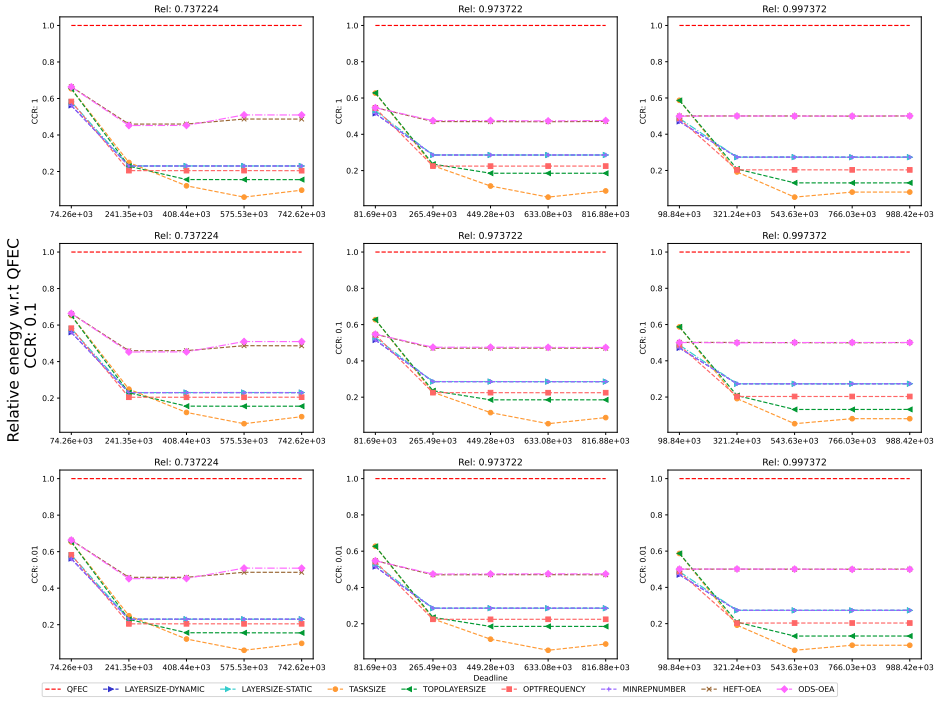

Figure 701: Performance of the different heuristics on the SRASearch.

E.7.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

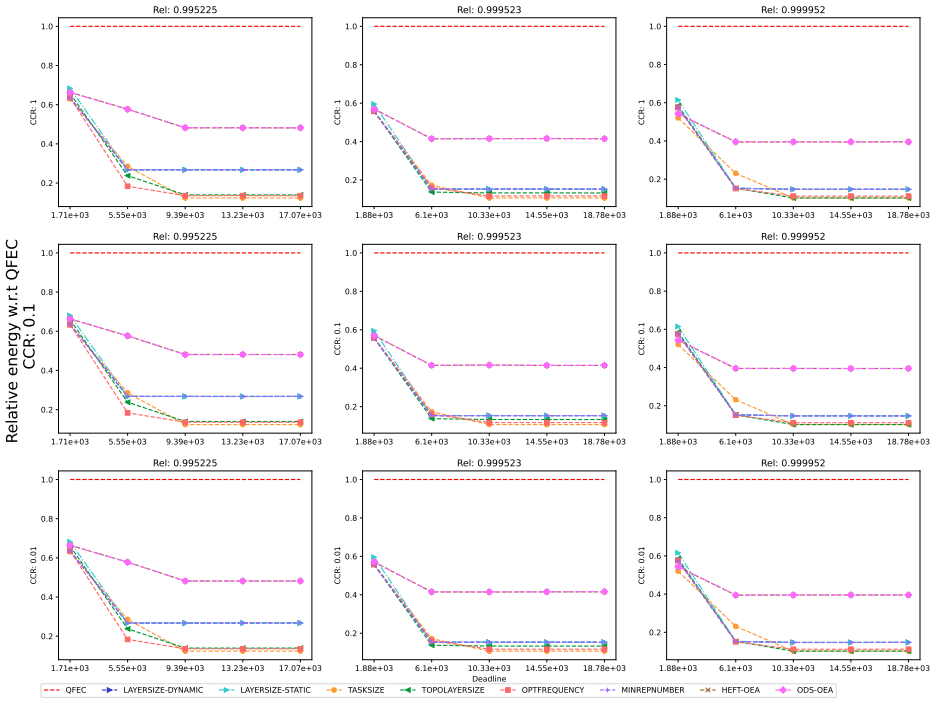

Figure 702: Performance of the different heuristics on the BWA workflow.

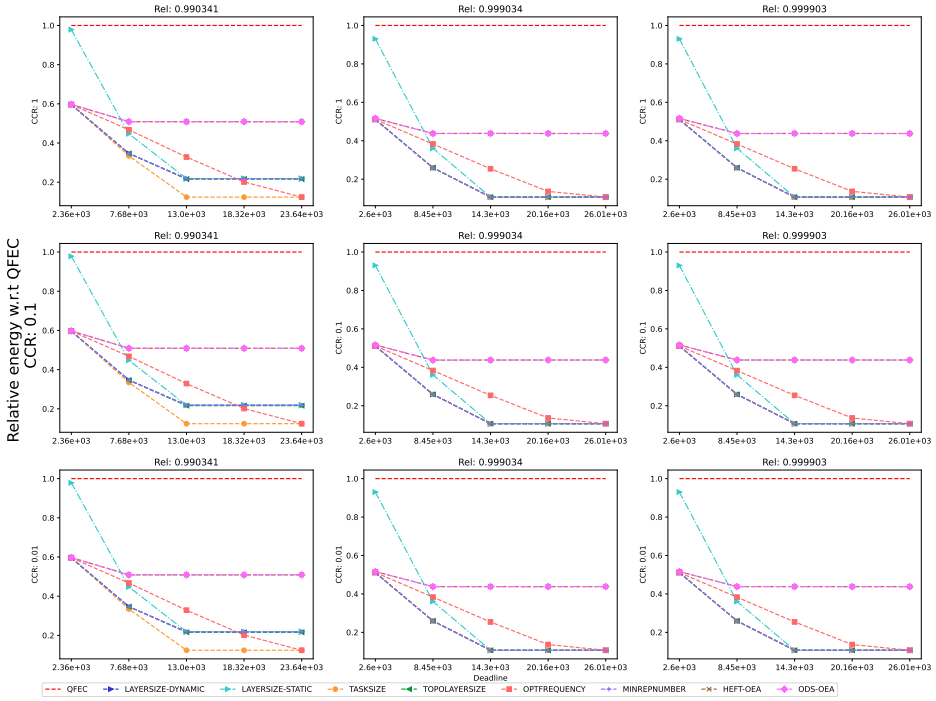

Figure 703: Performance of the different heuristics on the Cholesky workflow.

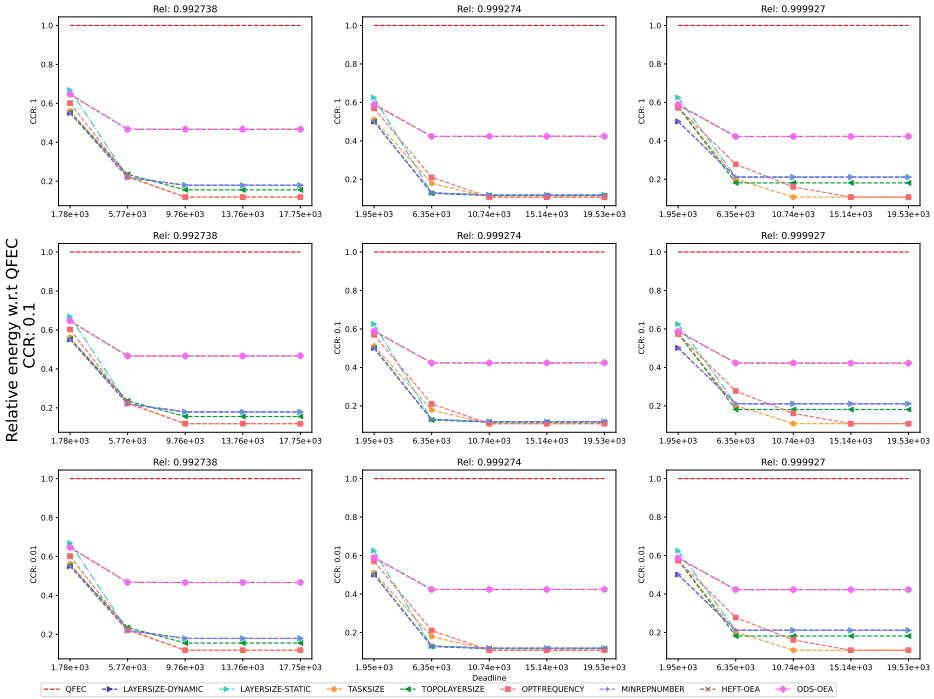

Figure 704: Performance of the different heuristics on the Cycles workflow.

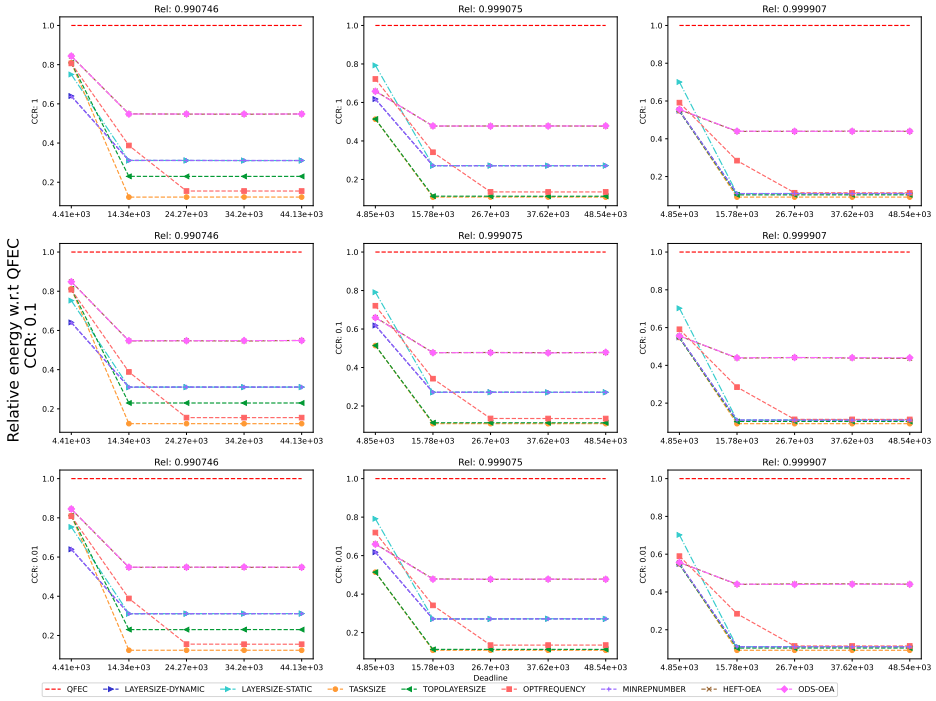

Figure 705: Performance of the different heuristics on the Epigenomics workflow.

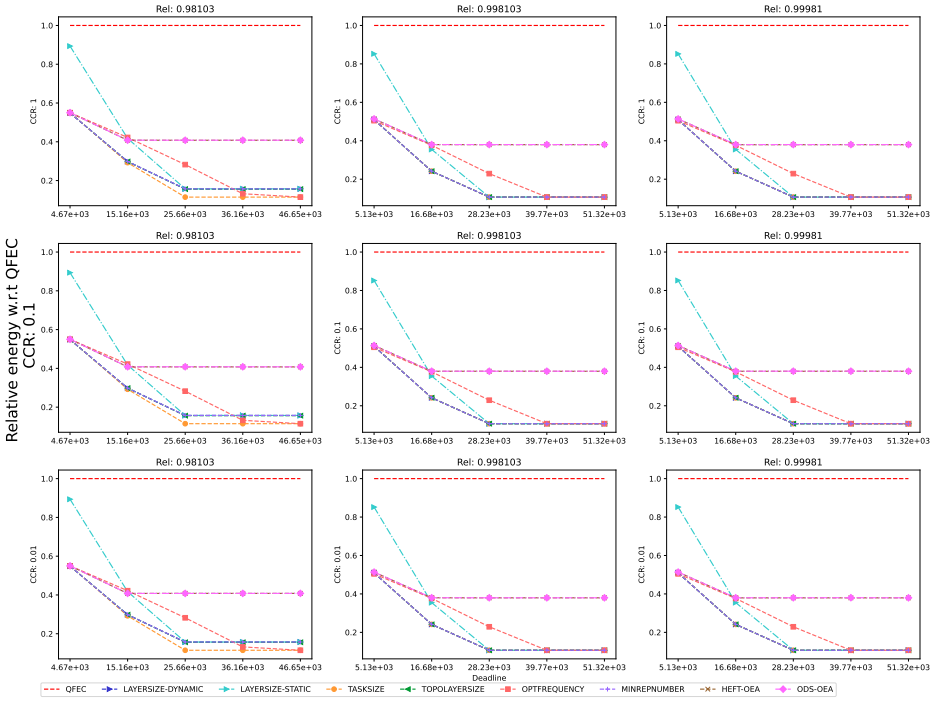

Figure 706: Performance of the different heuristics on the LU workflow.

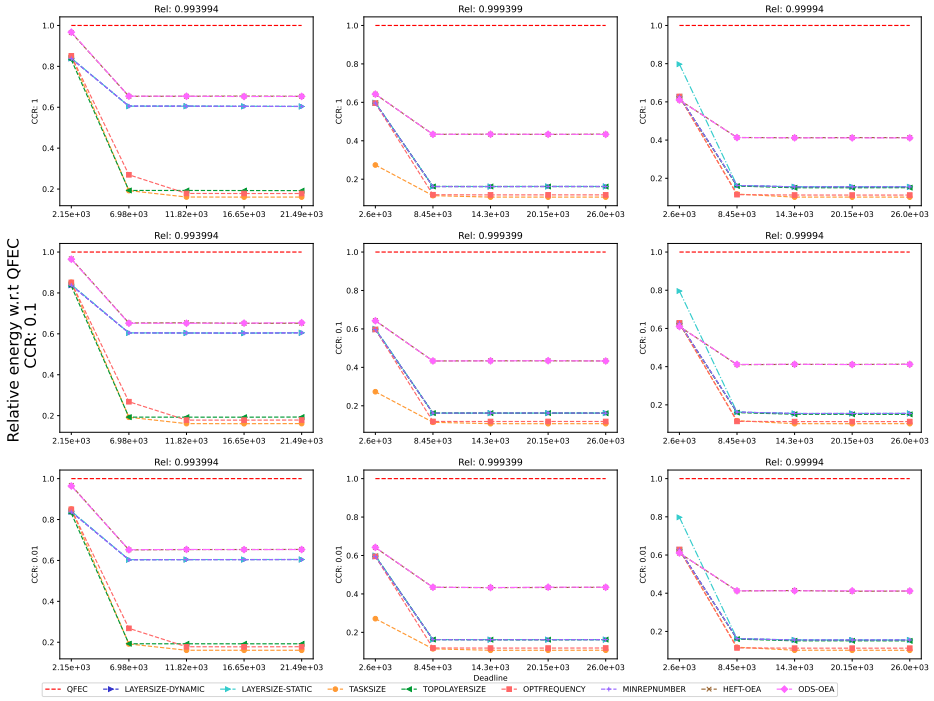

Figure 707: Performance of the different heuristics on the Montage workflow.

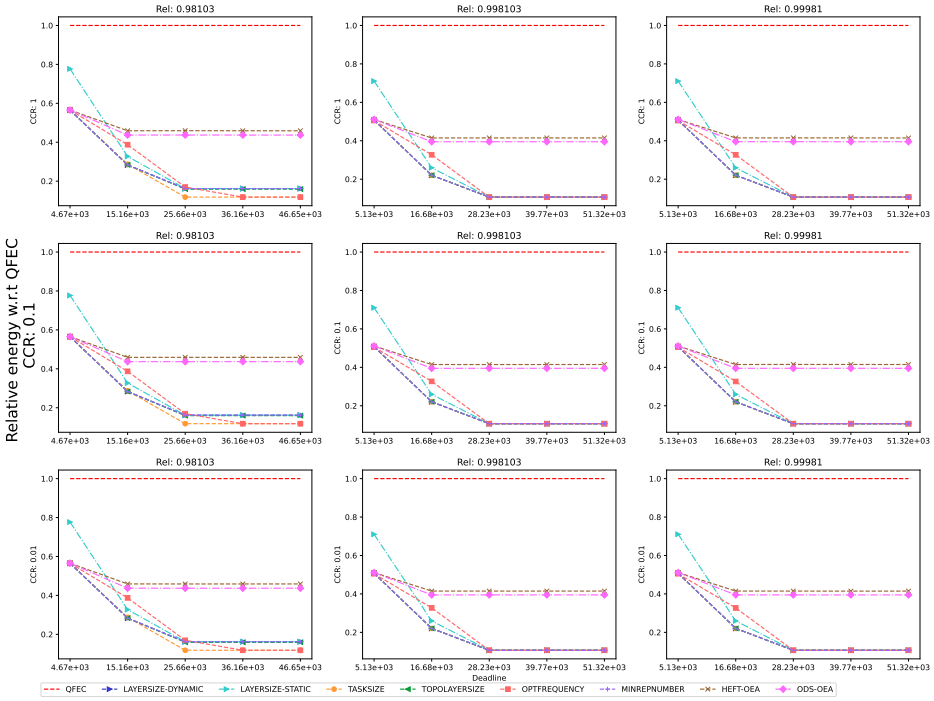

Figure 708: Performance of the different heuristics on the QR workflow.

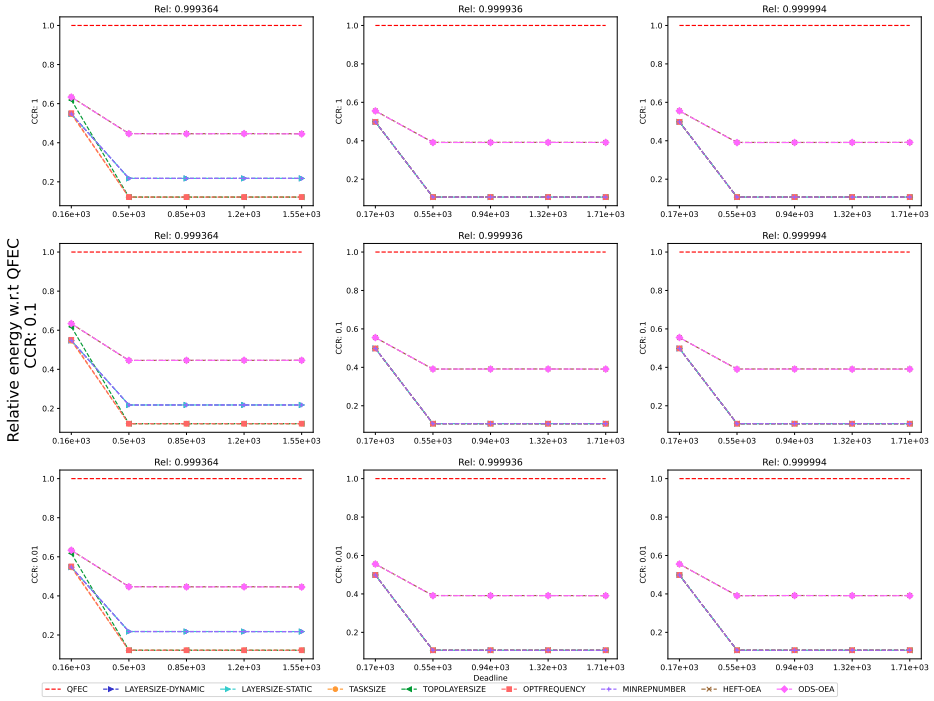

Figure 709: Performance of the different heuristics on the Seismology workflow.

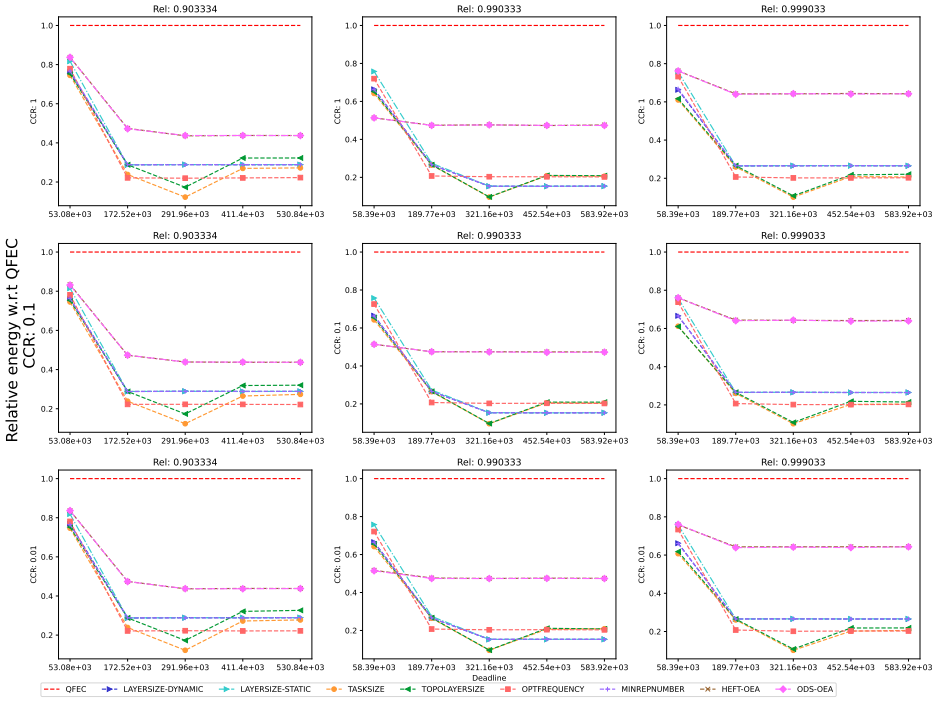

Figure 710: Performance of the different heuristics on the SoyKB workflow.

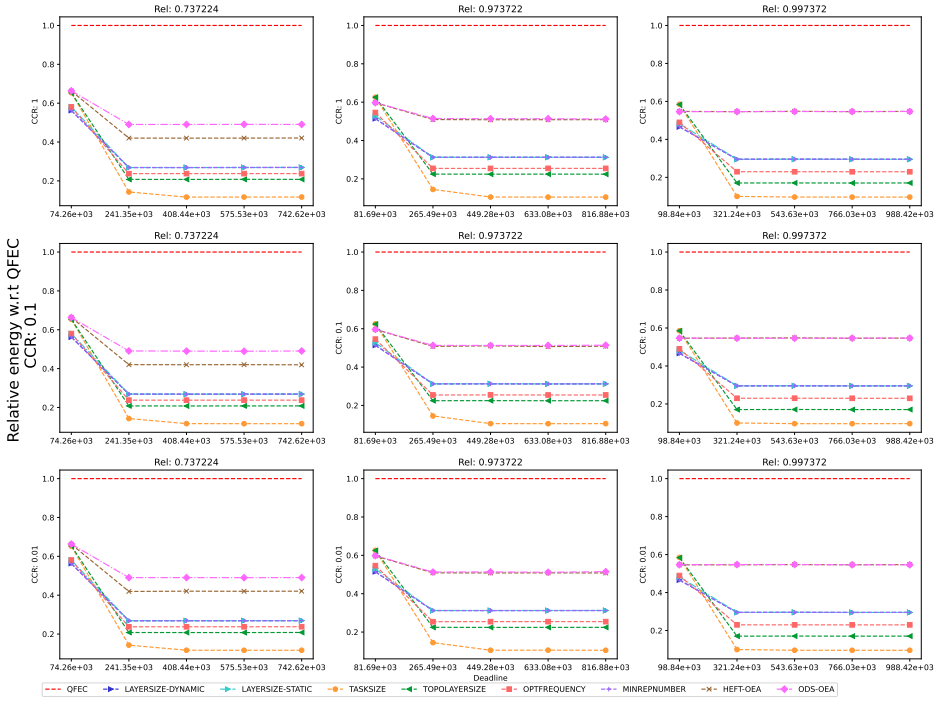

Figure 711: Performance of the different heuristics on the SRASearch.

E.7.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

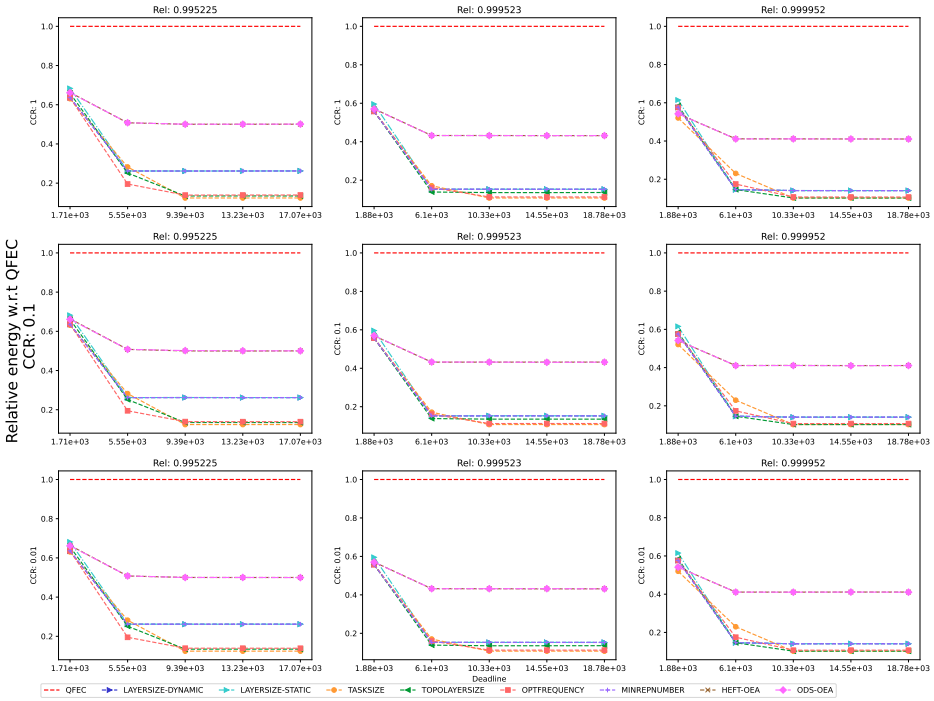

Figure 712: Performance of the different heuristics on the BWA workflow.

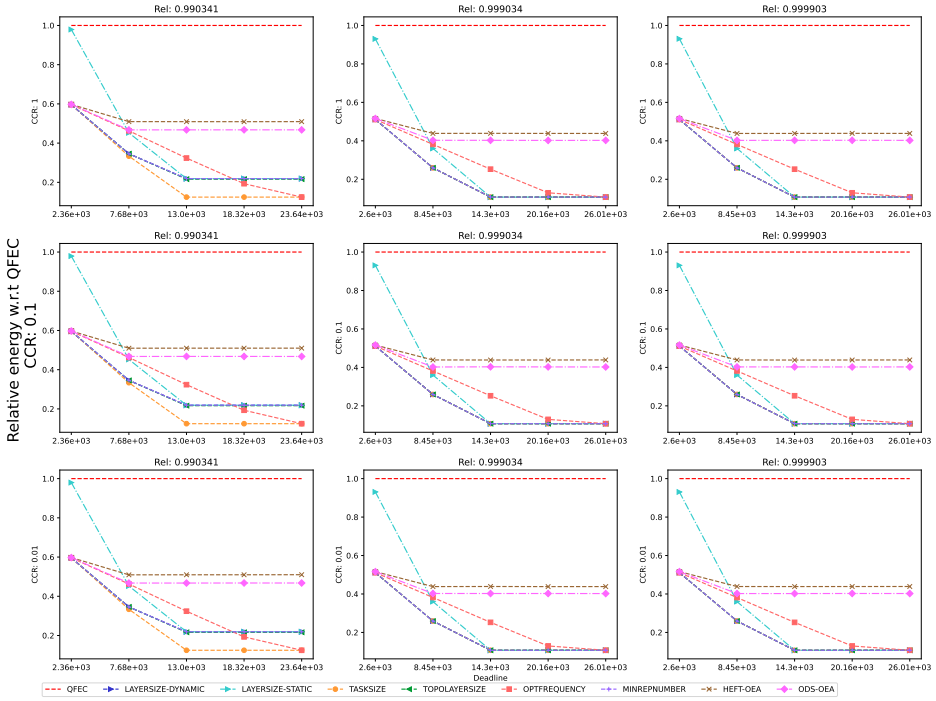

Figure 713: Performance of the different heuristics on the Cholesky workflow.

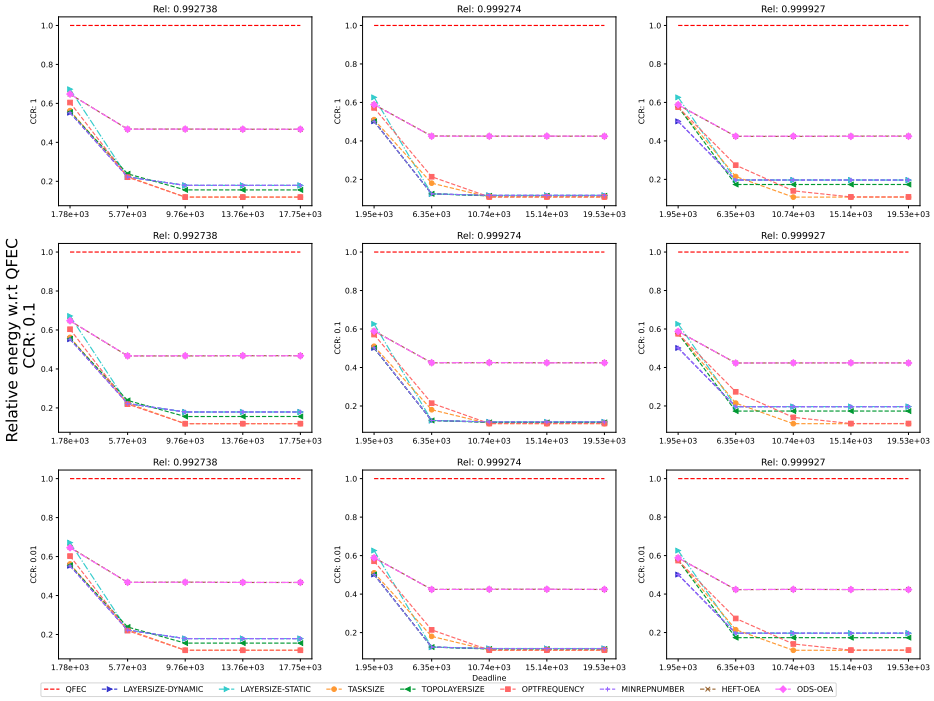

Figure 714: Performance of the different heuristics on the Cycles workflow.

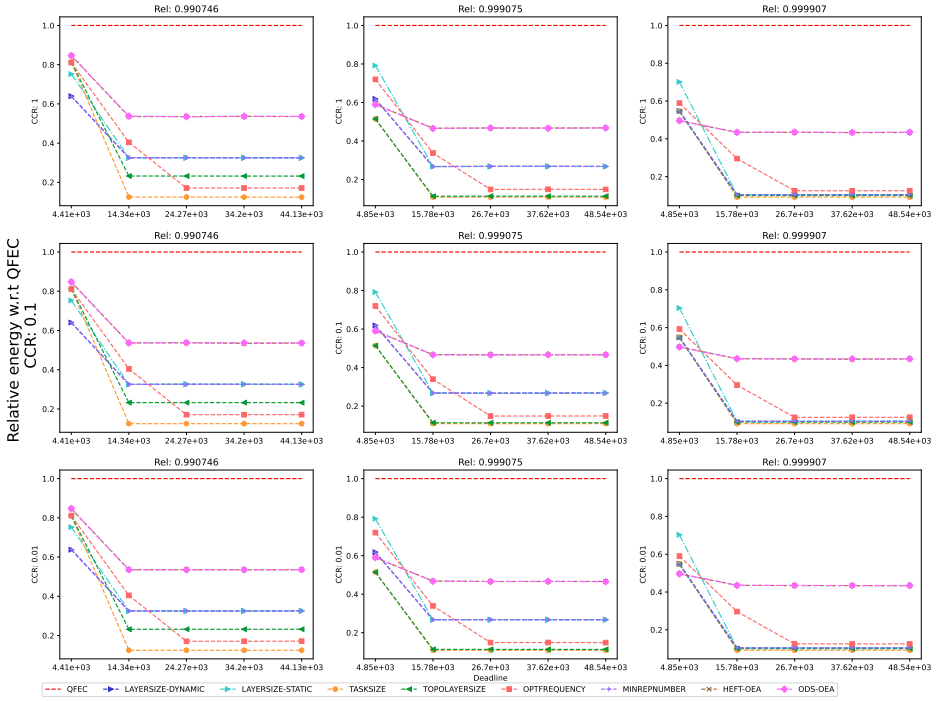

Figure 715: Performance of the different heuristics on the Epigenomics workflow.

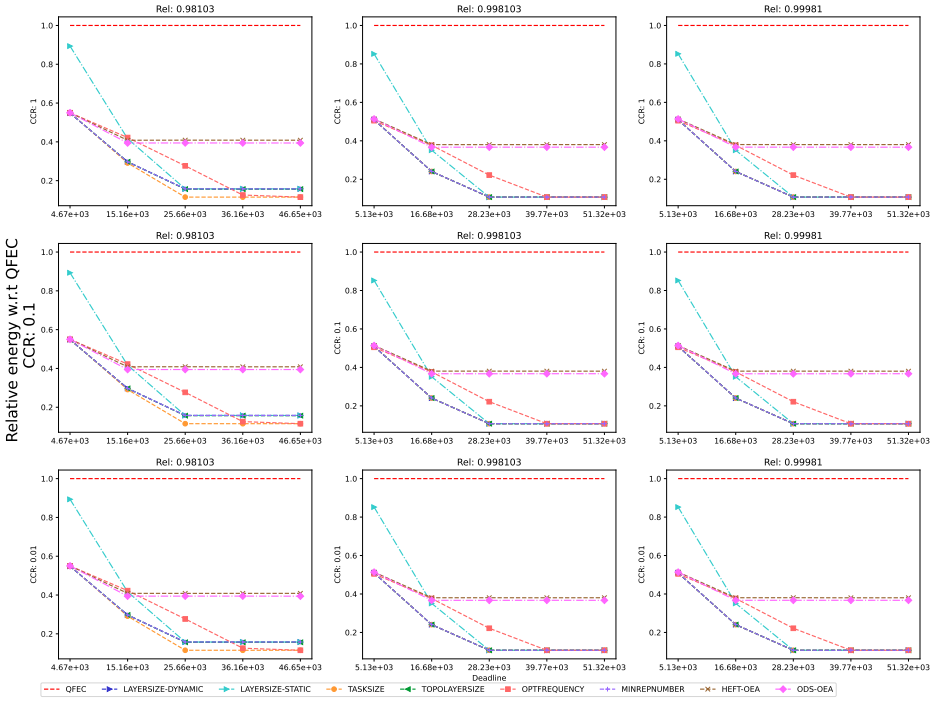

Figure 716: Performance of the different heuristics on the LU workflow.

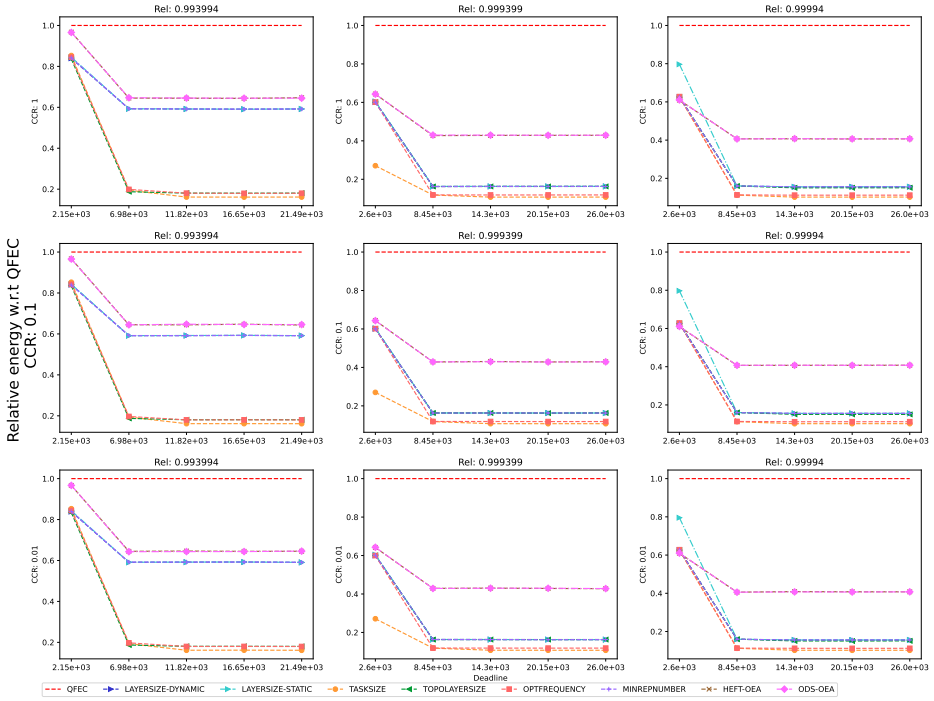

Figure 717: Performance of the different heuristics on the Montage workflow.

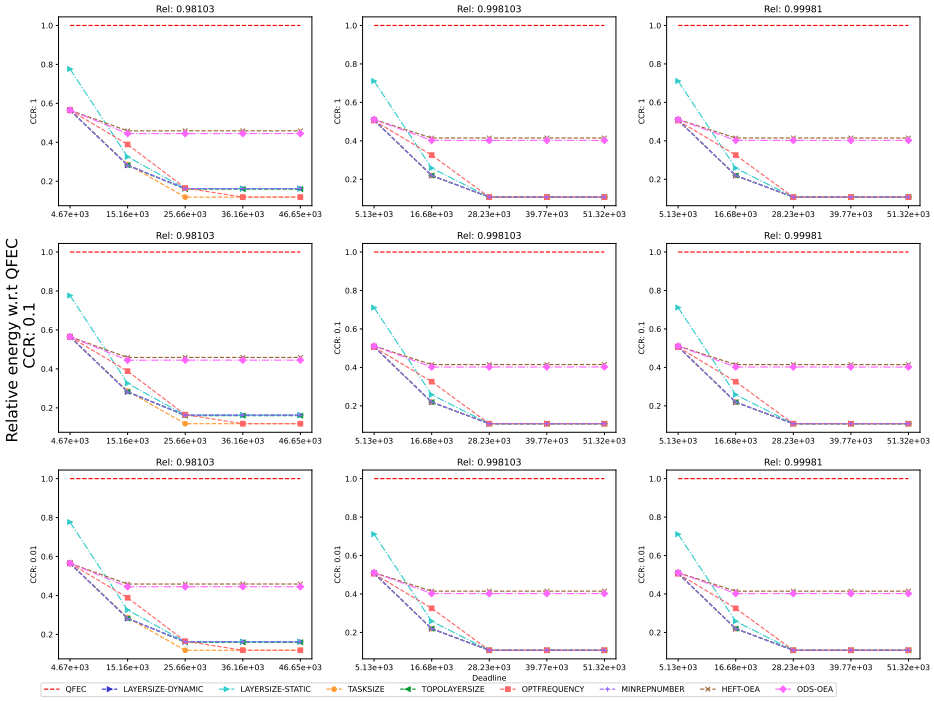

Figure 718: Performance of the different heuristics on the QR workflow.

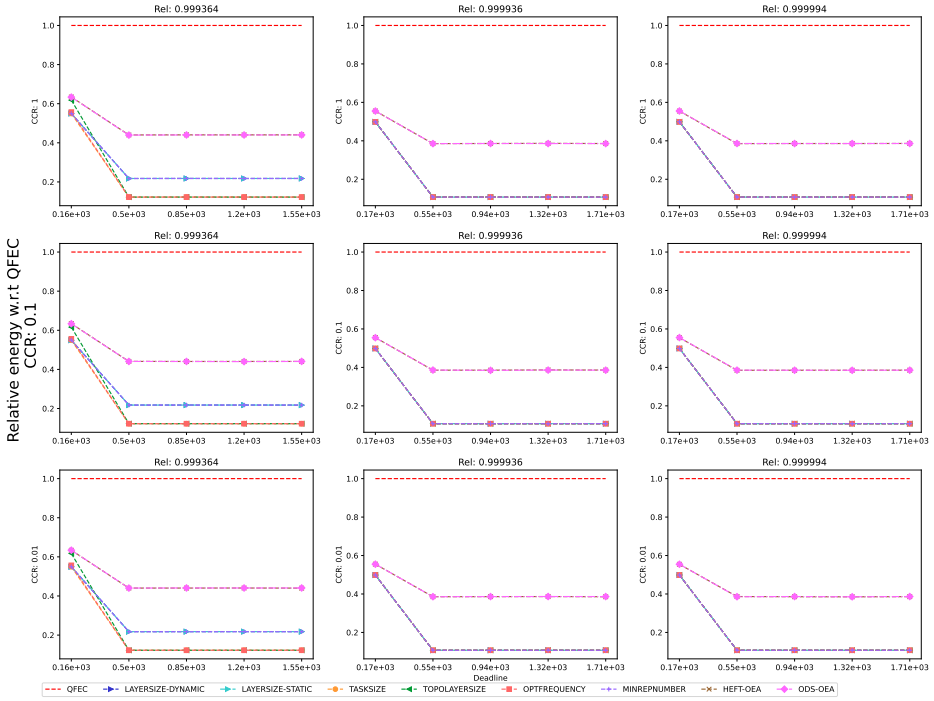

Figure 719: Performance of the different heuristics on the Seismology workflow.

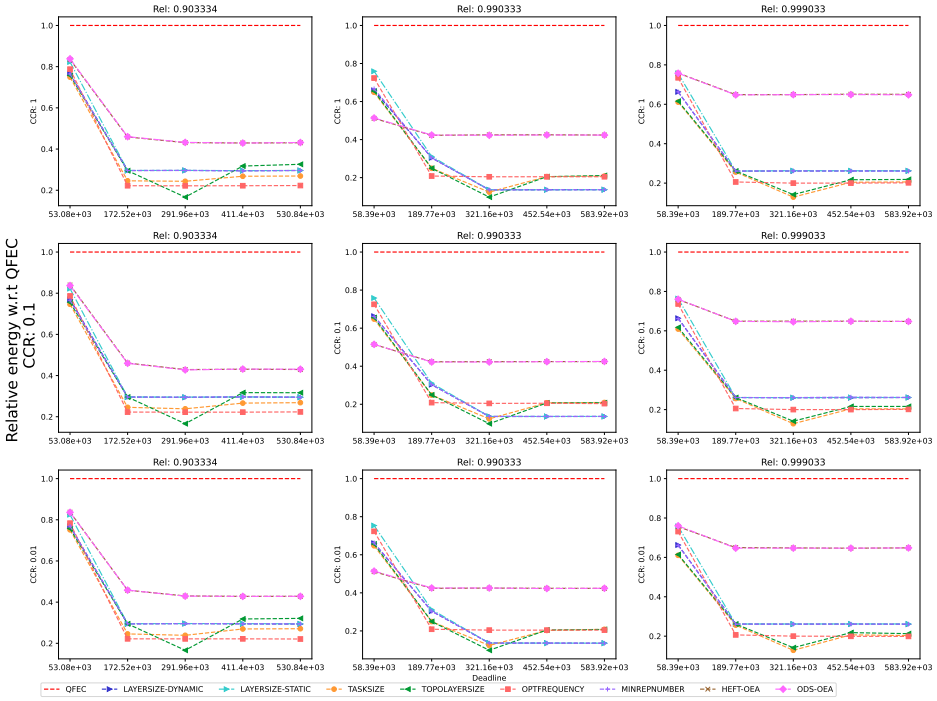

Figure 720: Performance of the different heuristics on the SoyKB workflow.

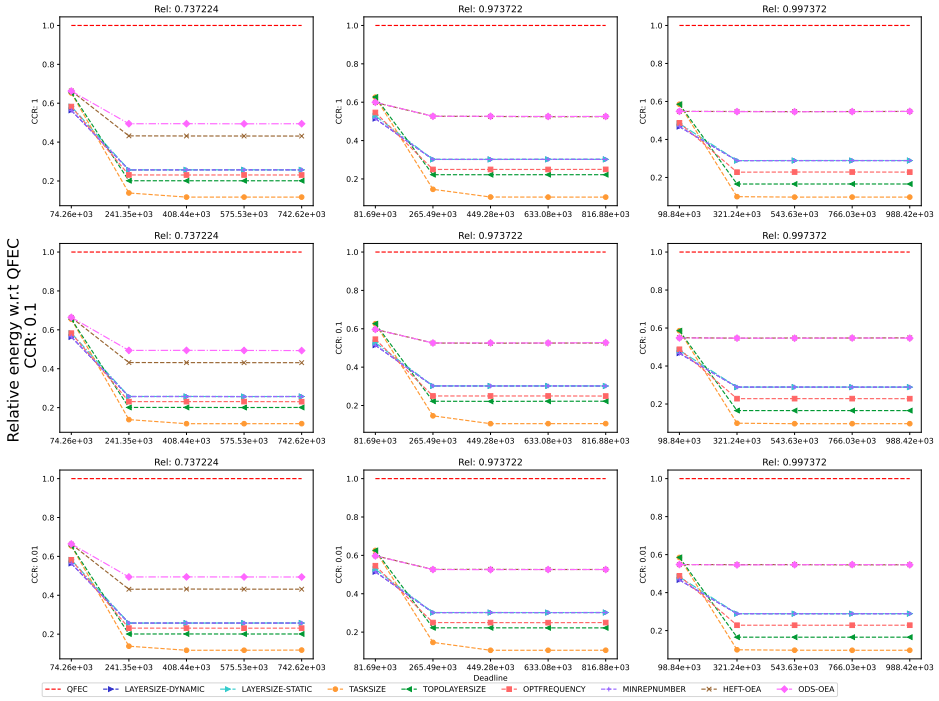

Figure 721: Performance of the different heuristics on the SRASearch.

E.8 $BC/WC = 0.8$

E.8.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

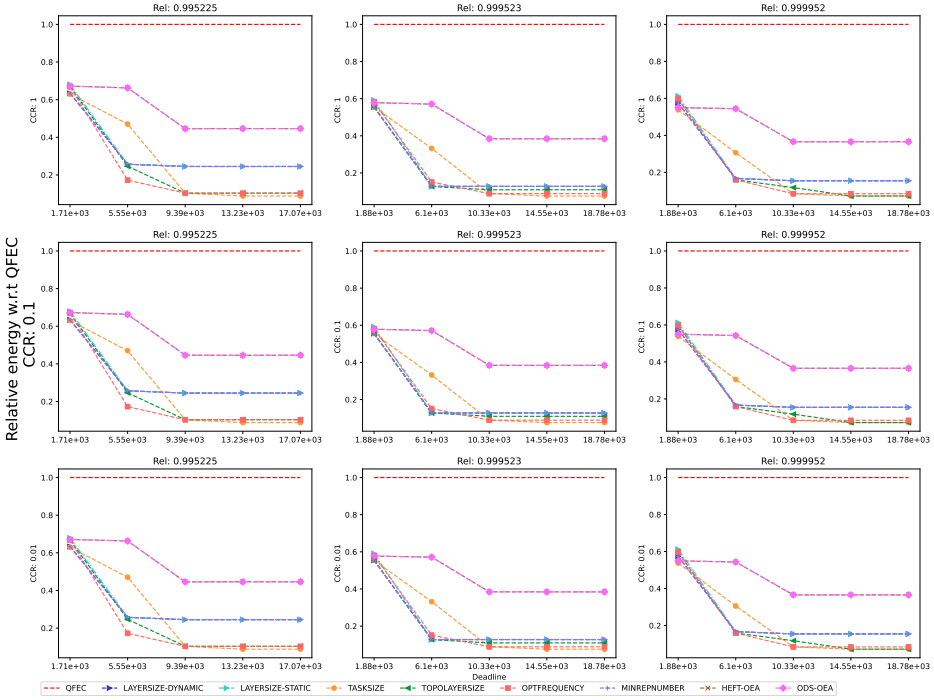

Figure 722: Performance of the different heuristics on the BWA workflow.

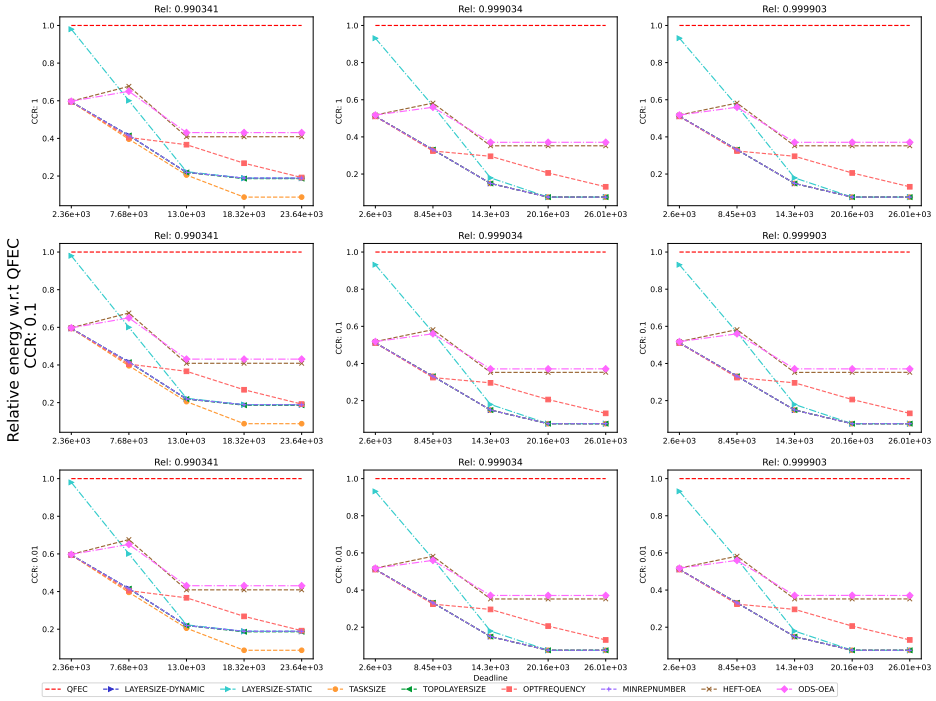

Figure 723: Performance of the different heuristics on the Cholesky workflow.

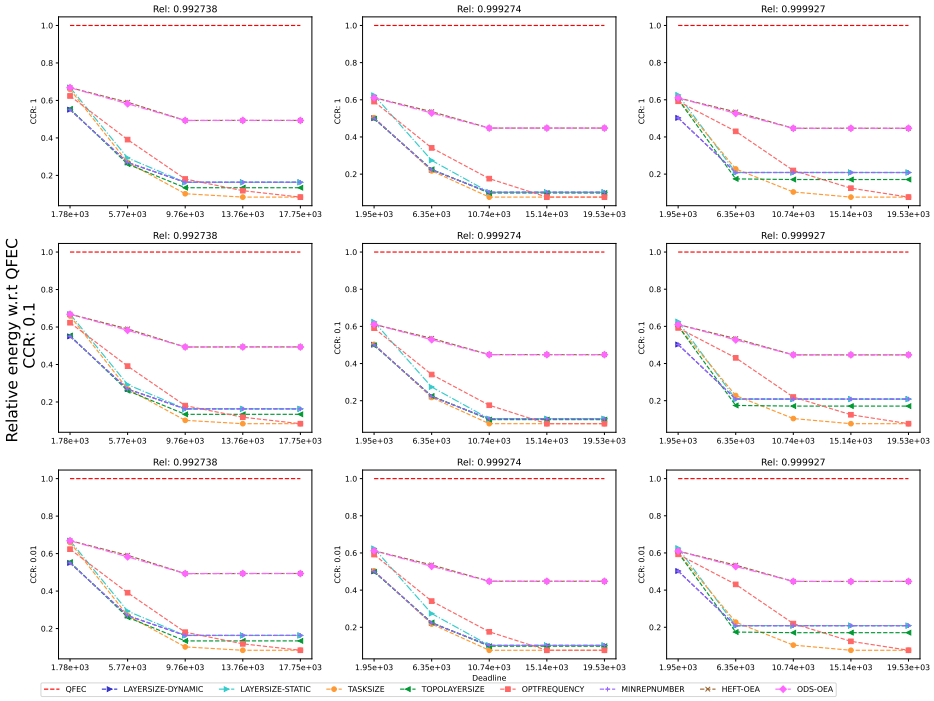

Figure 724: Performance of the different heuristics on the Cycles workflow.

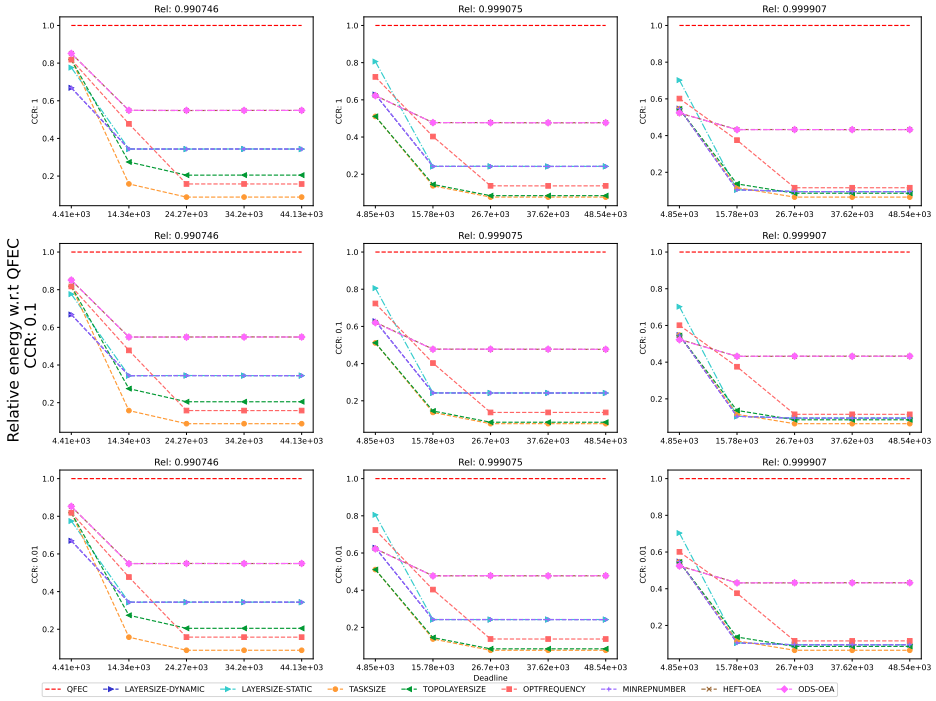

Figure 725: Performance of the different heuristics on the Epigenomics workflow.

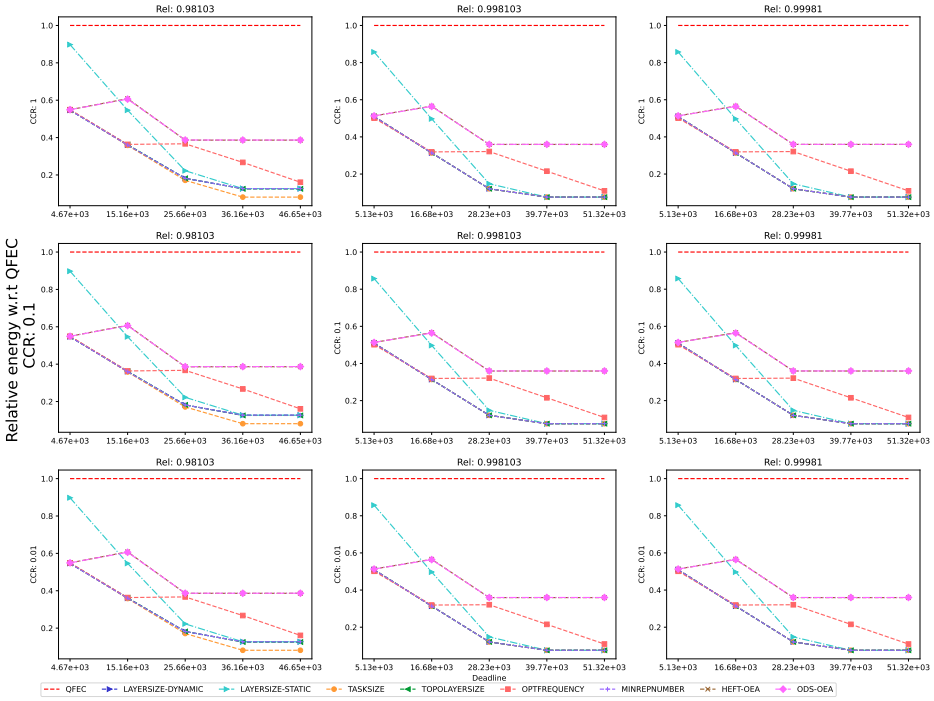

Figure 726: Performance of the different heuristics on the LU workflow.

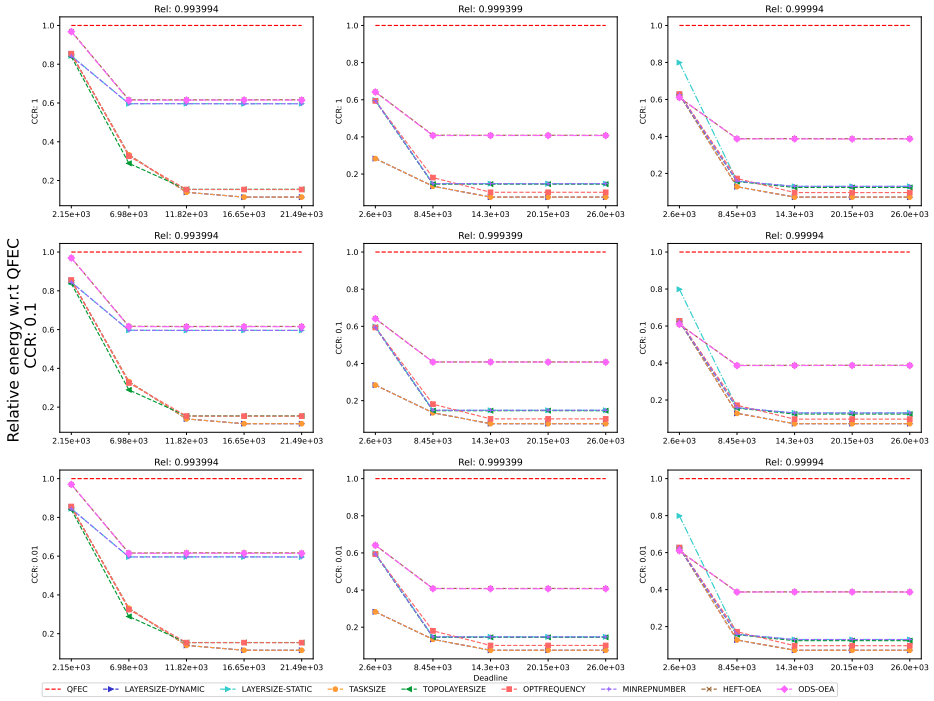

Figure 727: Performance of the different heuristics on the Montage workflow.

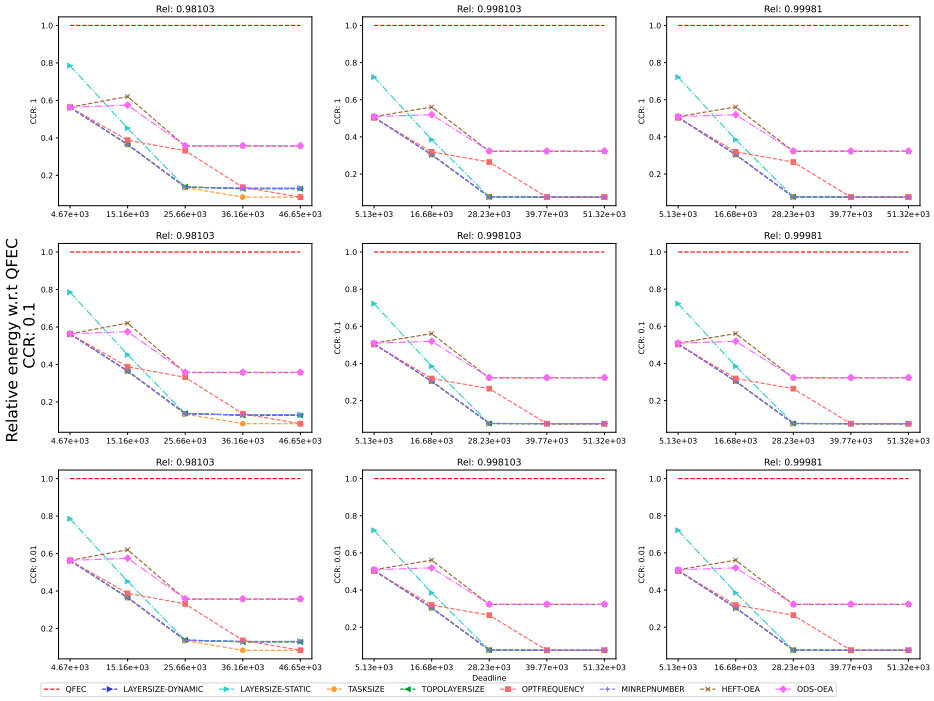

Figure 728: Performance of the different heuristics on the QR workflow.

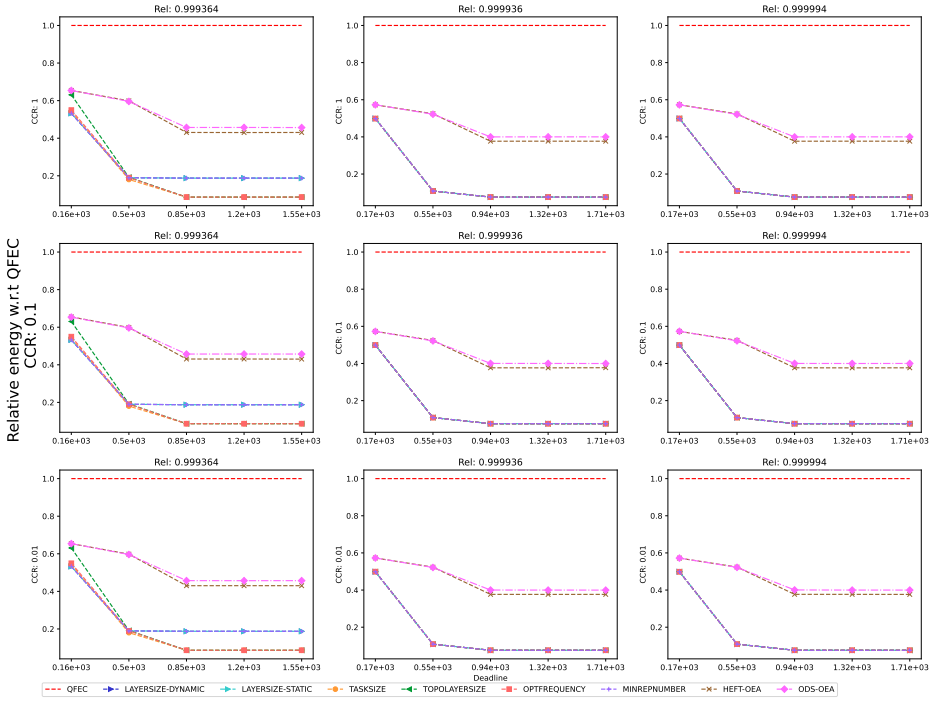

Figure 729: Performance of the different heuristics on the Seismology workflow.

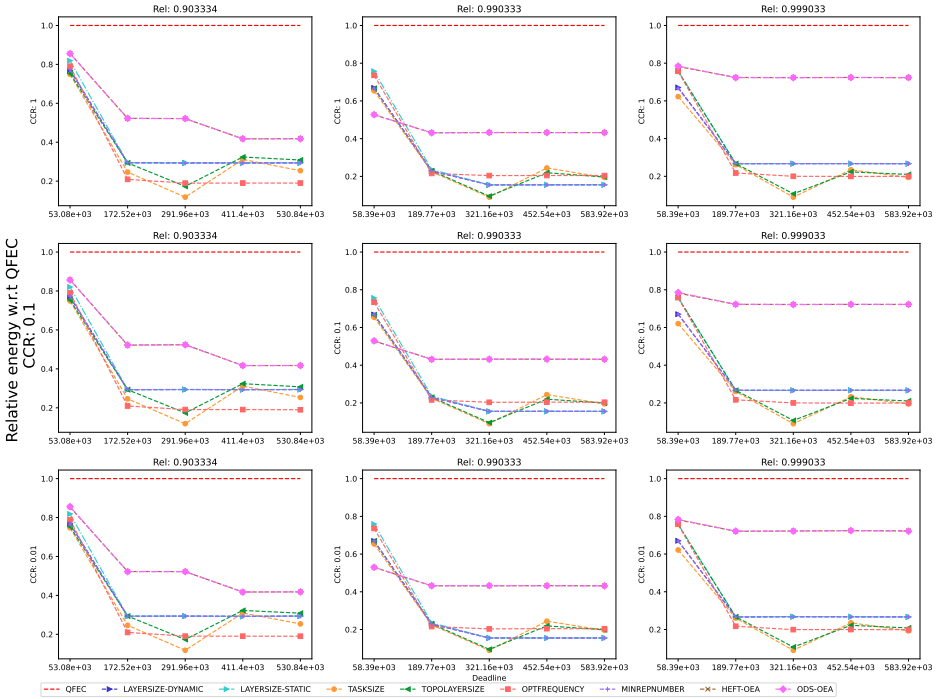

Figure 730: Performance of the different heuristics on the SoyKB workflow.

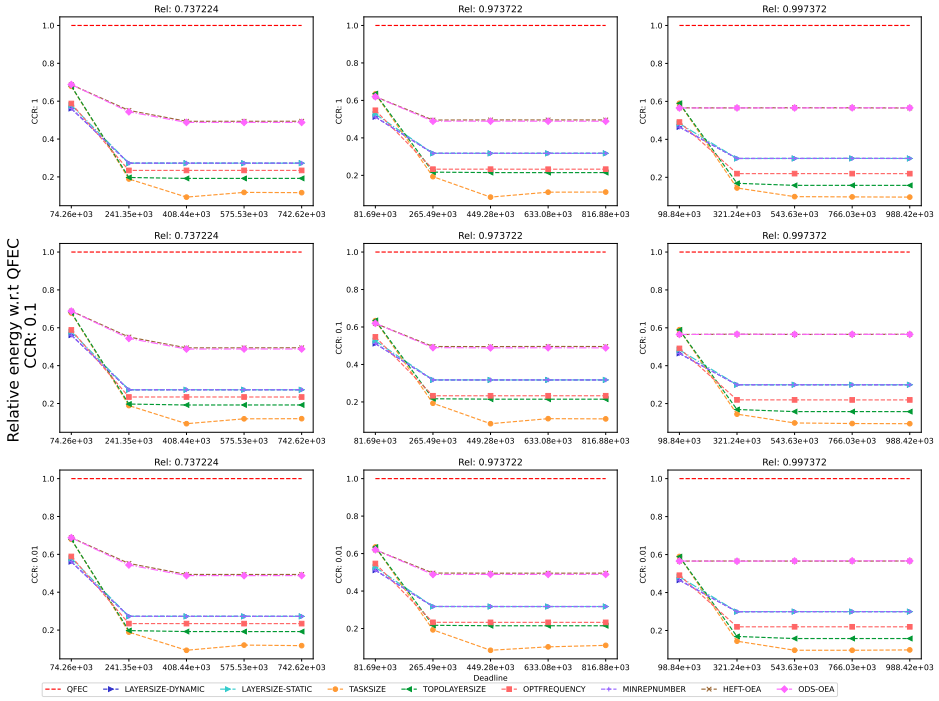

Figure 731: Performance of the different heuristics on the SRASearch.

E.8.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

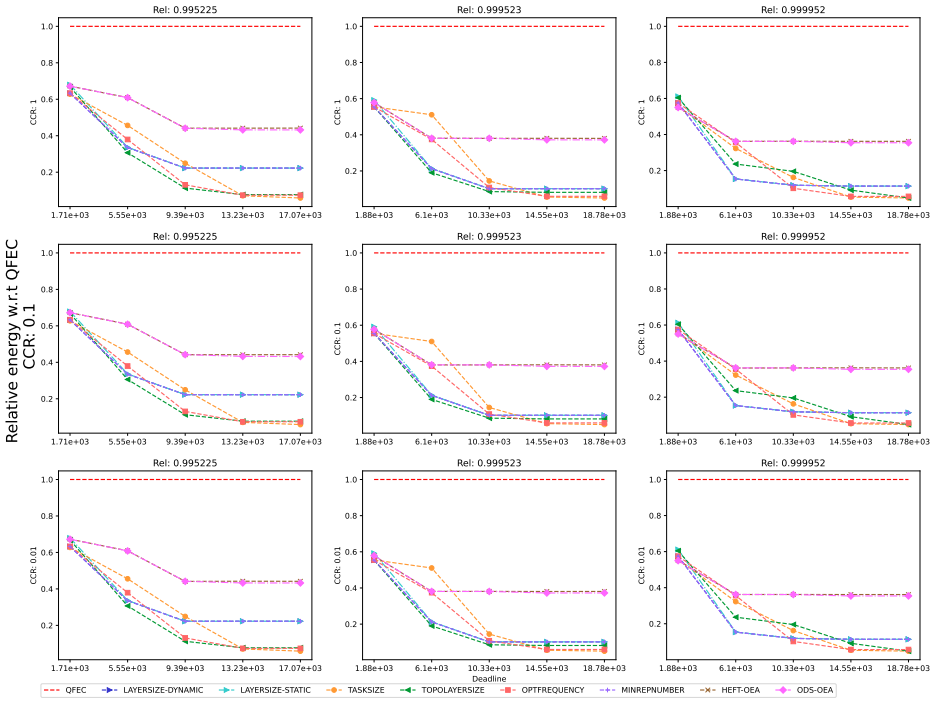

Figure 732: Performance of the different heuristics on the BWA workflow.

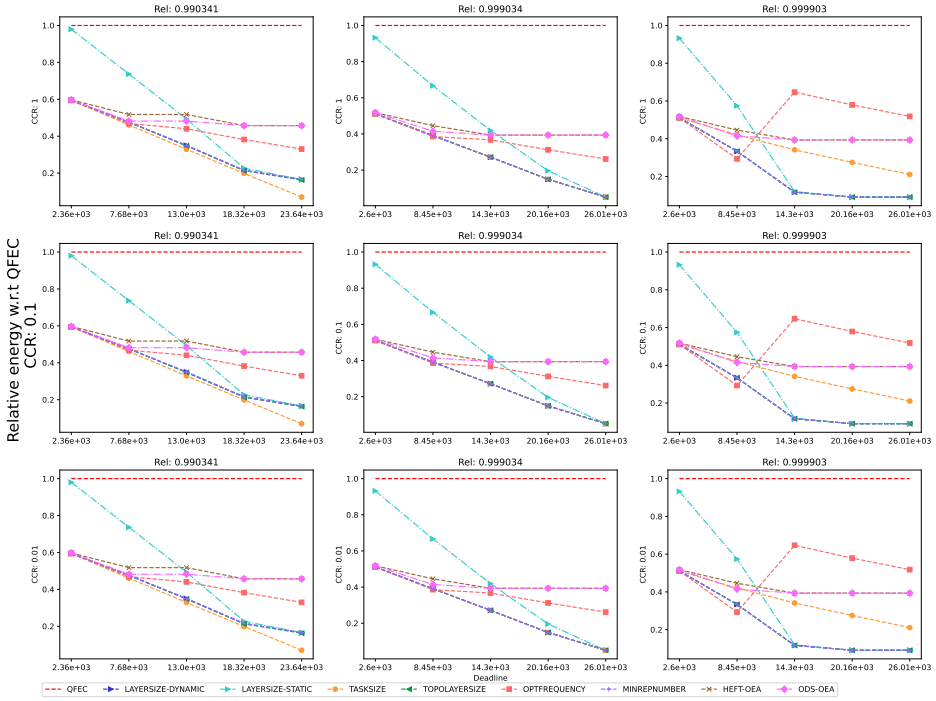

Figure 733: Performance of the different heuristics on the Cholesky workflow.

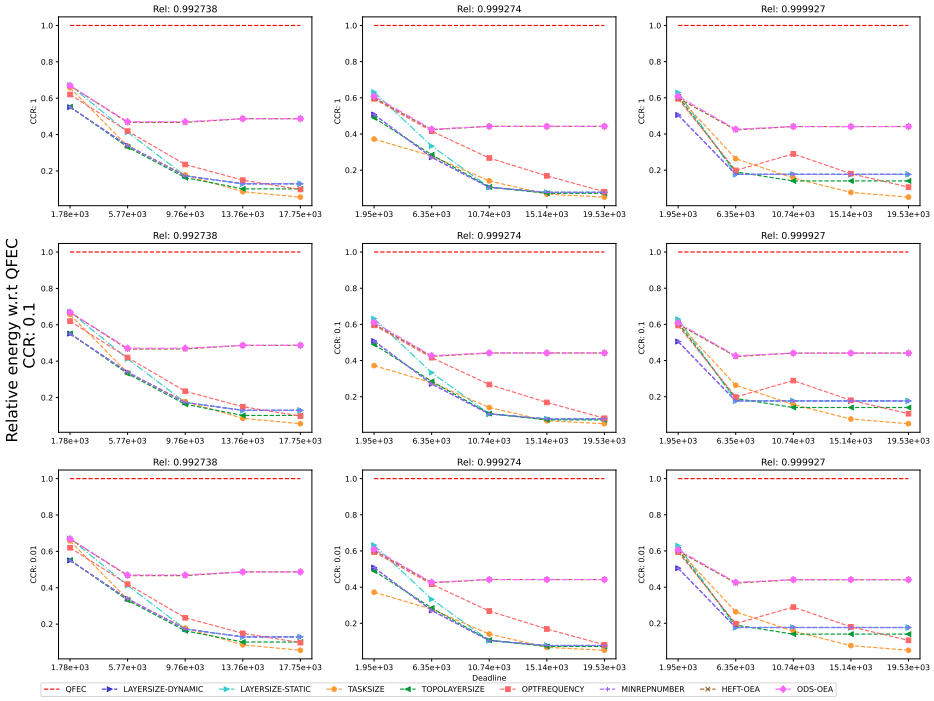

Figure 734: Performance of the different heuristics on the Cycles workflow.

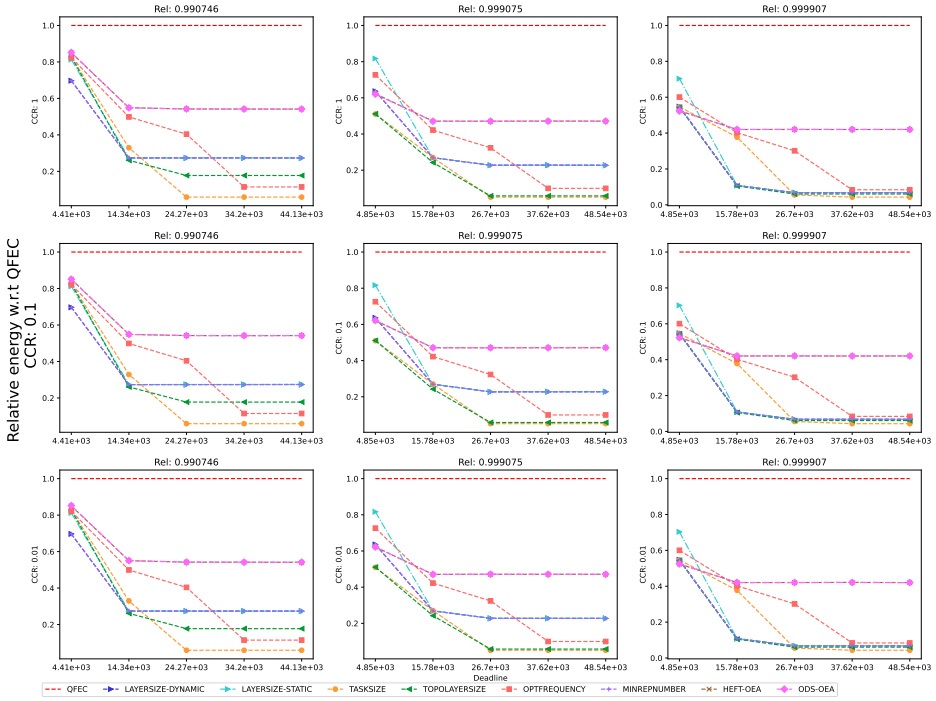

Figure 735: Performance of the different heuristics on the Epigenomics workflow.

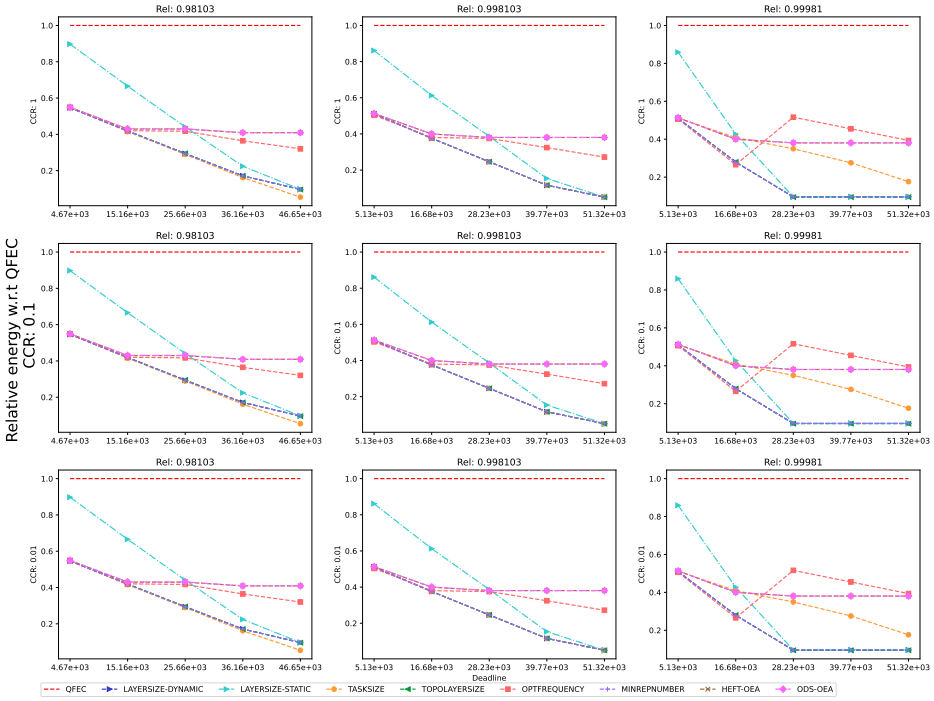

Figure 736: Performance of the different heuristics on the LU workflow.

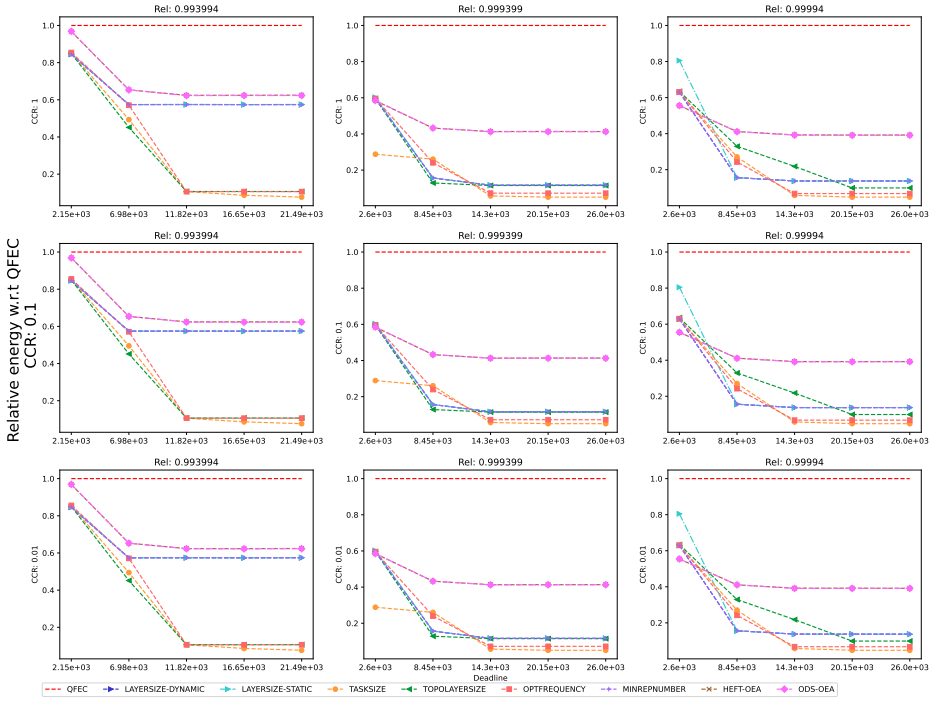

Figure 737: Performance of the different heuristics on the Montage workflow.

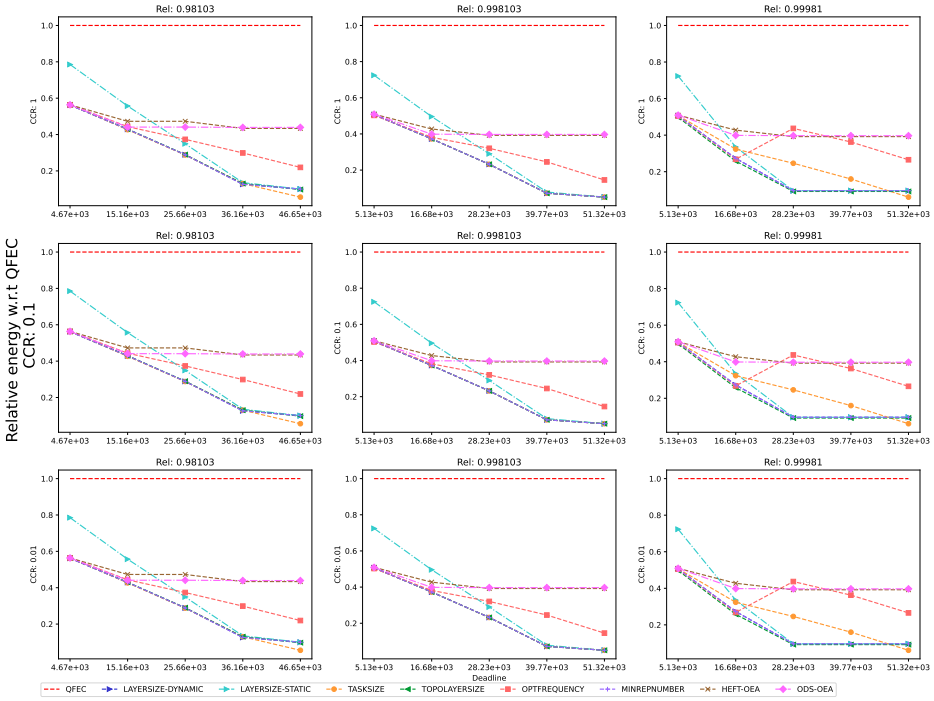

Figure 738: Performance of the different heuristics on the QR workflow.

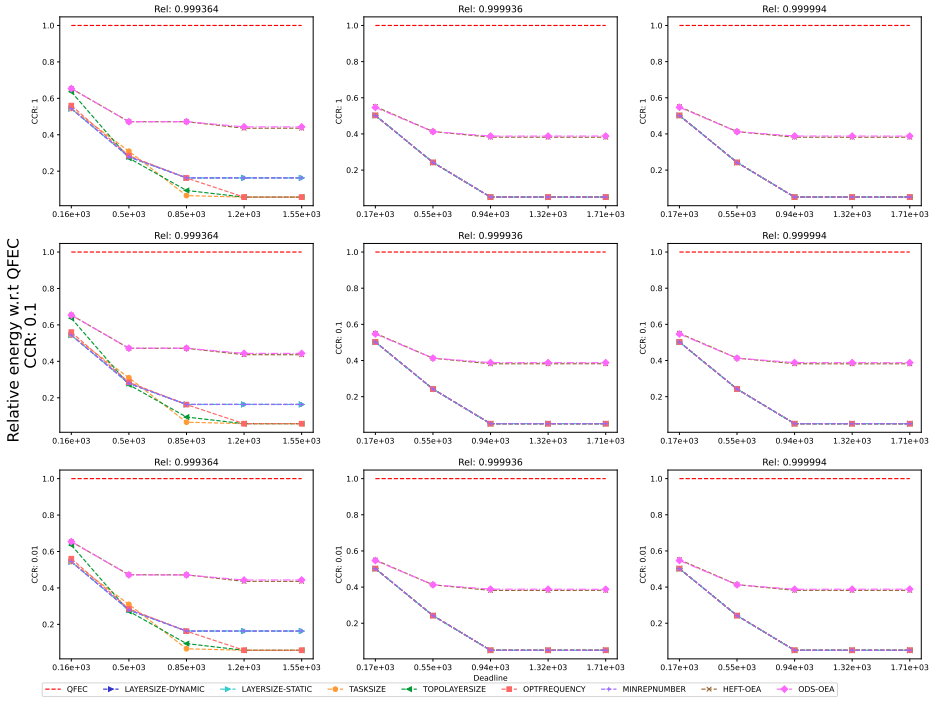

Figure 739: Performance of the different heuristics on the Seismology workflow.

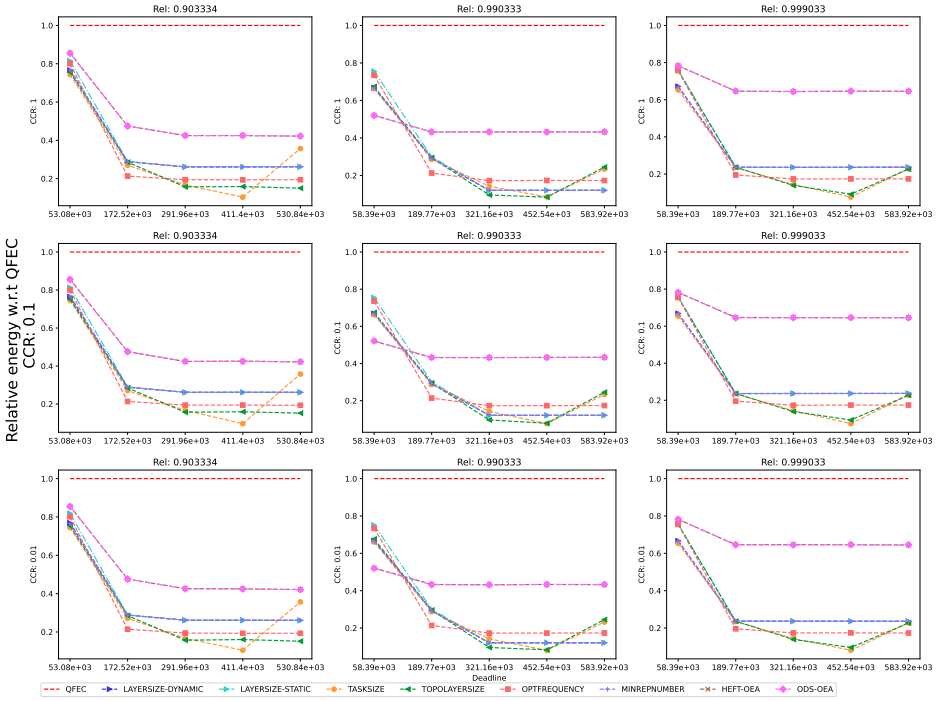

Figure 740: Performance of the different heuristics on the SoyKB workflow.

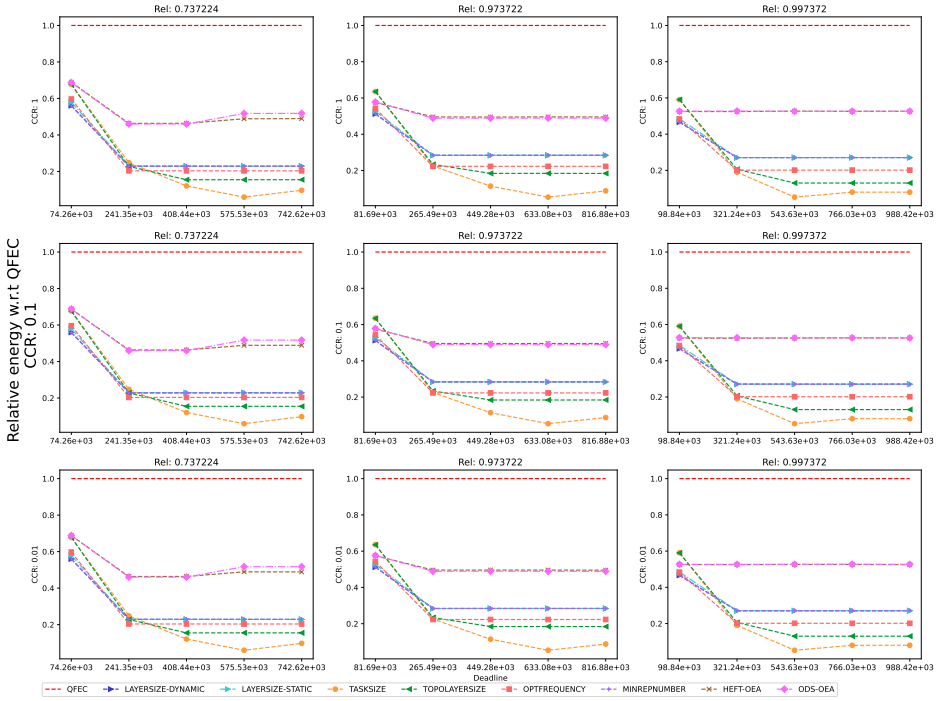

Figure 741: Performance of the different heuristics on the SRASearch.

E.8.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

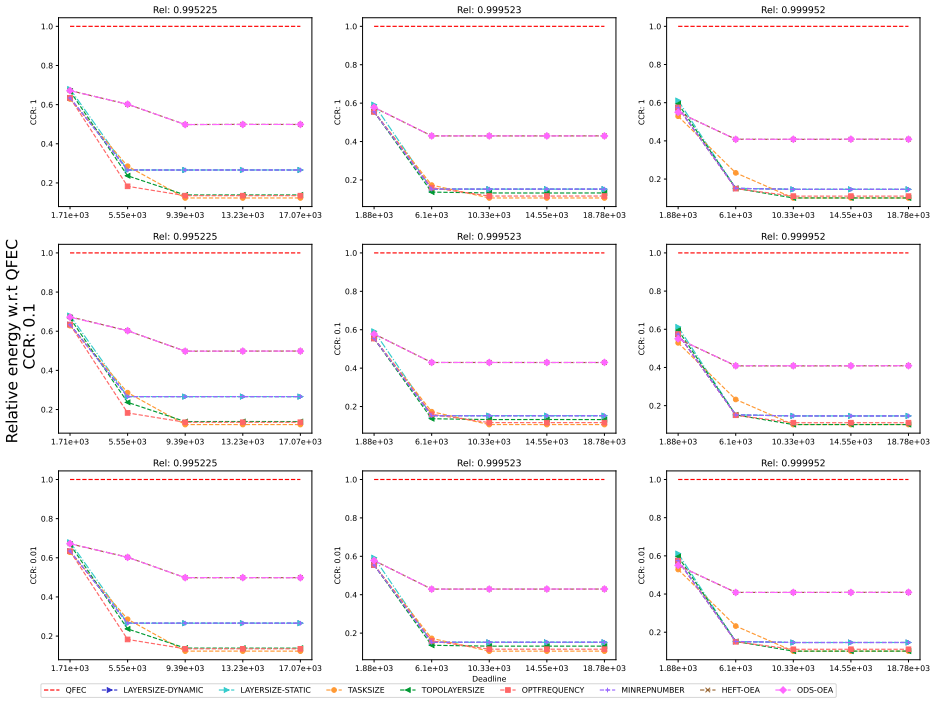

Figure 742: Performance of the different heuristics on the BWA workflow.

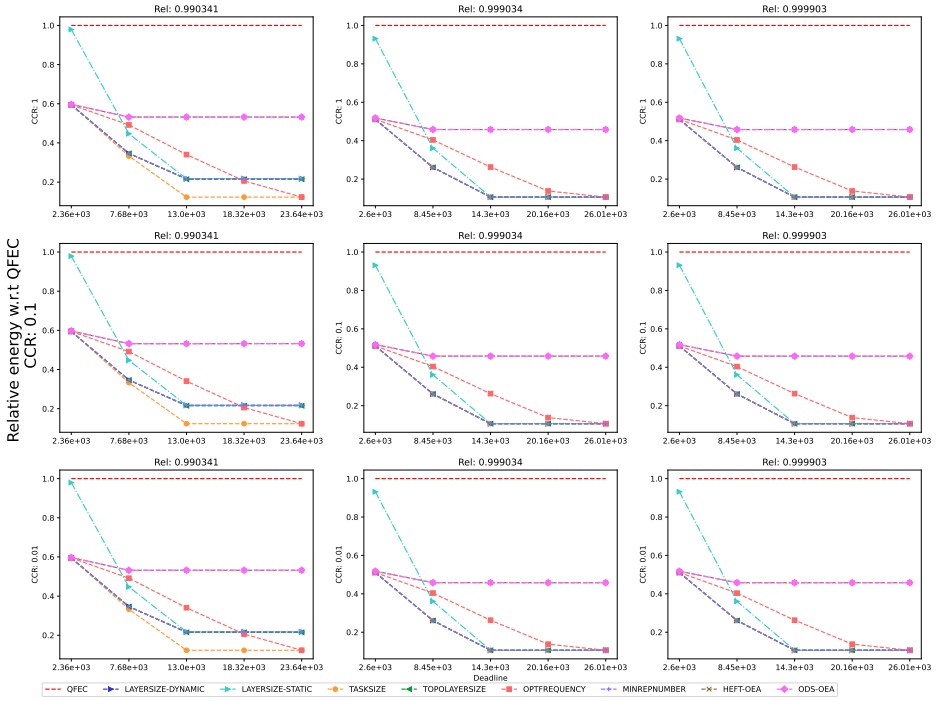

Figure 743: Performance of the different heuristics on the Cholesky workflow.

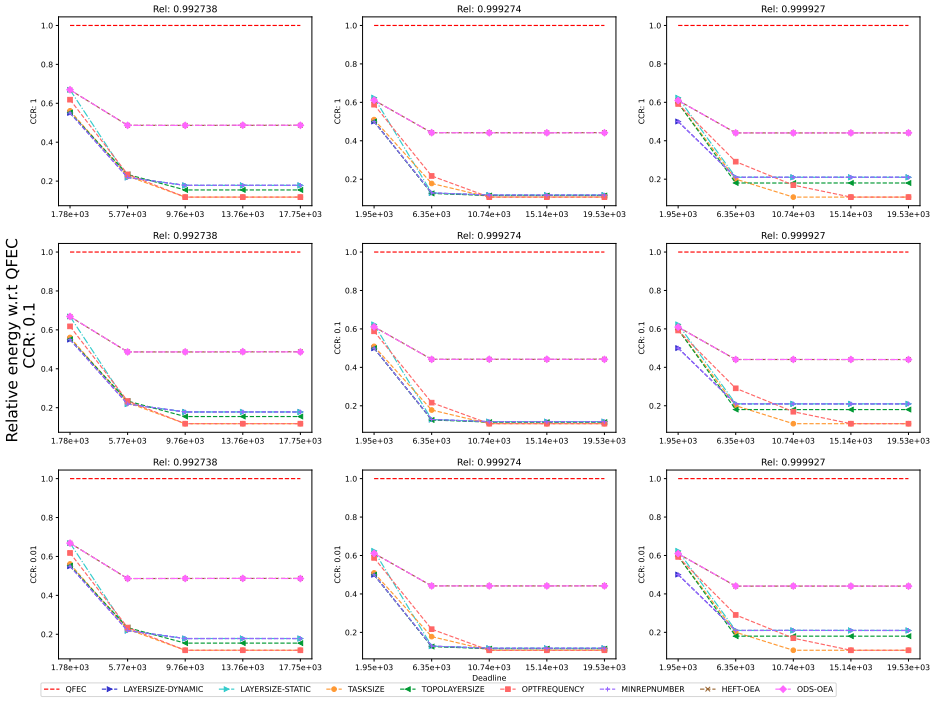

Figure 744: Performance of the different heuristics on the Cycles workflow.

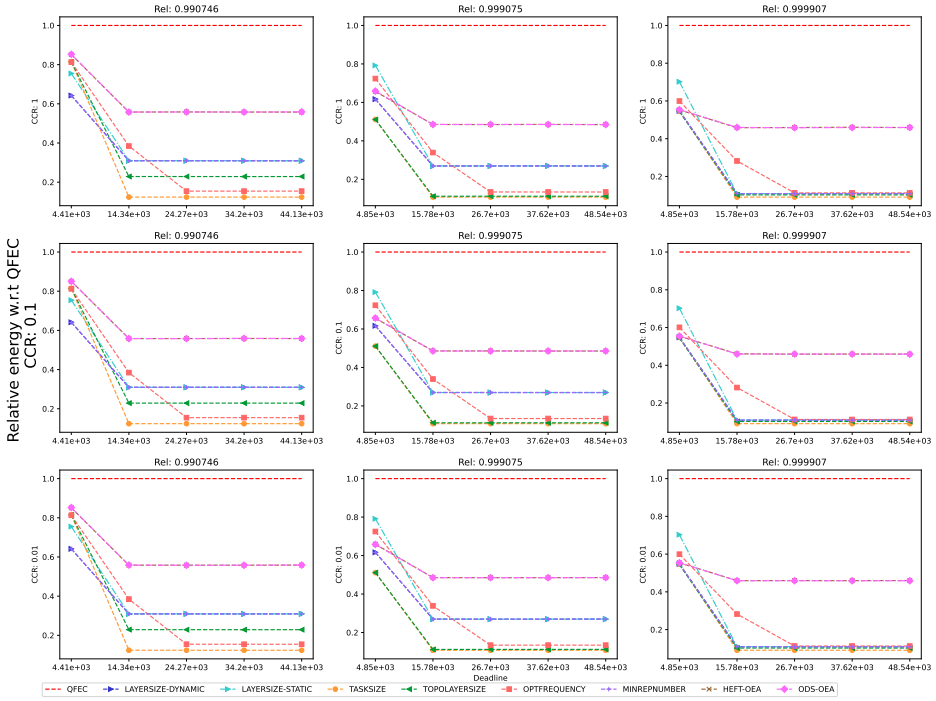

Figure 745: Performance of the different heuristics on the Epigenomics workflow.

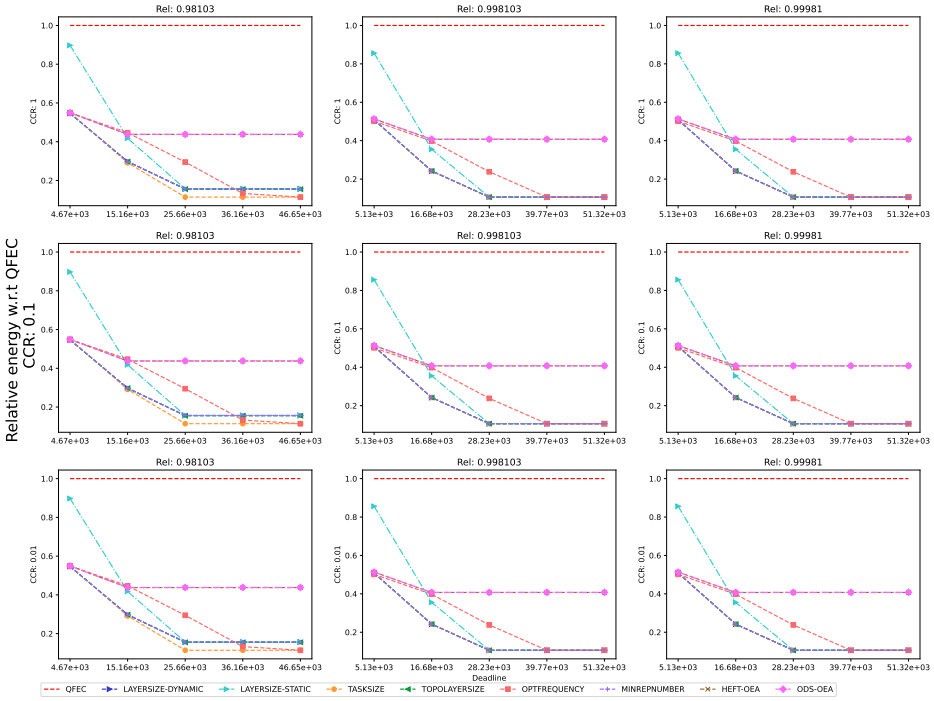

Figure 746: Performance of the different heuristics on the LU workflow.

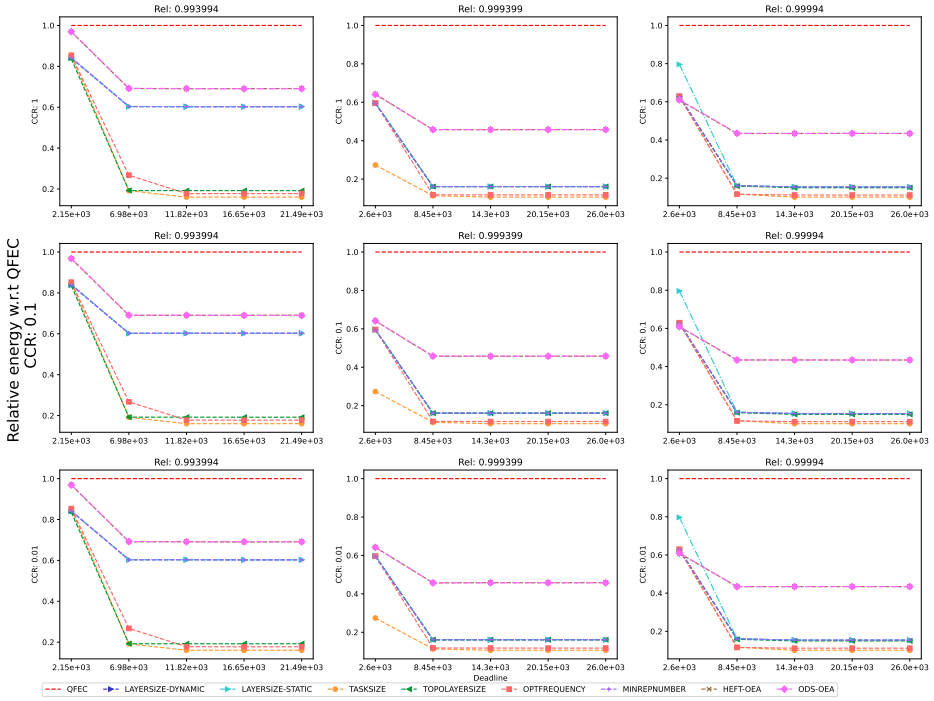

Figure 747: Performance of the different heuristics on the Montage workflow.

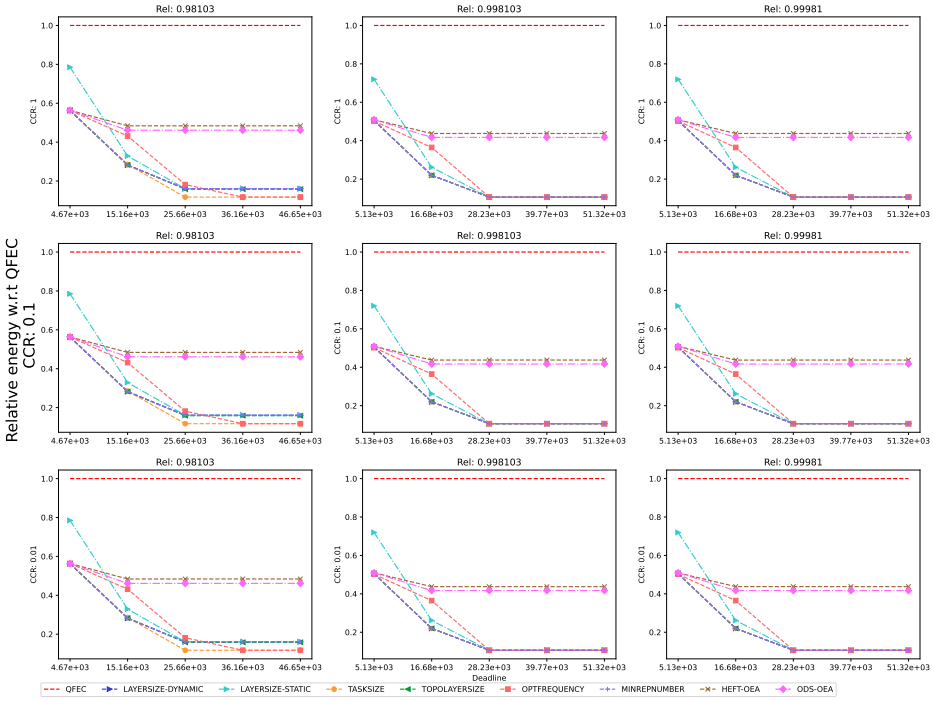

Figure 748: Performance of the different heuristics on the QR workflow.

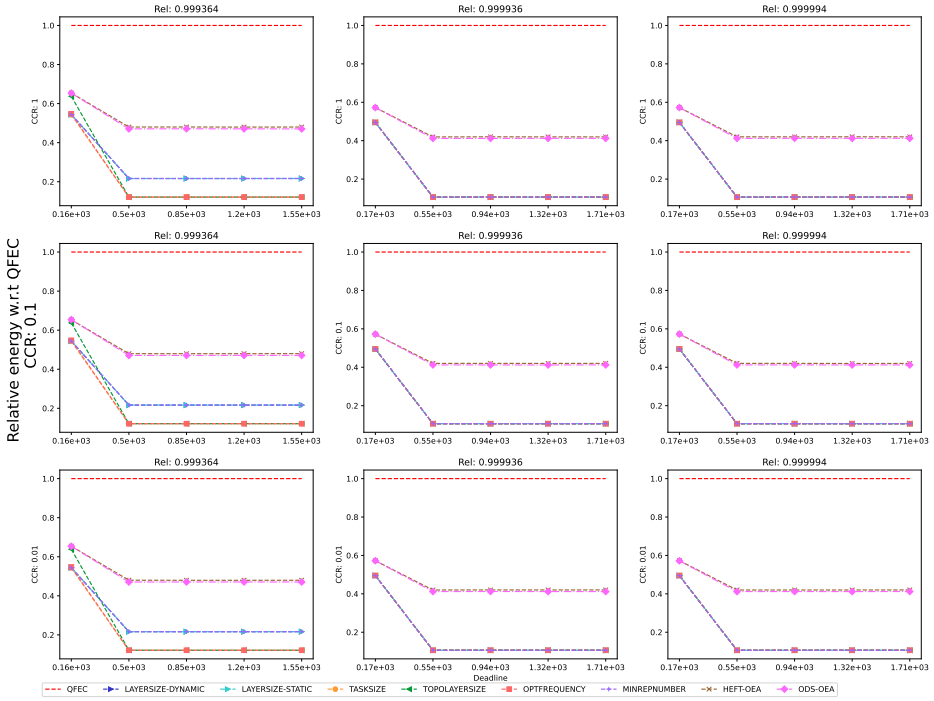

Figure 749: Performance of the different heuristics on the Seismology workflow.

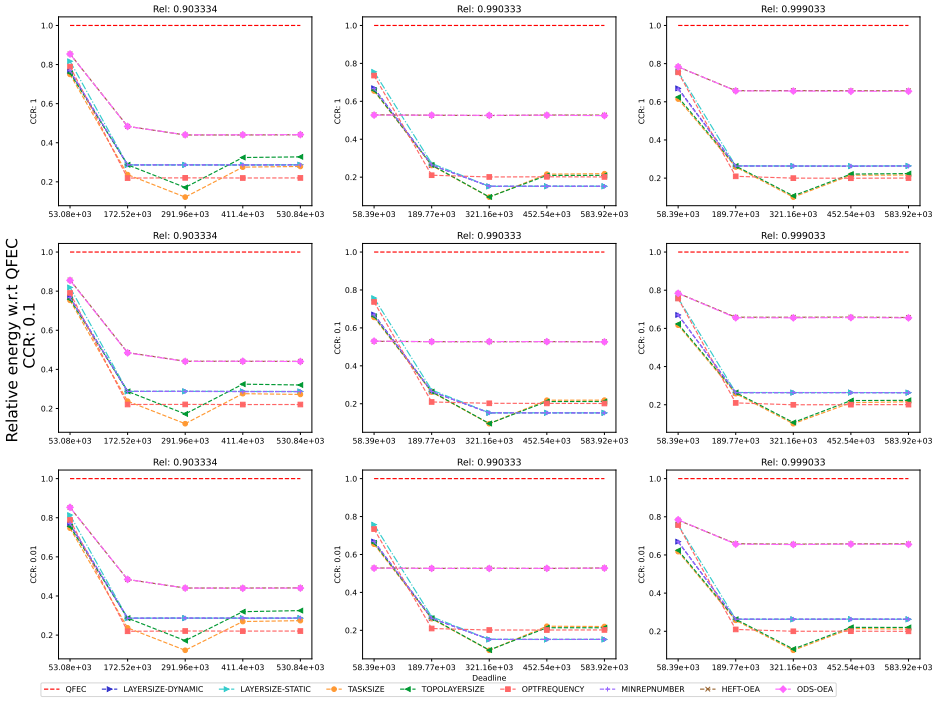

Figure 750: Performance of the different heuristics on the SoyKB workflow.

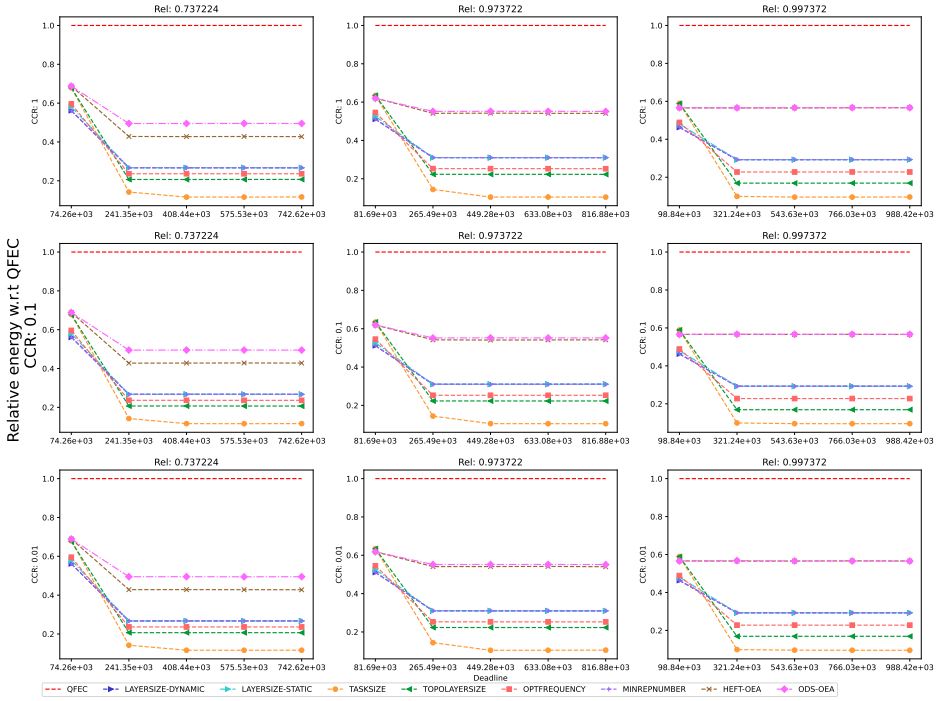

Figure 751: Performance of the different heuristics on the SRASearch.

E.8.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

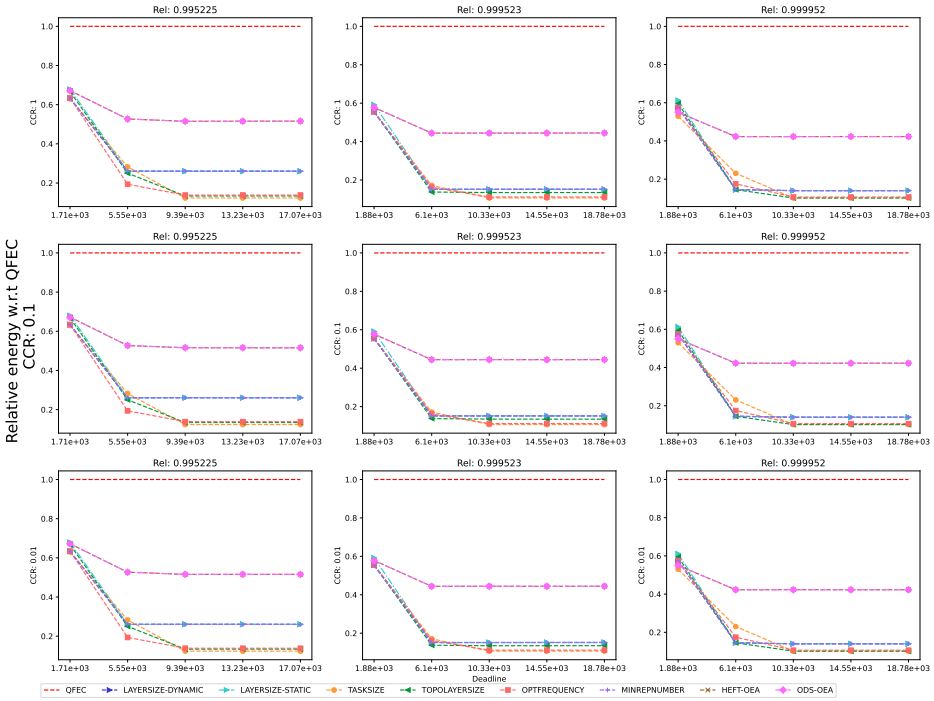

Figure 752: Performance of the different heuristics on the BWA workflow.

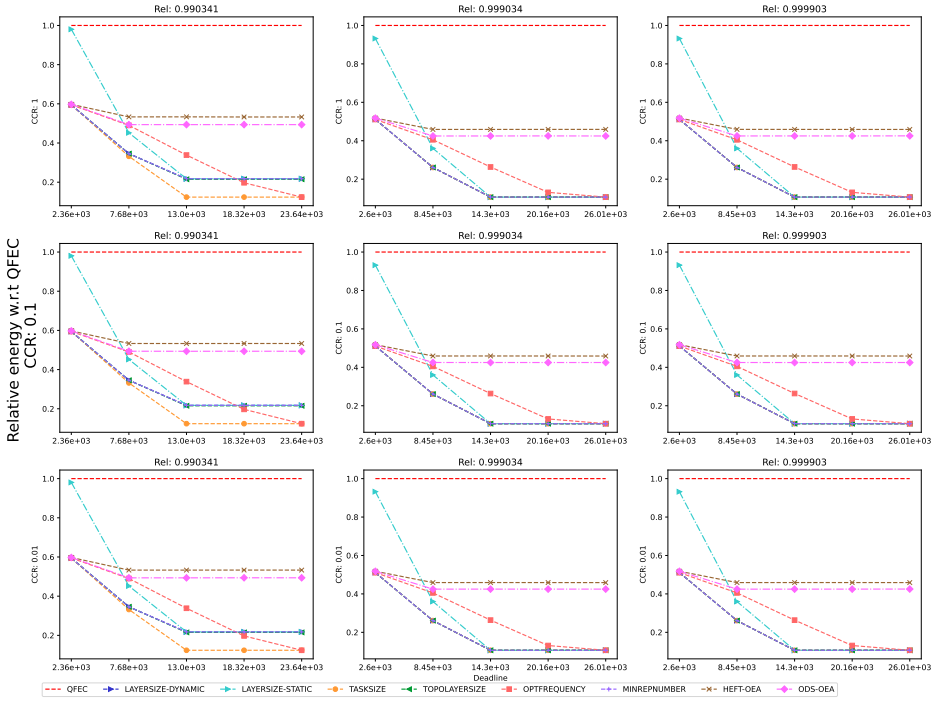

Figure 753: Performance of the different heuristics on the Cholesky workflow.

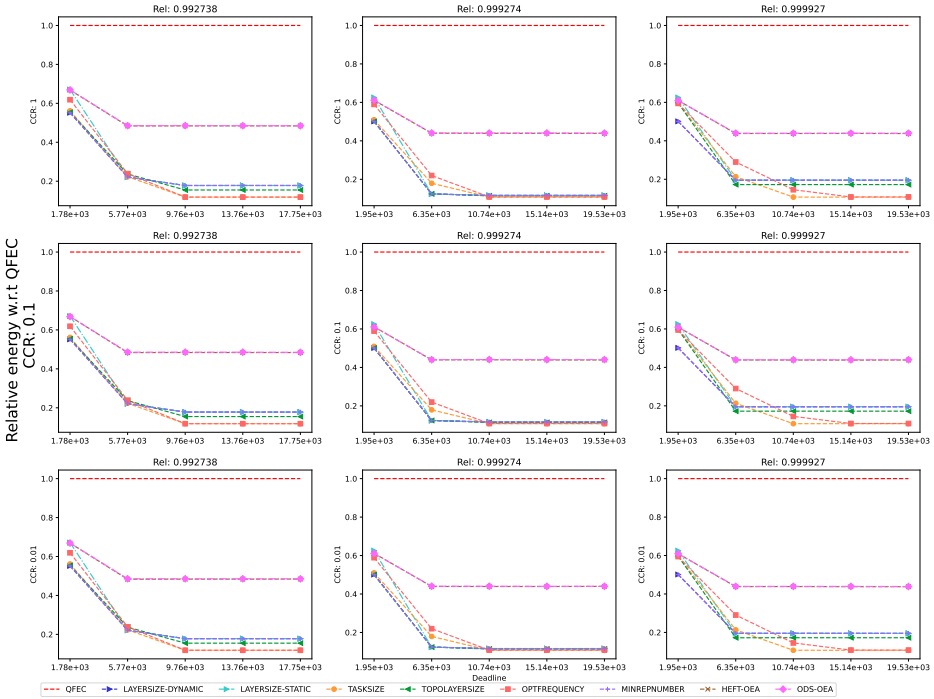

Figure 754: Performance of the different heuristics on the Cycles workflow.

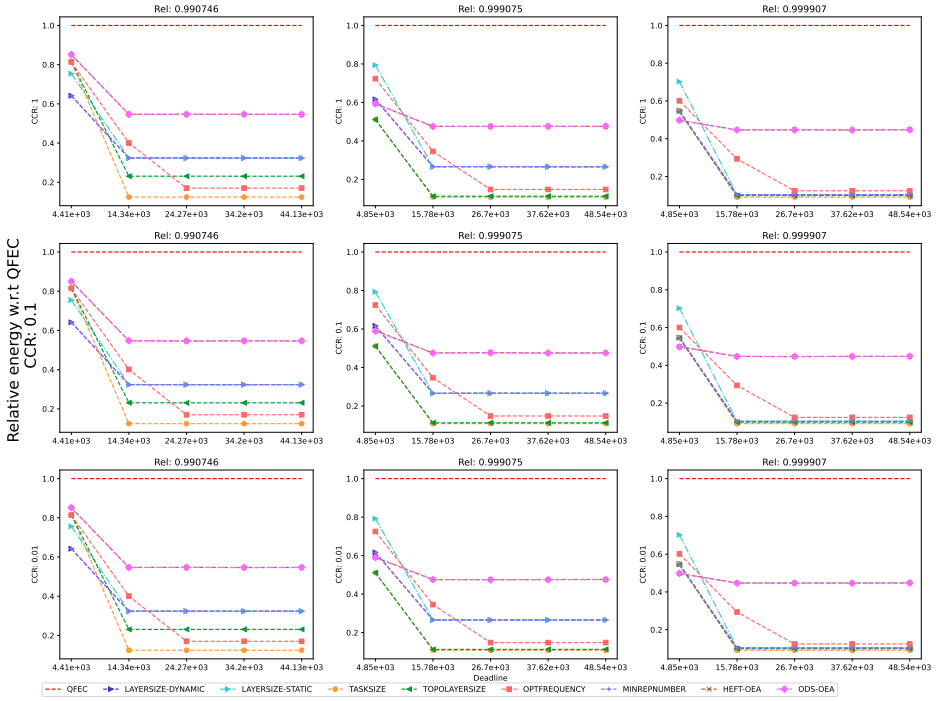

Figure 755: Performance of the different heuristics on the Epigenomics workflow.

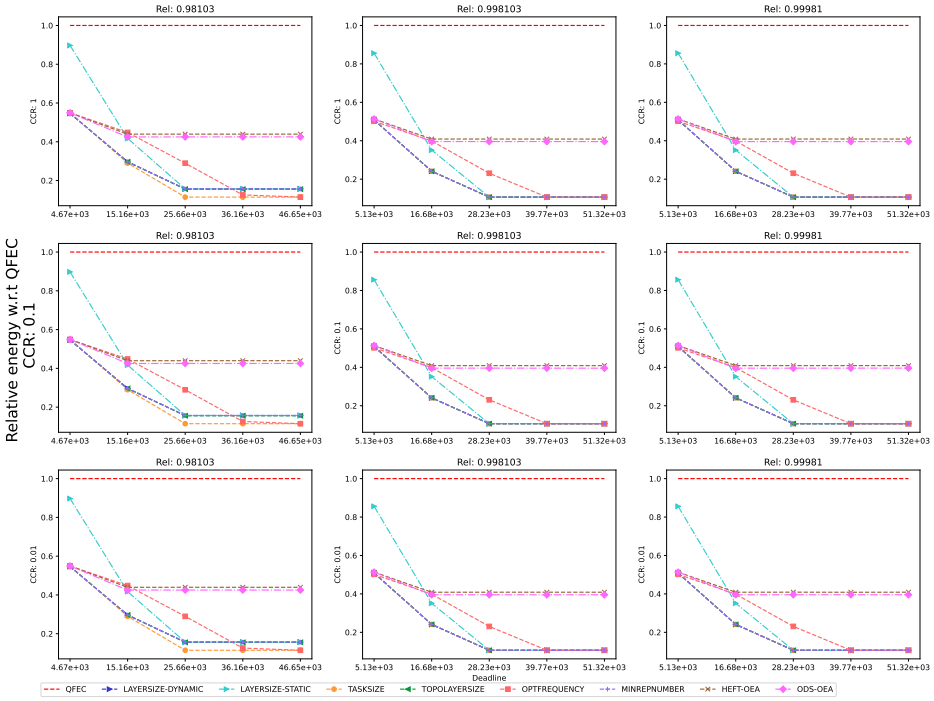

Figure 756: Performance of the different heuristics on the LU workflow.

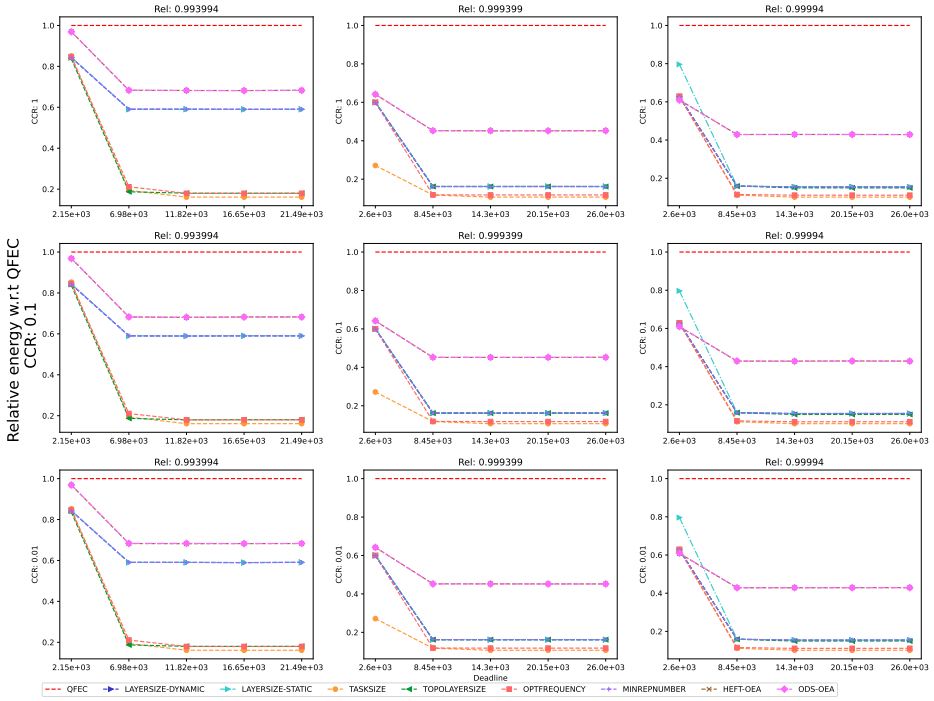

Figure 757: Performance of the different heuristics on the Montage workflow.

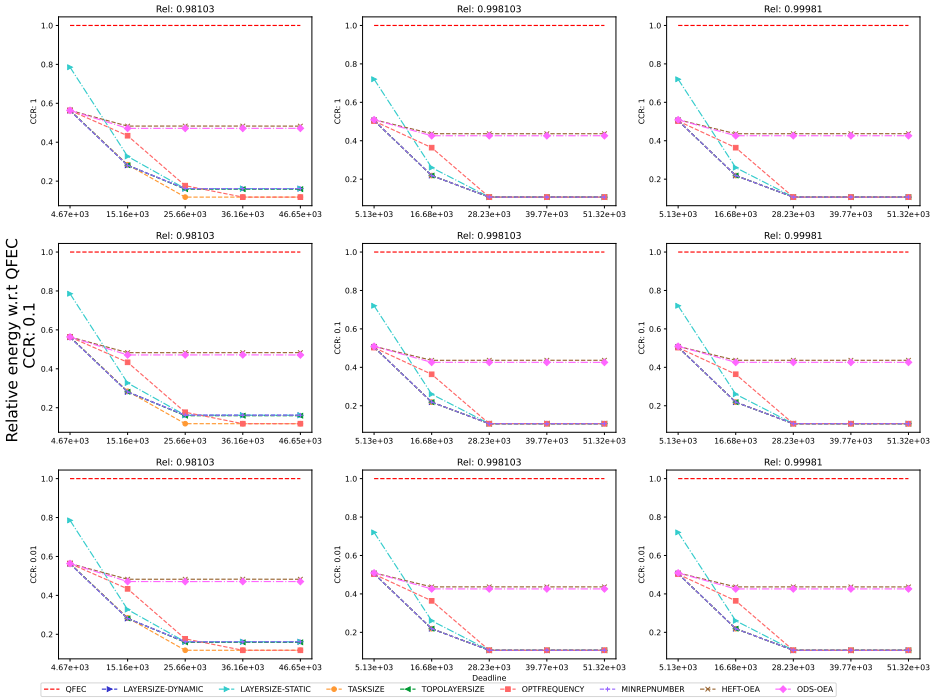

Figure 758: Performance of the different heuristics on the QR workflow.

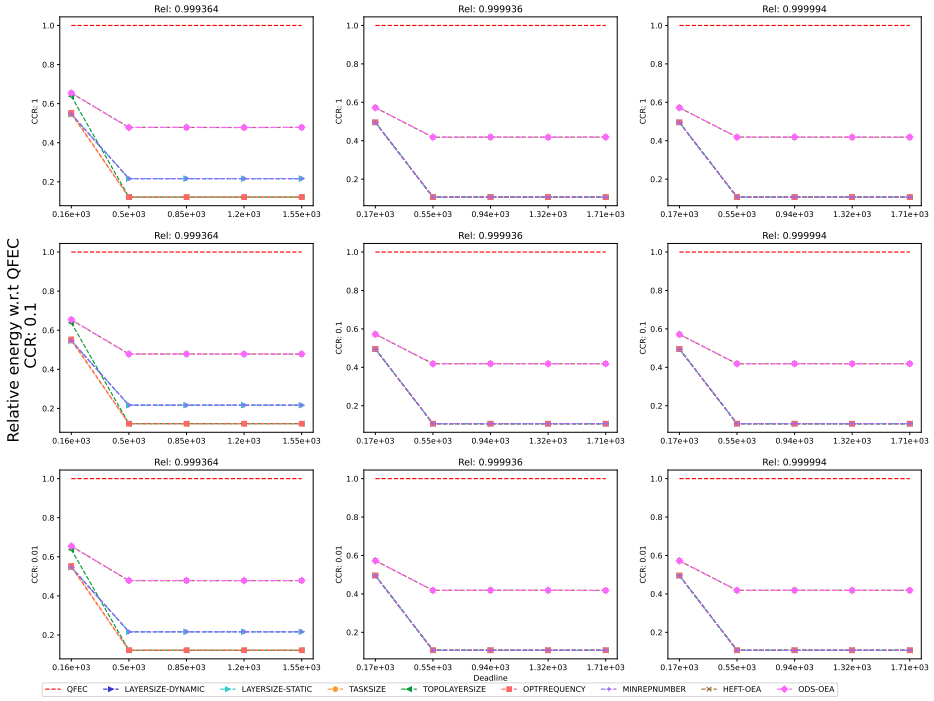

Figure 759: Performance of the different heuristics on the Seismology workflow.

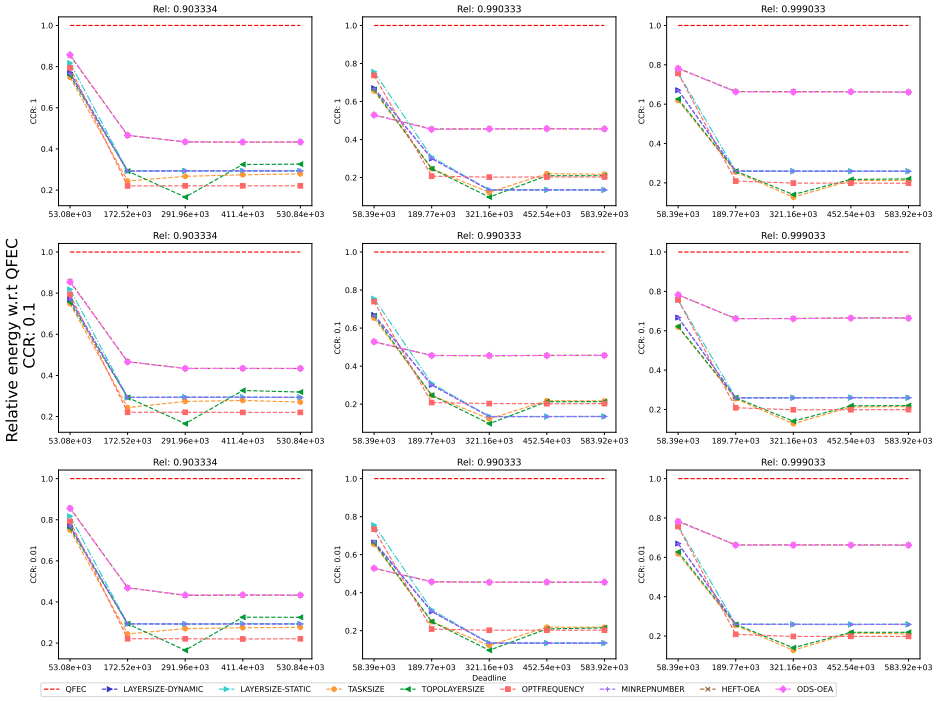

Figure 760: Performance of the different heuristics on the SoyKB workflow.

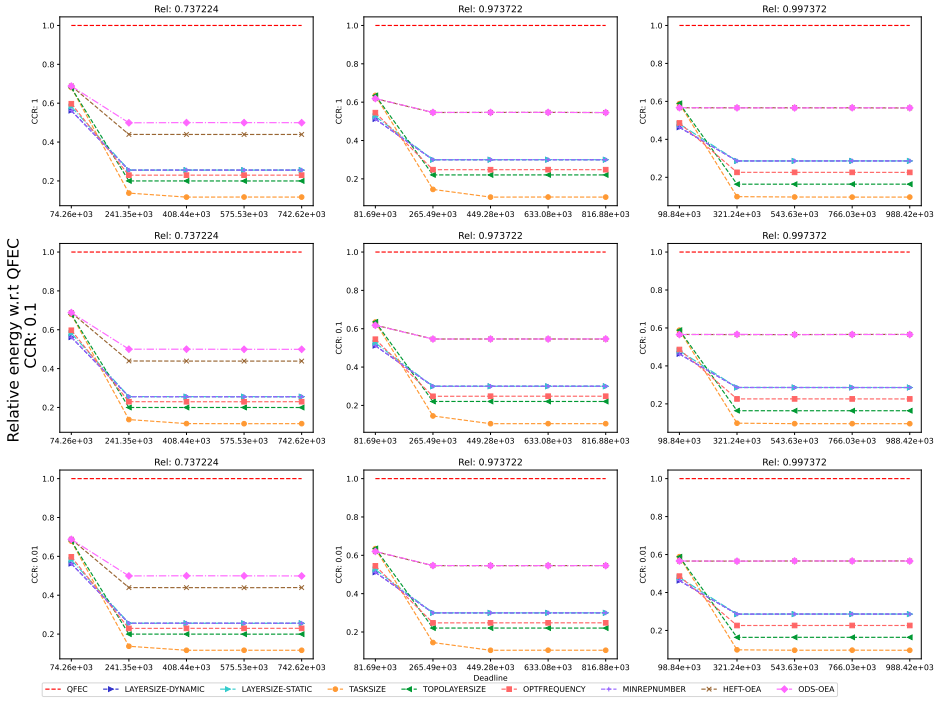

Figure 761: Performance of the different heuristics on the SRASearch.

E.9 $BC/WC = 0.9$

E.9.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

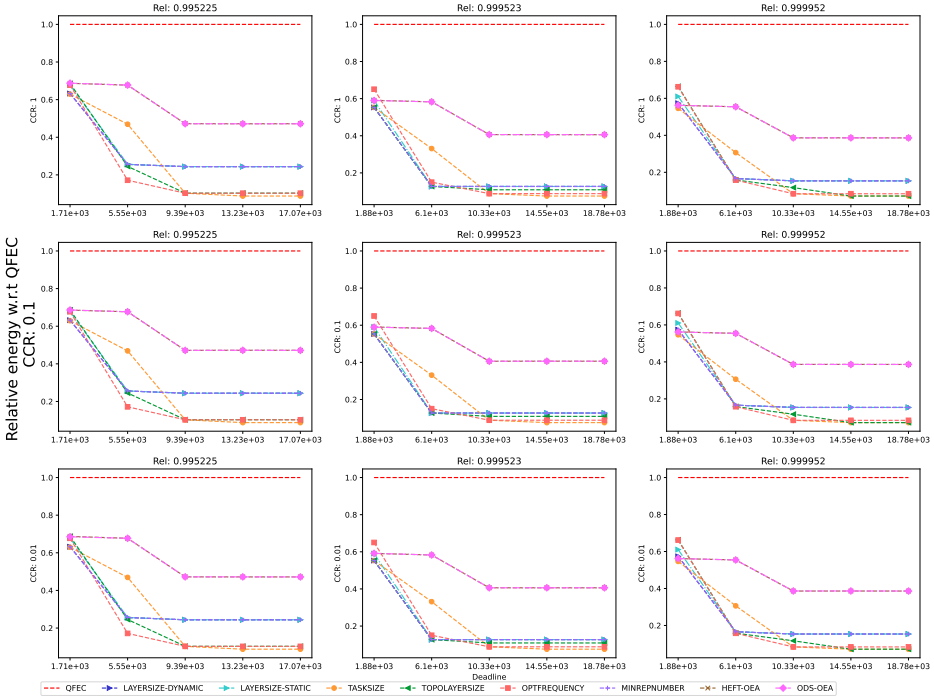

Figure 762: Performance of the different heuristics on the BWA workflow.

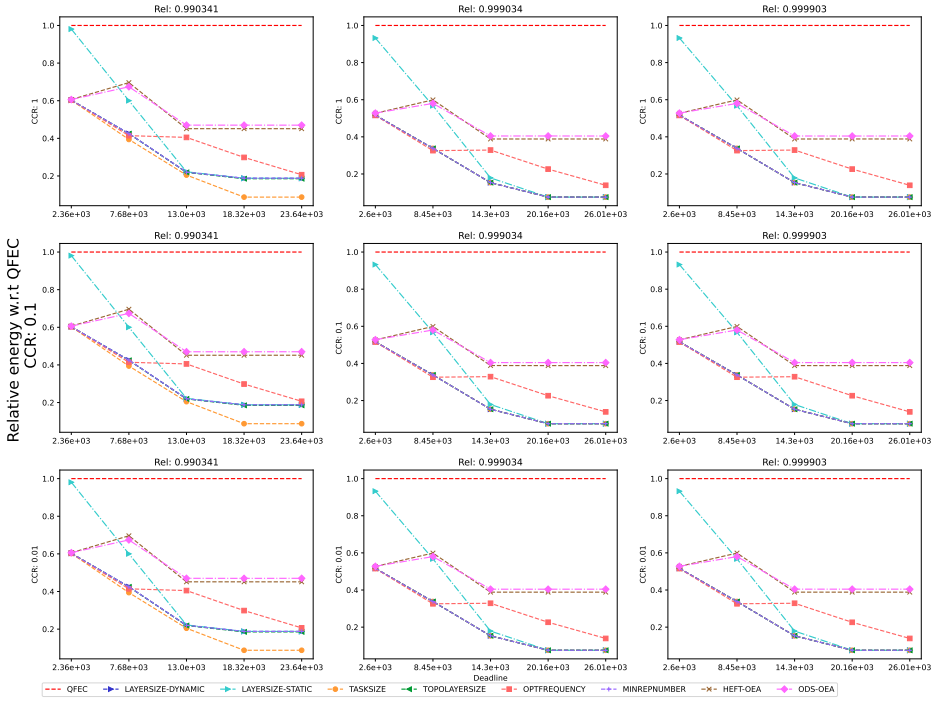

Figure 763: Performance of the different heuristics on the Cholesky workflow.

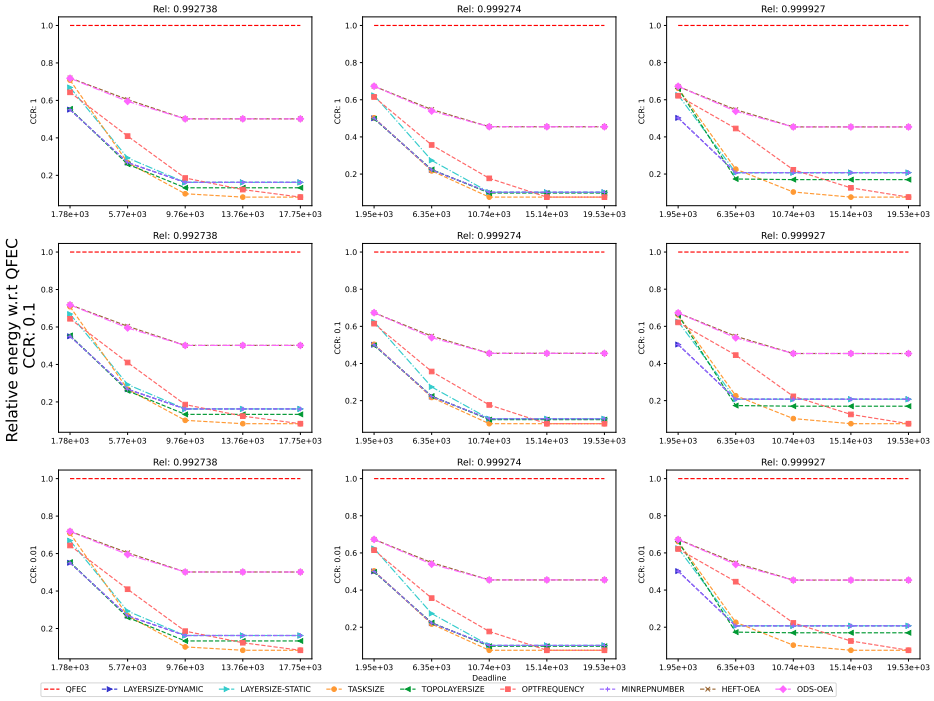

Figure 764: Performance of the different heuristics on the Cycles workflow.

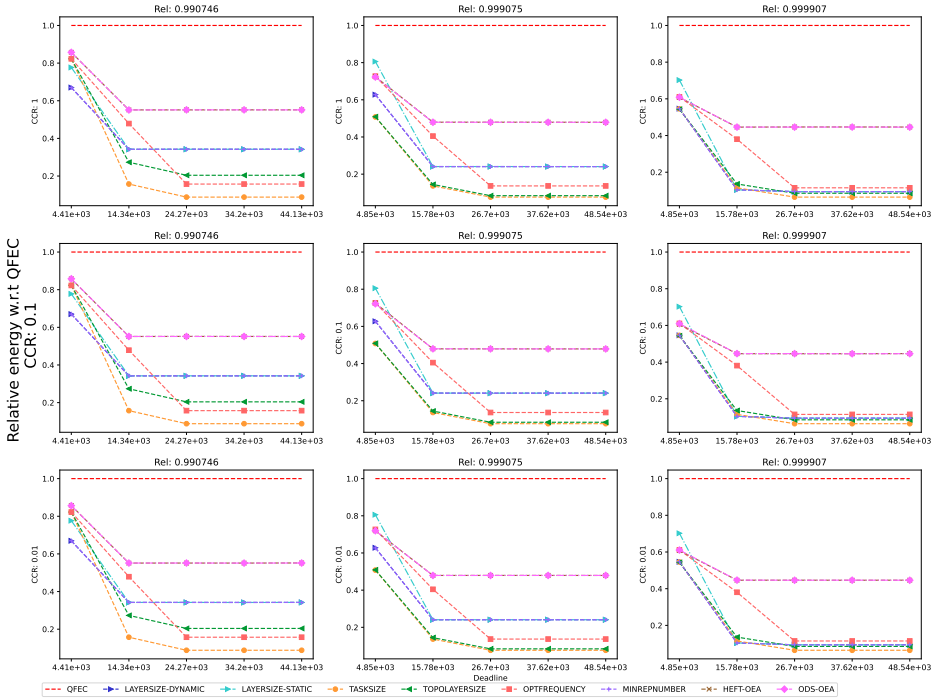

Figure 765: Performance of the different heuristics on the Epigenomics workflow.

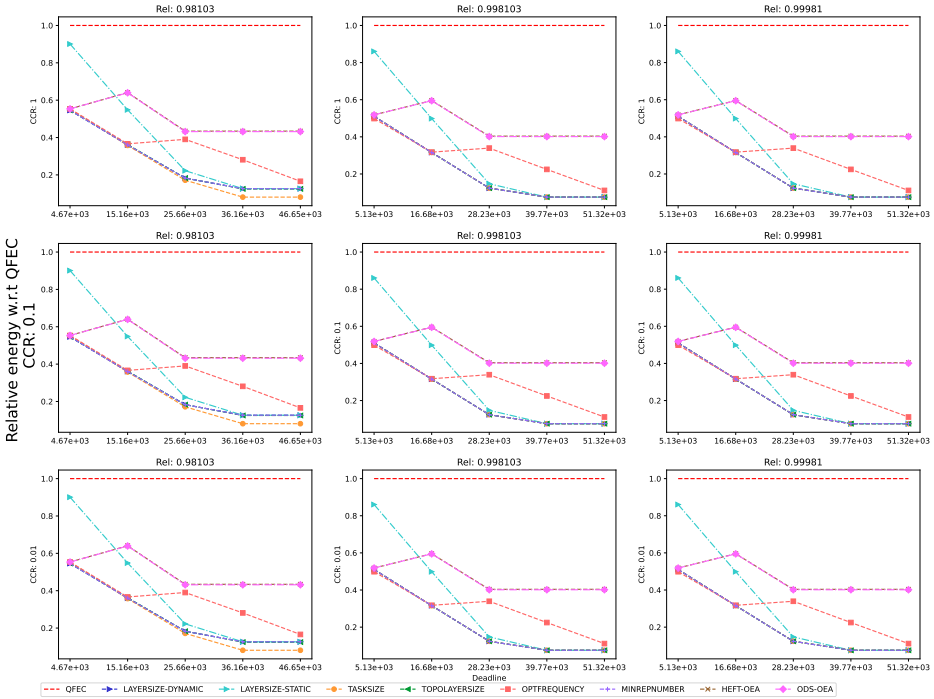

Figure 766: Performance of the different heuristics on the LU workflow.

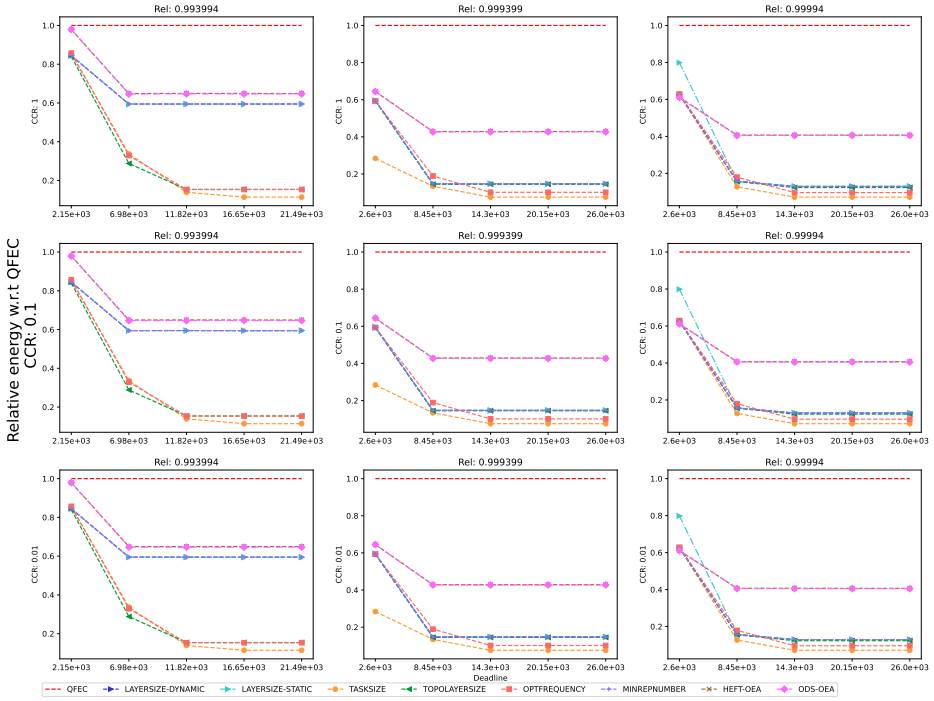

Figure 767: Performance of the different heuristics on the Montage workflow.

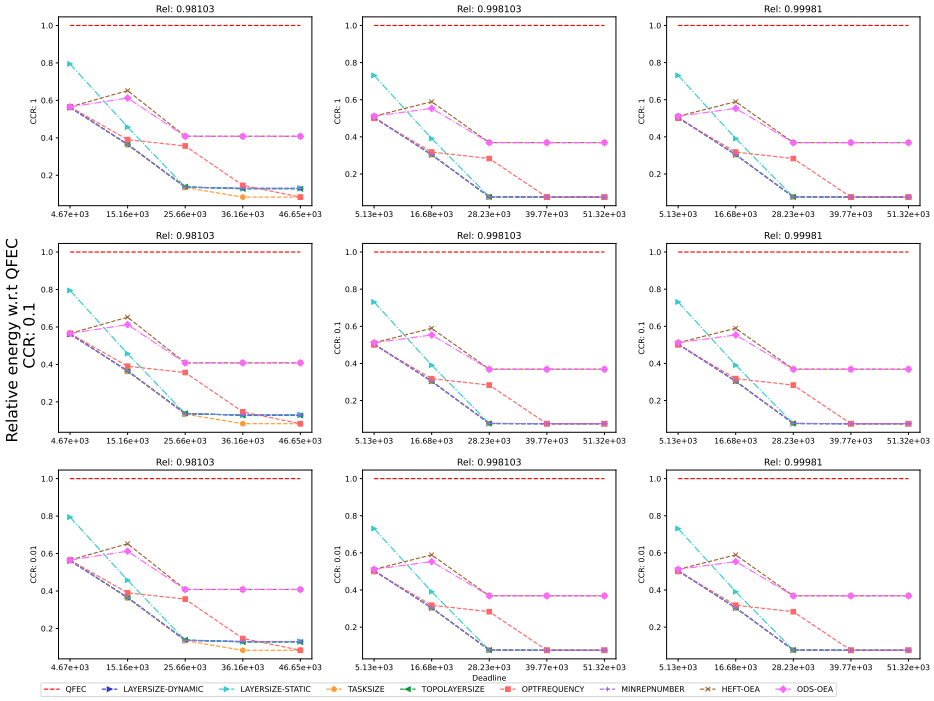

Figure 768: Performance of the different heuristics on the QR workflow.

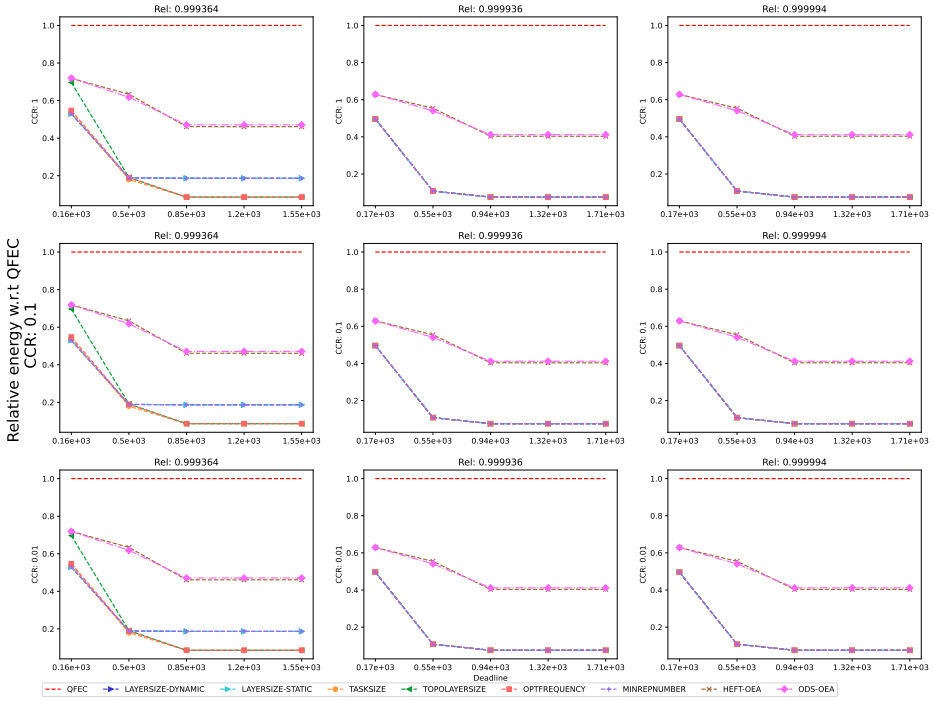

Figure 769: Performance of the different heuristics on the Seismology workflow.

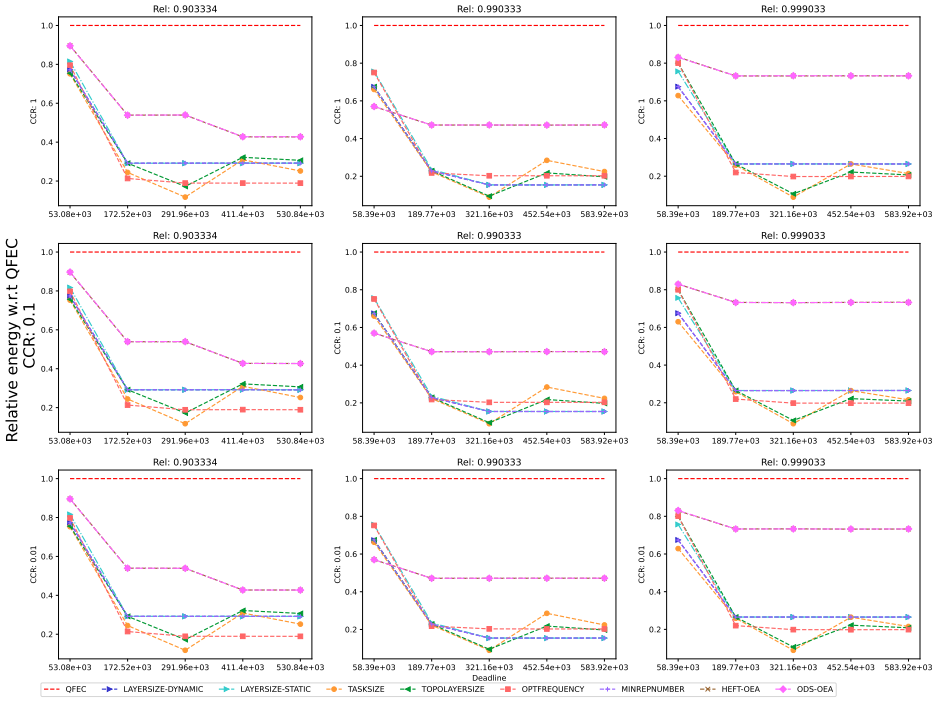

Figure 770: Performance of the different heuristics on the SoyKB workflow.

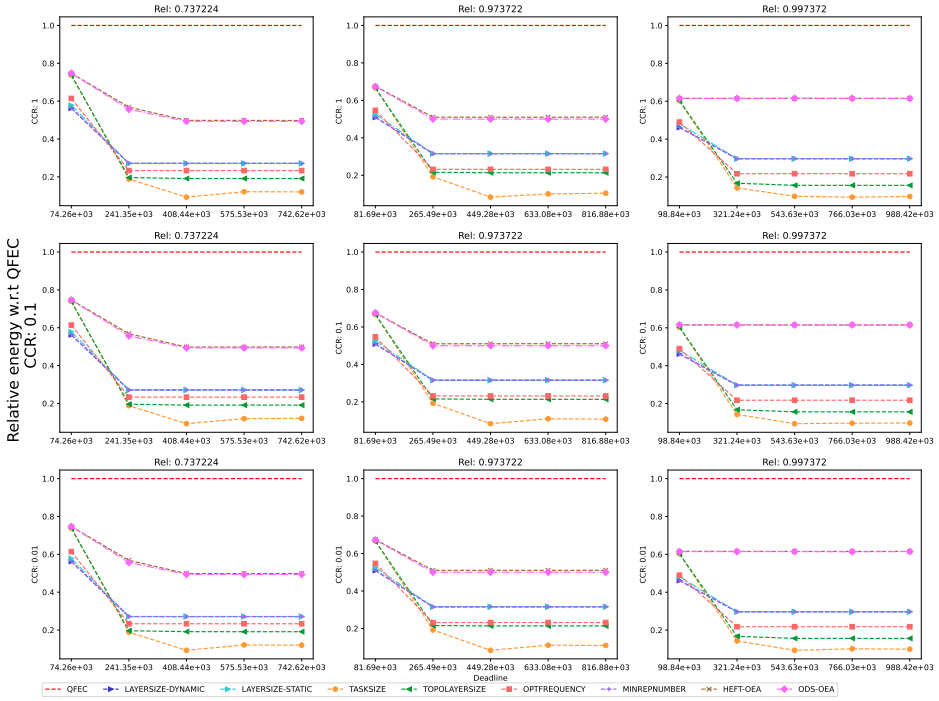

Figure 771: Performance of the different heuristics on the SRASearch.

E.9.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

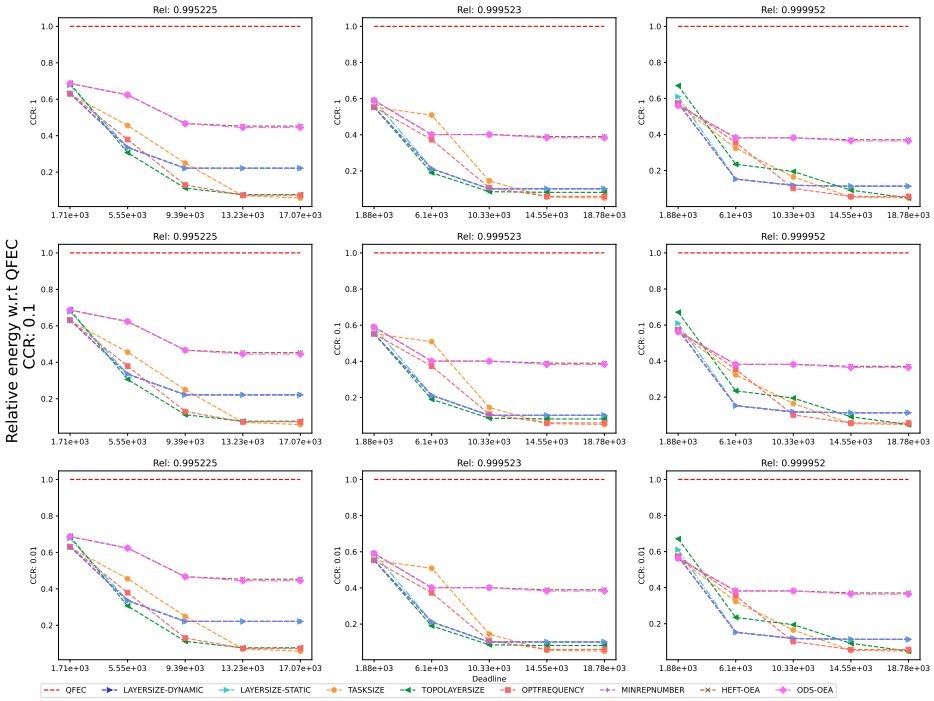

Figure 772: Performance of the different heuristics on the BWA workflow.

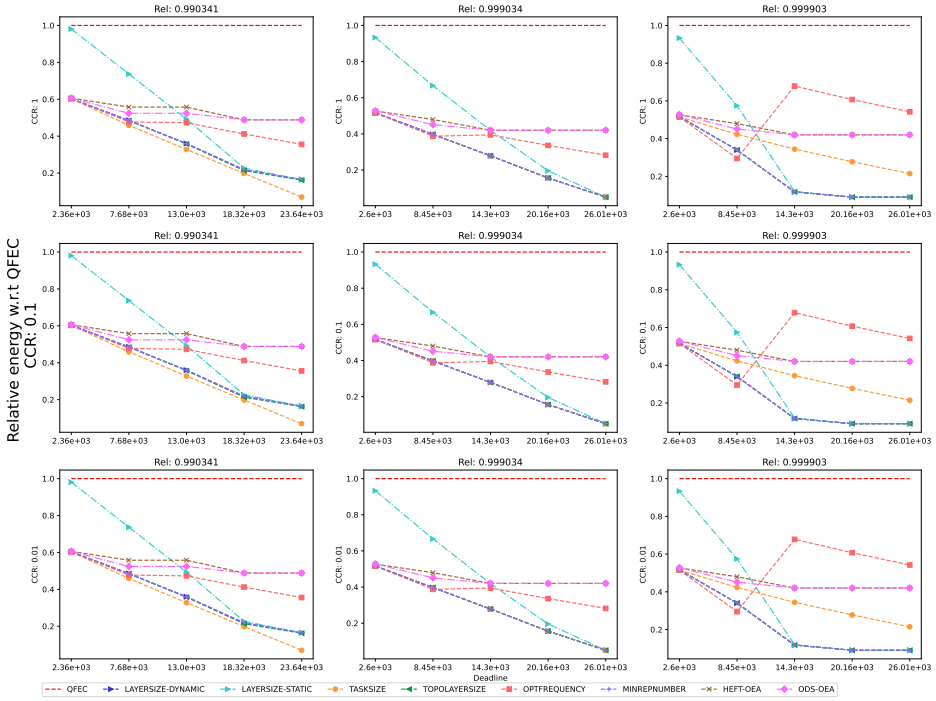

Figure 773: Performance of the different heuristics on the Cholesky workflow.

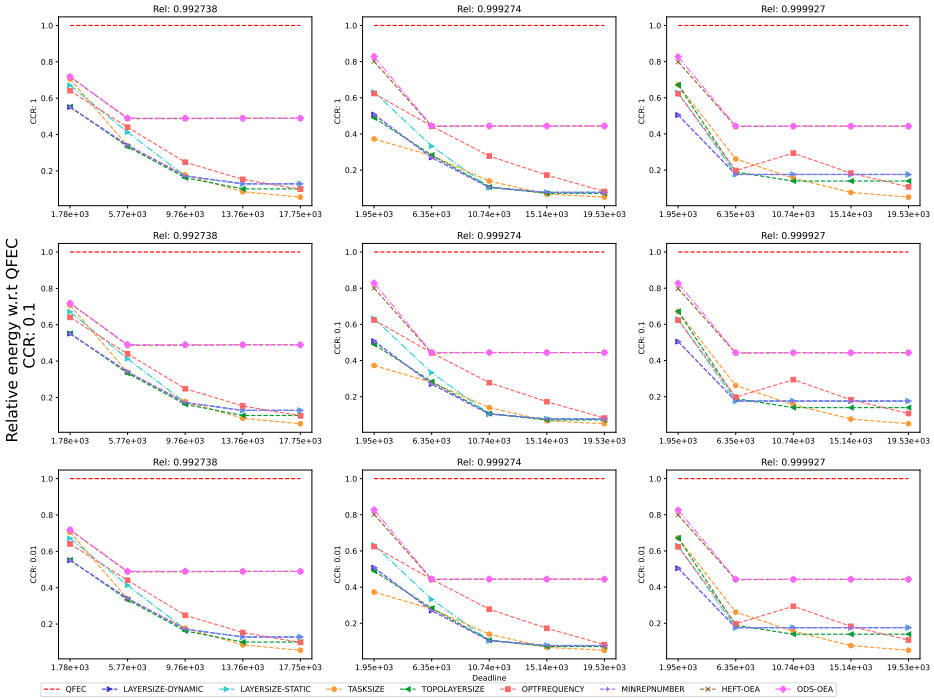

Figure 774: Performance of the different heuristics on the Cycles workflow.

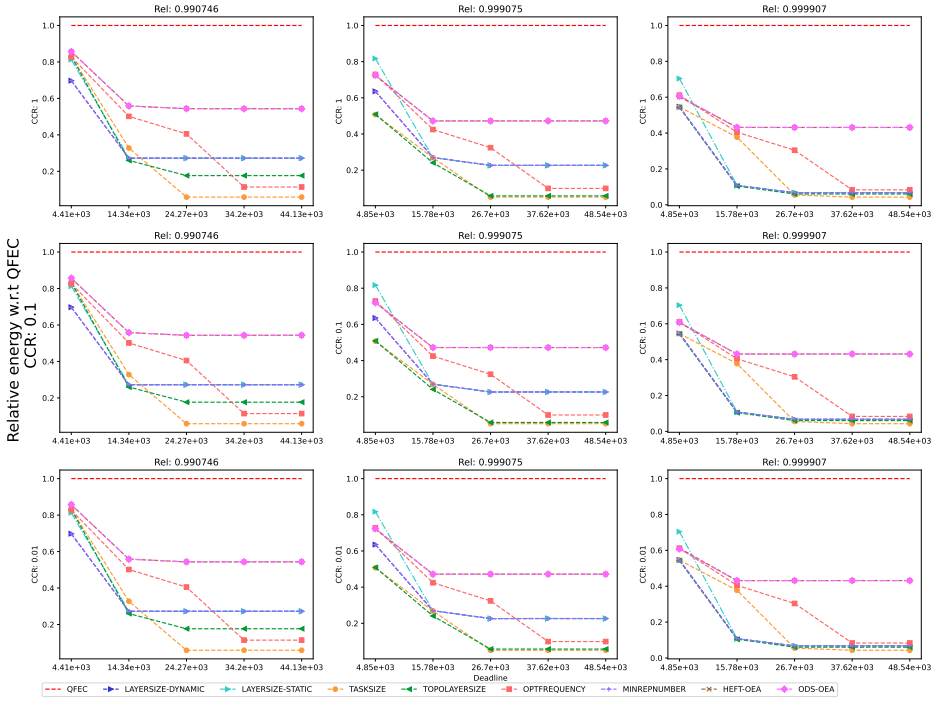

Figure 775: Performance of the different heuristics on the Epigenomics workflow.

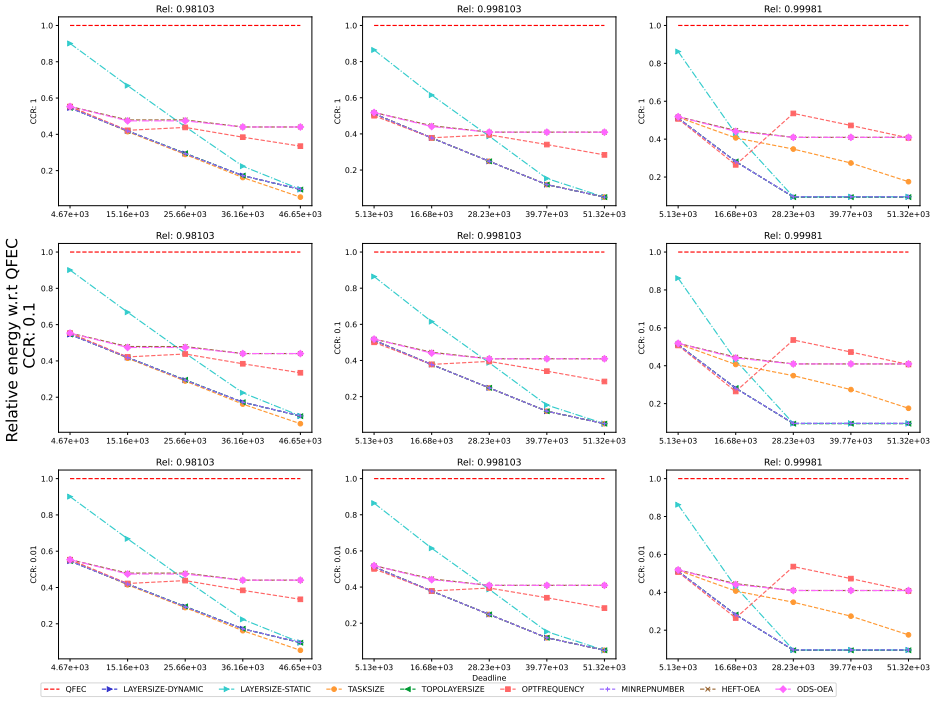

Figure 776: Performance of the different heuristics on the LU workflow.

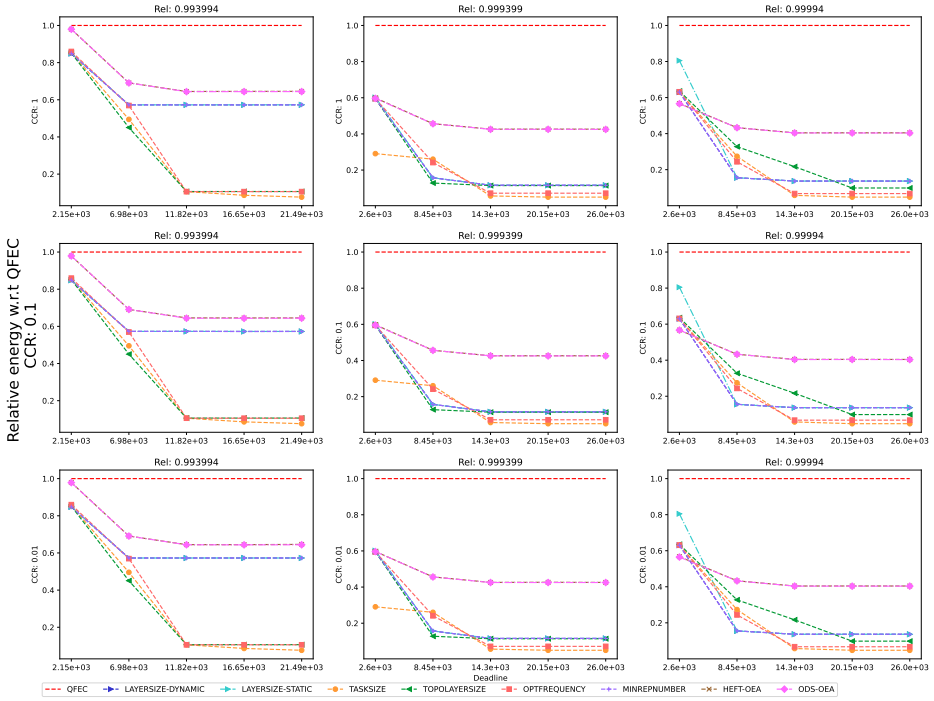

Figure 777: Performance of the different heuristics on the Montage workflow.

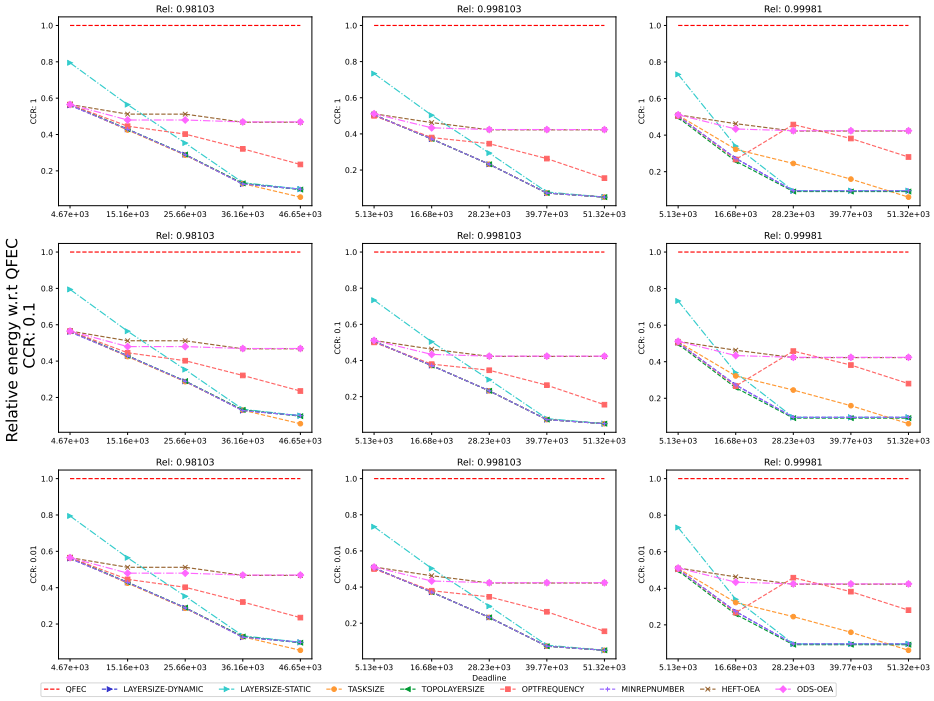

Figure 778: Performance of the different heuristics on the QR workflow.

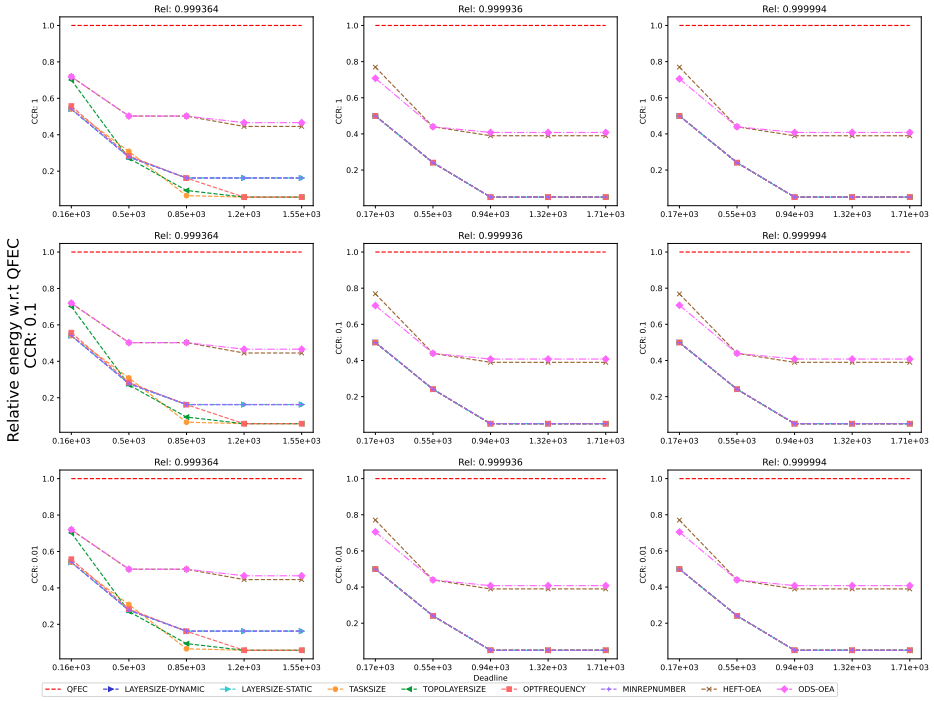

Figure 779: Performance of the different heuristics on the Seismology workflow.

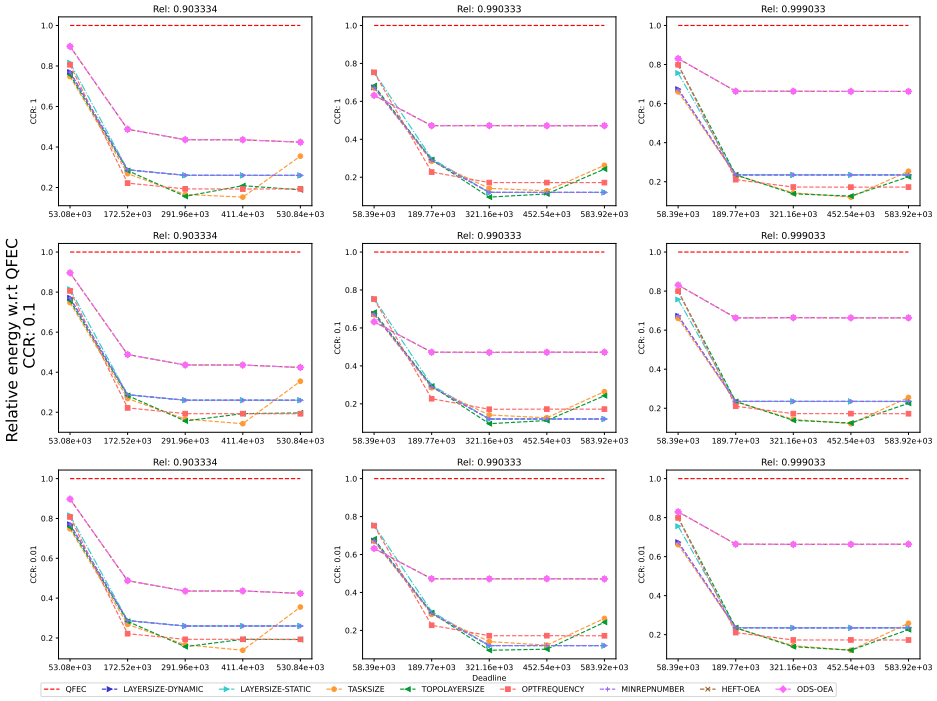

Figure 780: Performance of the different heuristics on the SoyKB workflow.

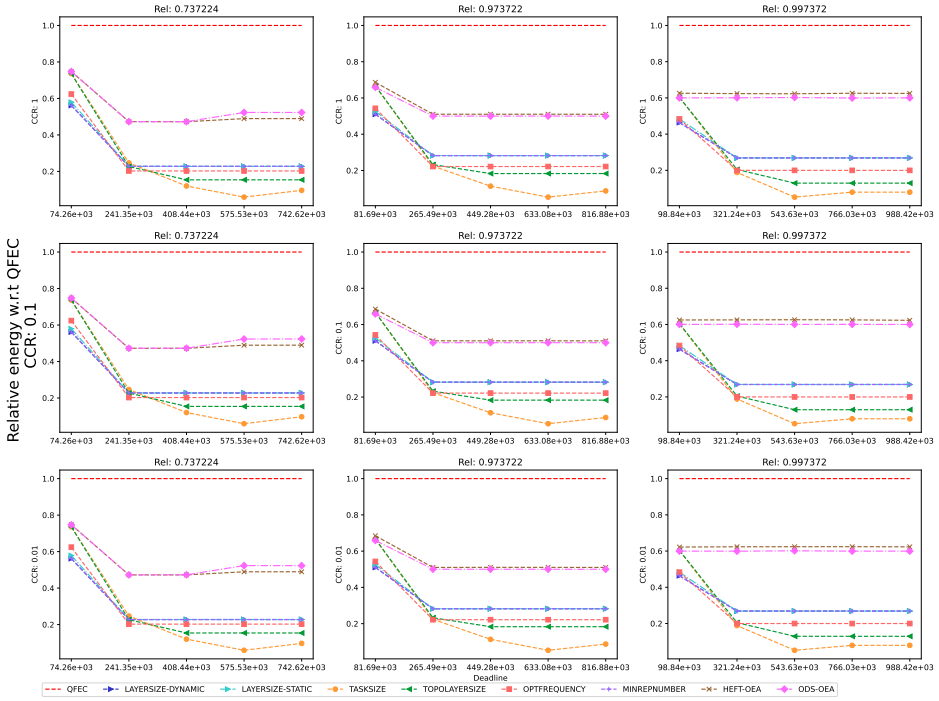

Figure 781: Performance of the different heuristics on the SRASearch.

E.9.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

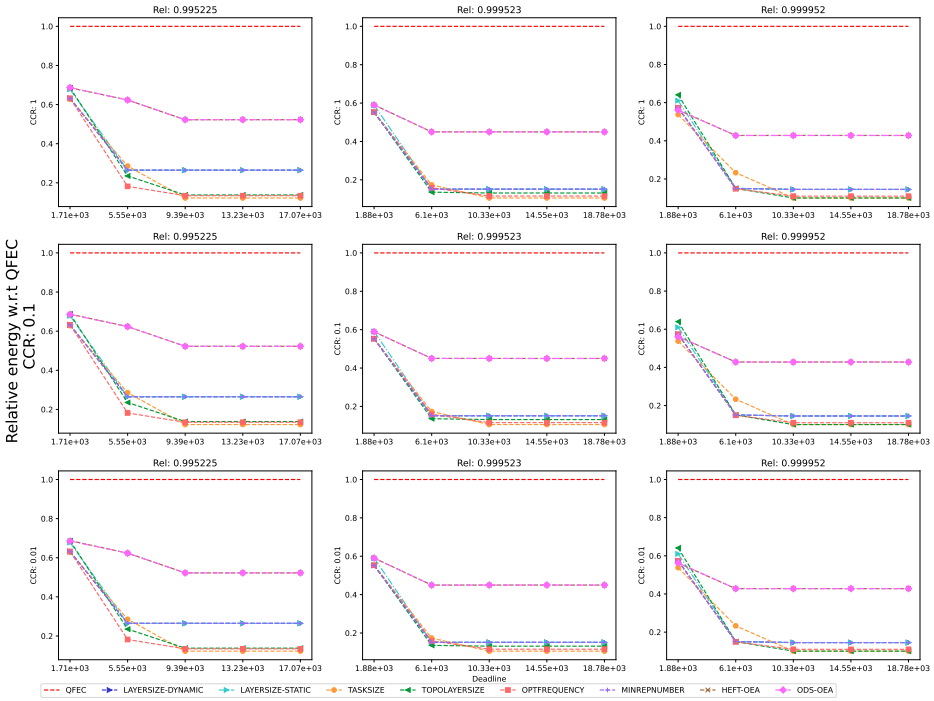

Figure 782: Performance of the different heuristics on the BWA workflow.

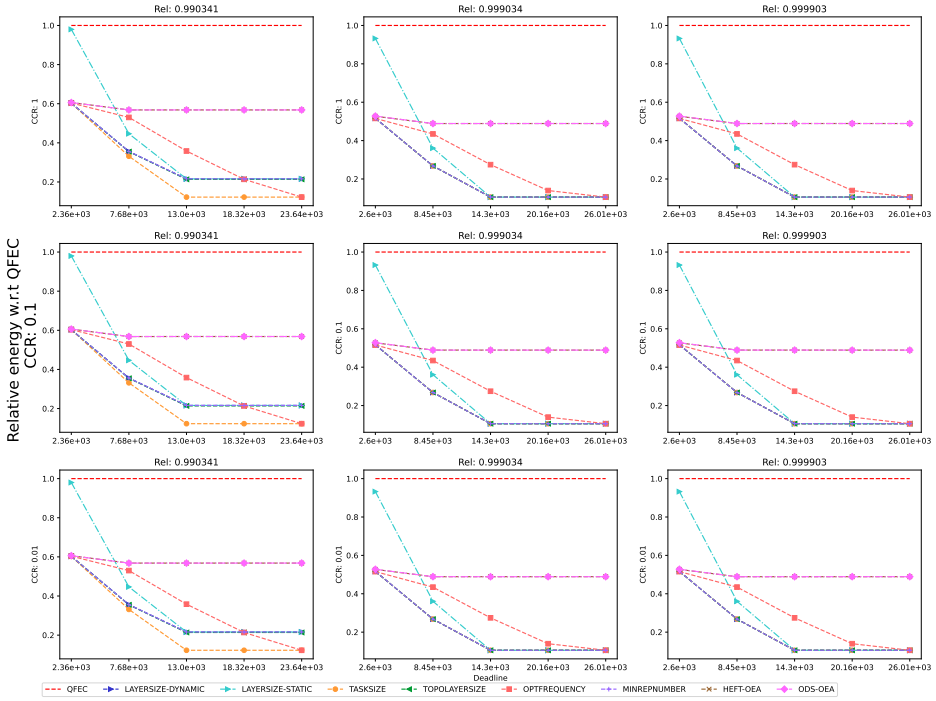

Figure 783: Performance of the different heuristics on the Cholesky workflow.

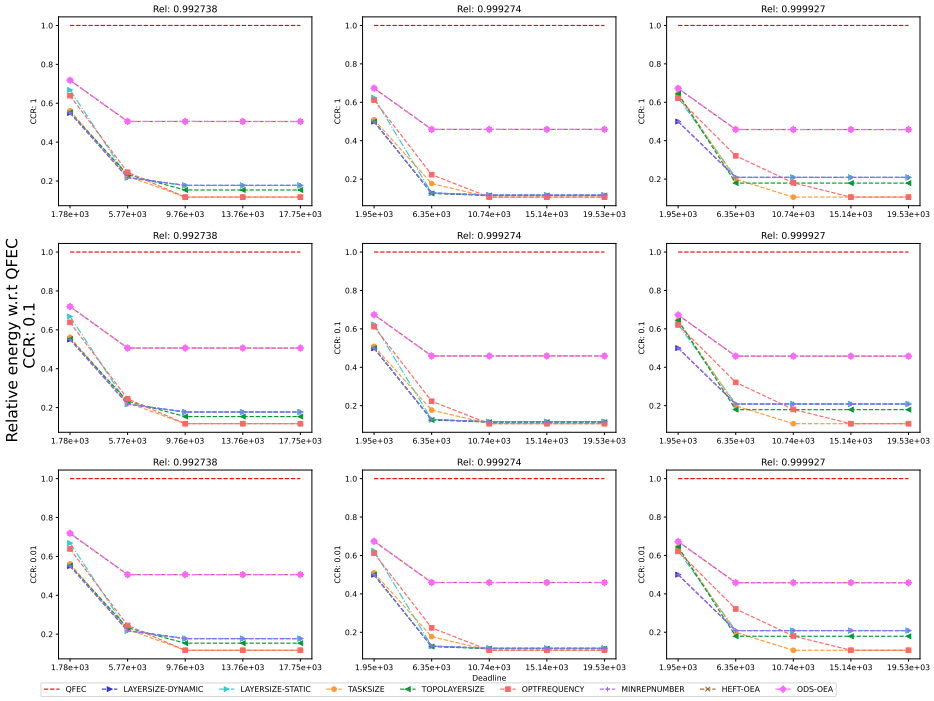

Figure 784: Performance of the different heuristics on the Cycles workflow.

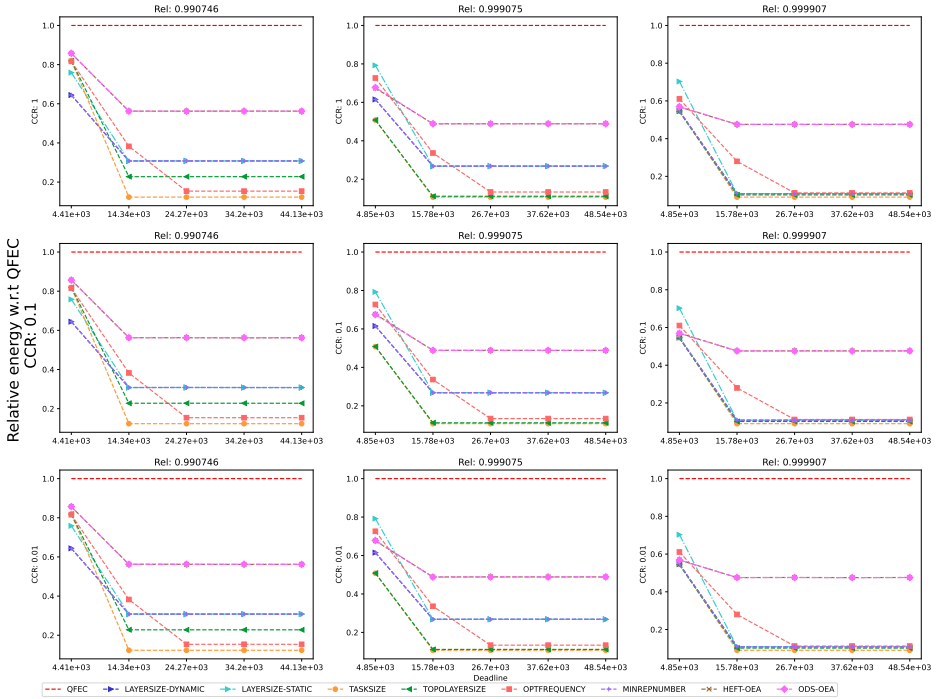

Figure 785: Performance of the different heuristics on the Epigenomics workflow.

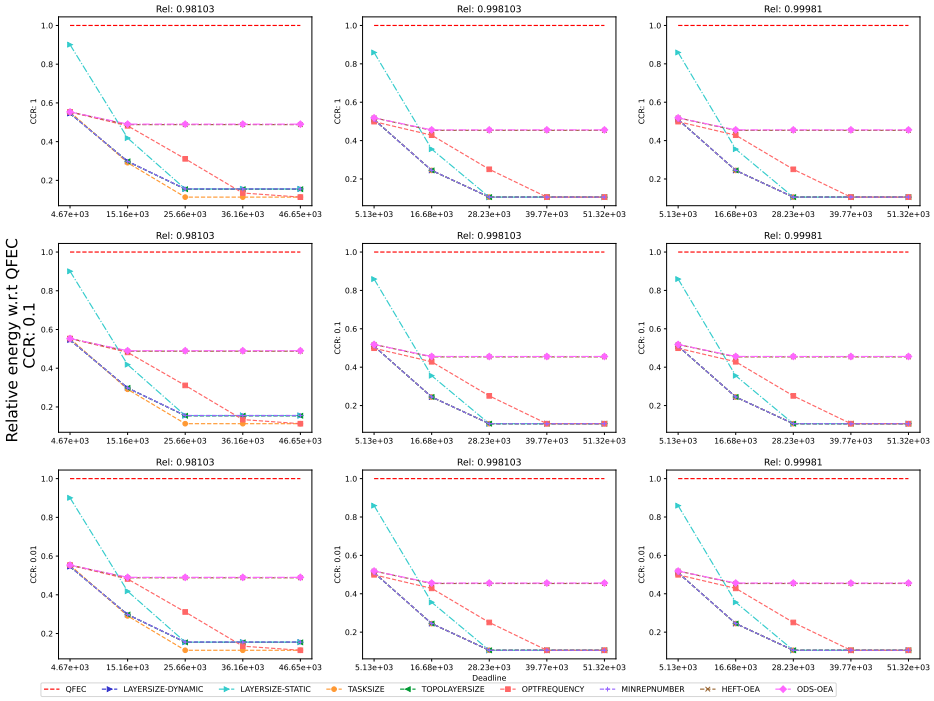

Figure 786: Performance of the different heuristics on the LU workflow.

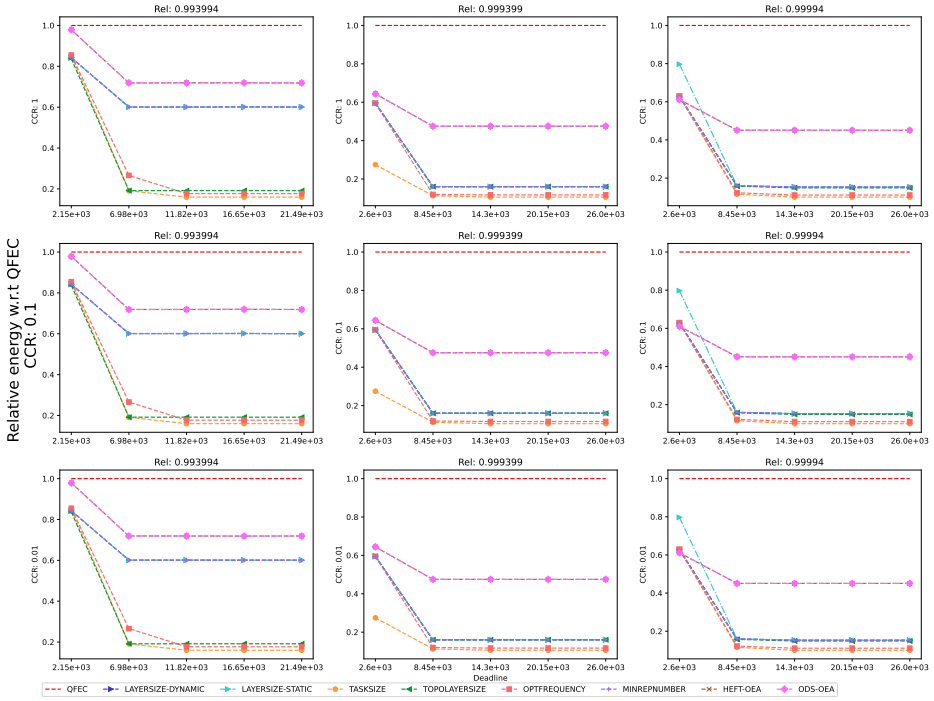

Figure 787: Performance of the different heuristics on the Montage workflow.

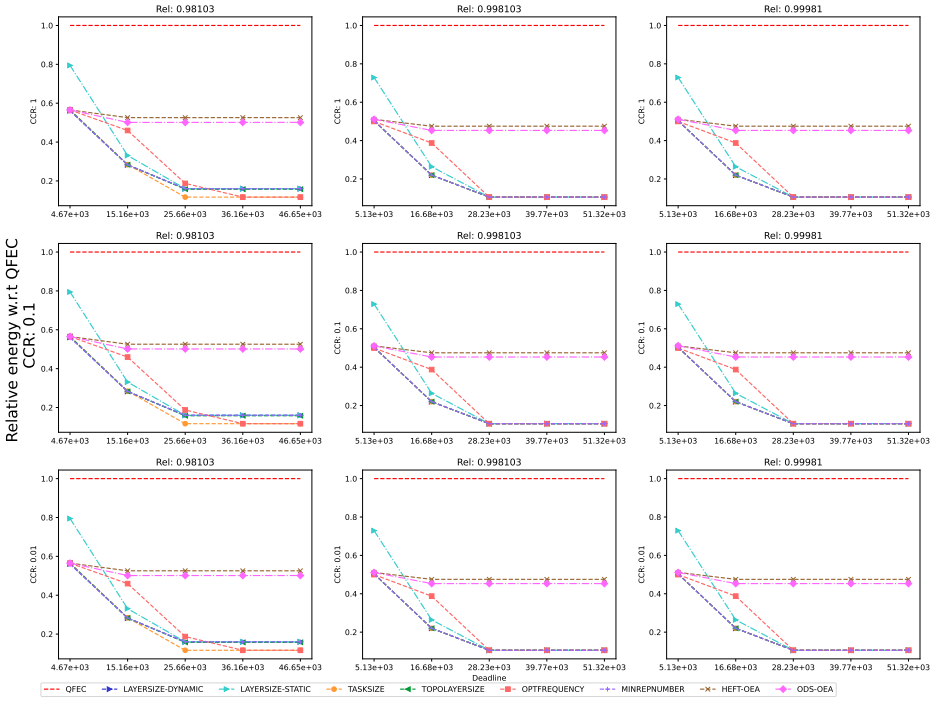

Figure 788: Performance of the different heuristics on the QR workflow.

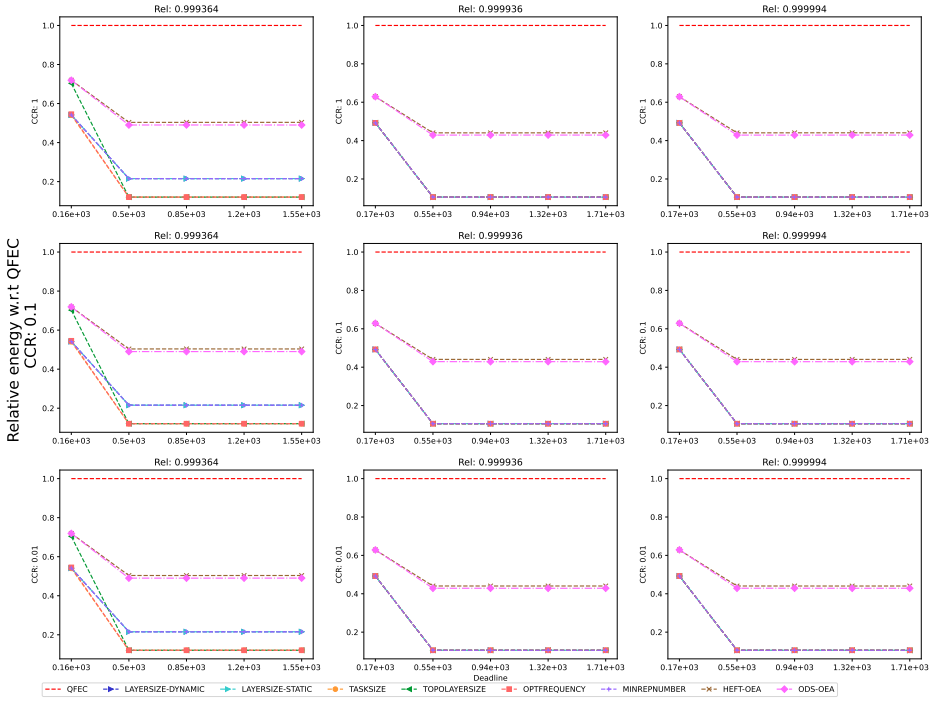

Figure 789: Performance of the different heuristics on the Seismology workflow.

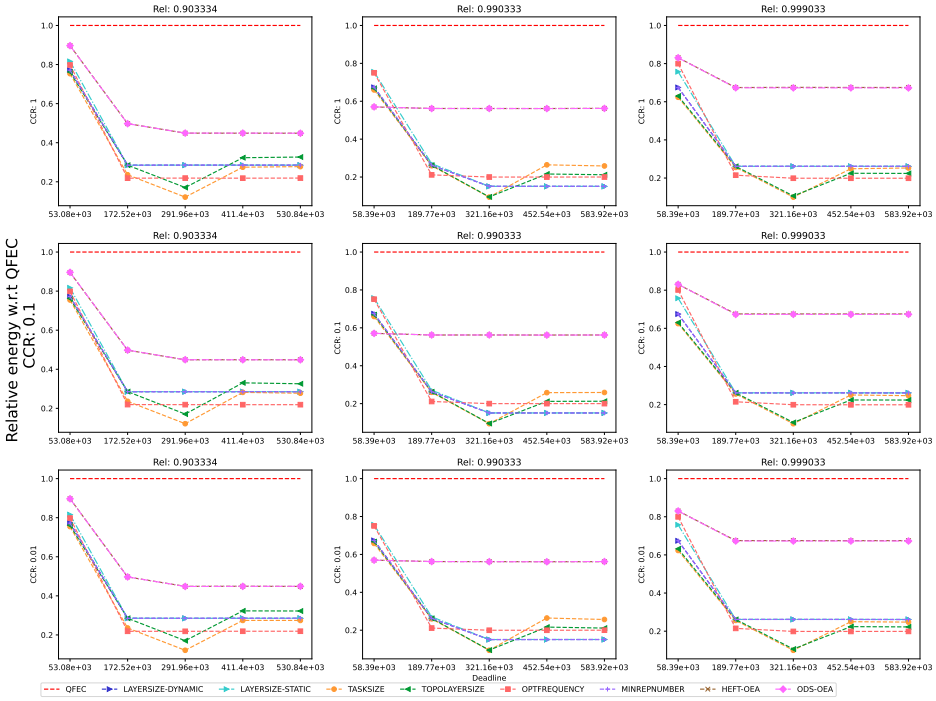

Figure 790: Performance of the different heuristics on the SoyKB workflow.

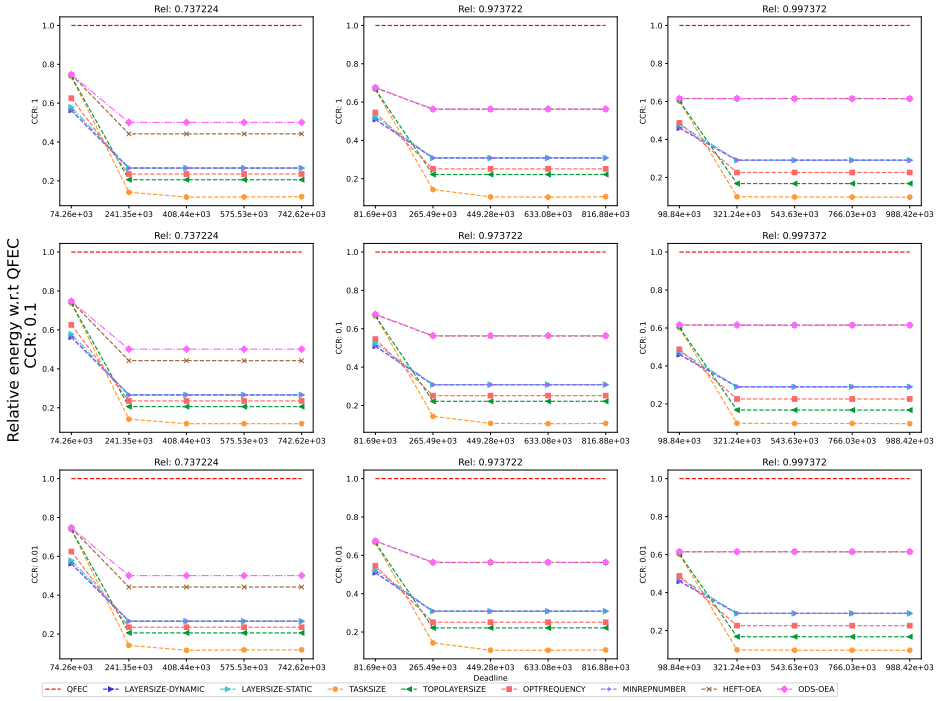

Figure 791: Performance of the different heuristics on the SRASearch.

E.9.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

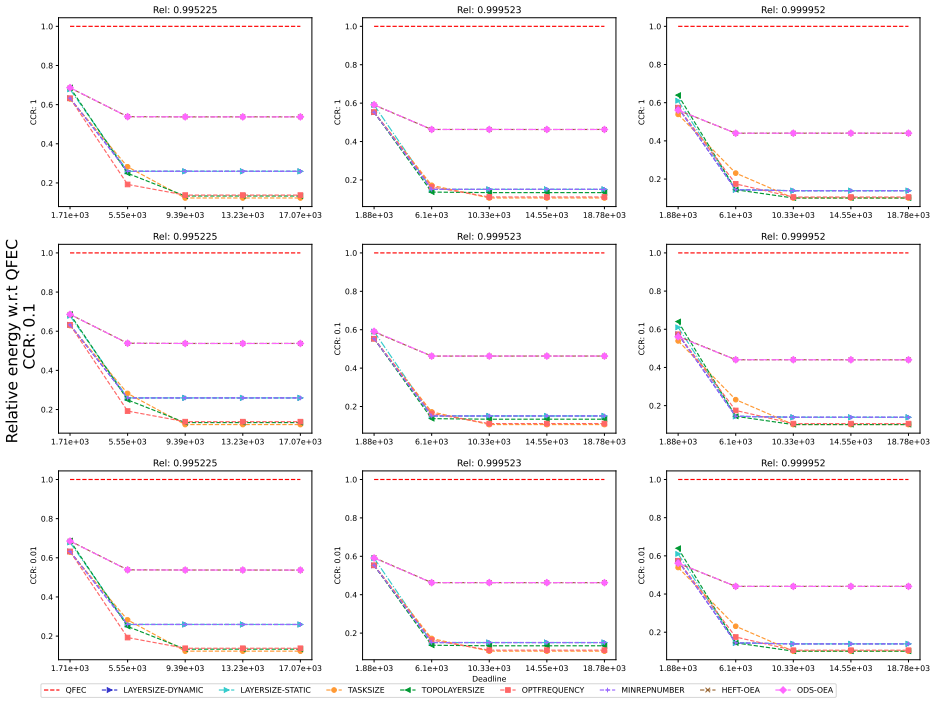

Figure 792: Performance of the different heuristics on the BWA workflow.

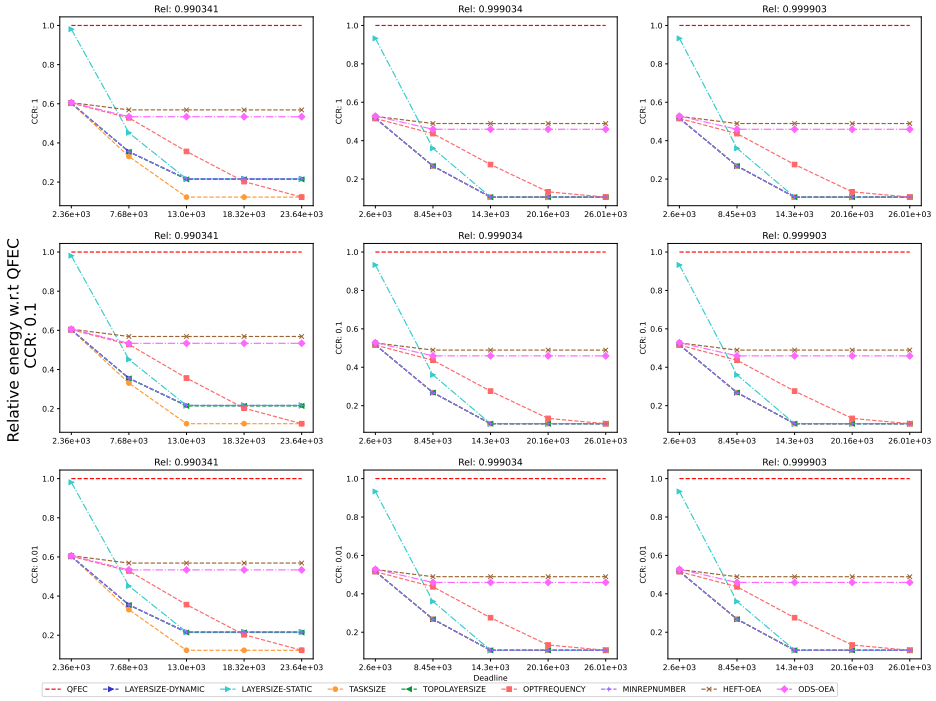

Figure 793: Performance of the different heuristics on the Cholesky workflow.

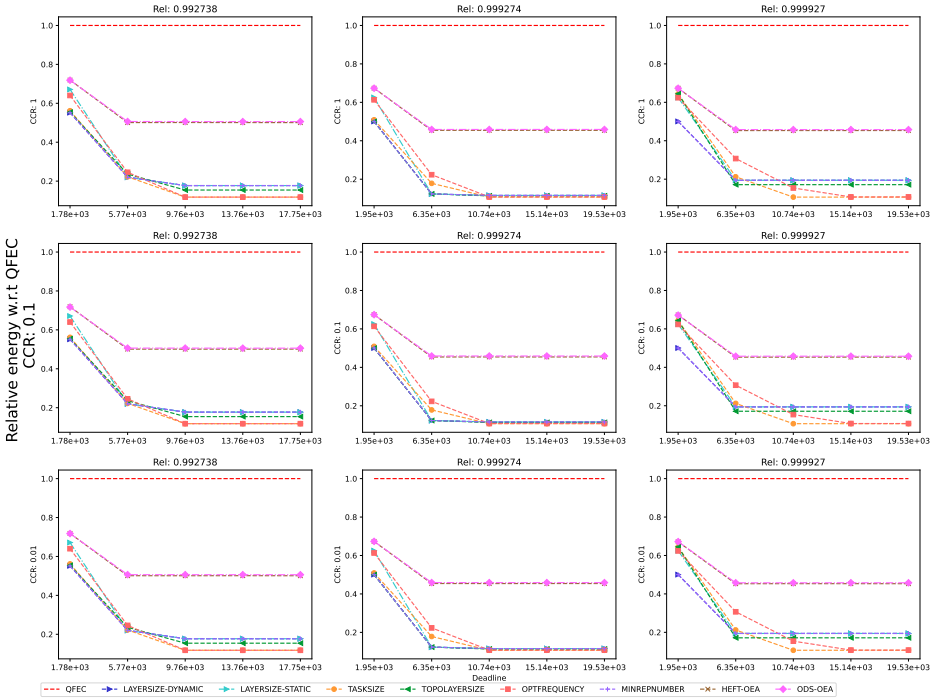

Figure 794: Performance of the different heuristics on the Cycles workflow.

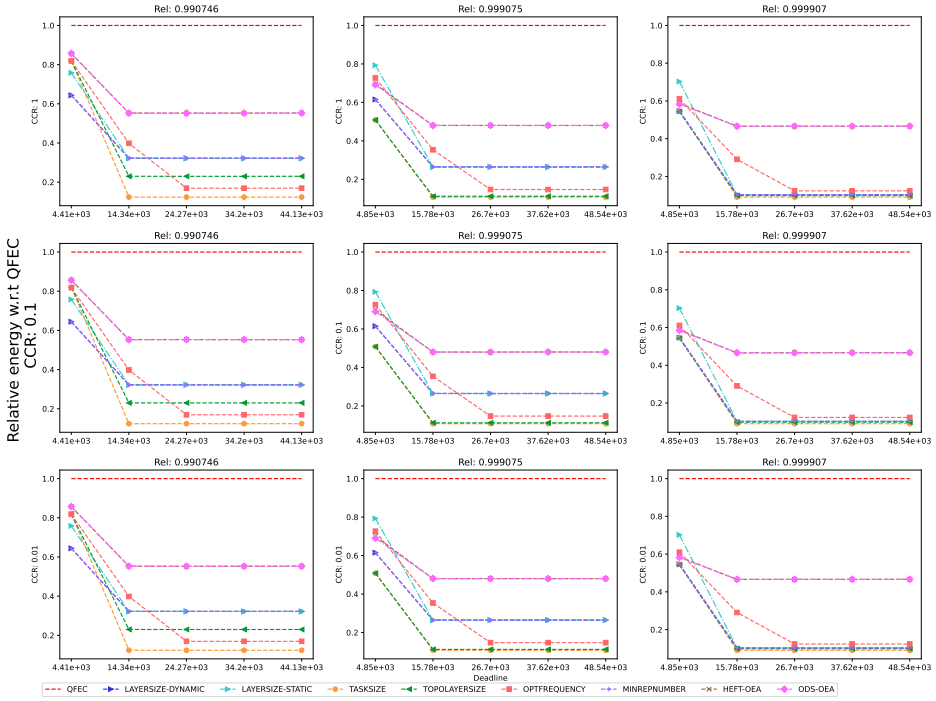

Figure 795: Performance of the different heuristics on the Epigenomics workflow.

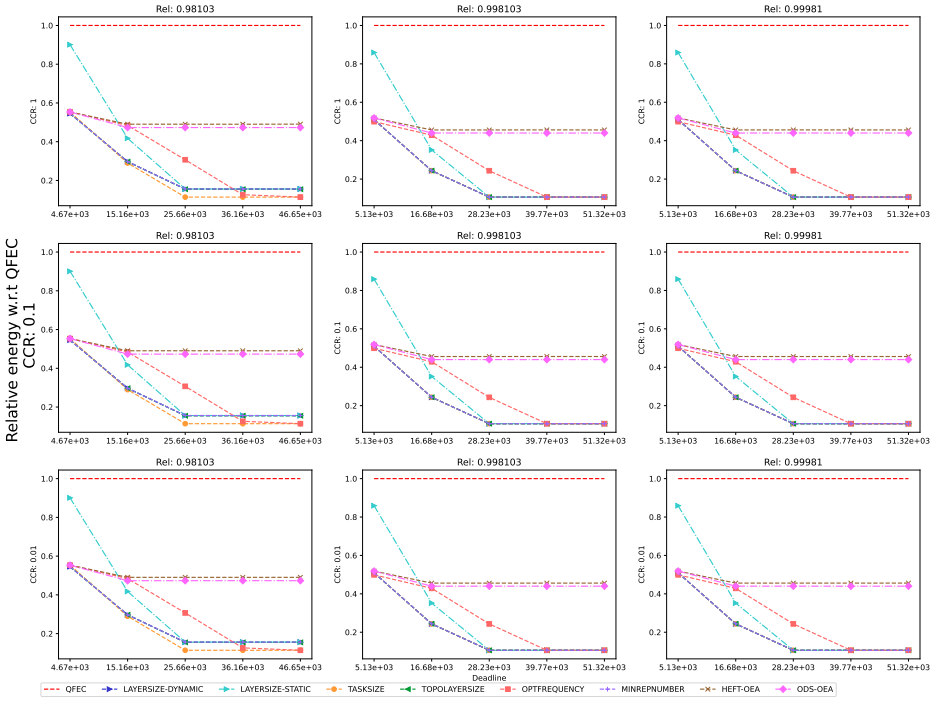

Figure 796: Performance of the different heuristics on the LU workflow.

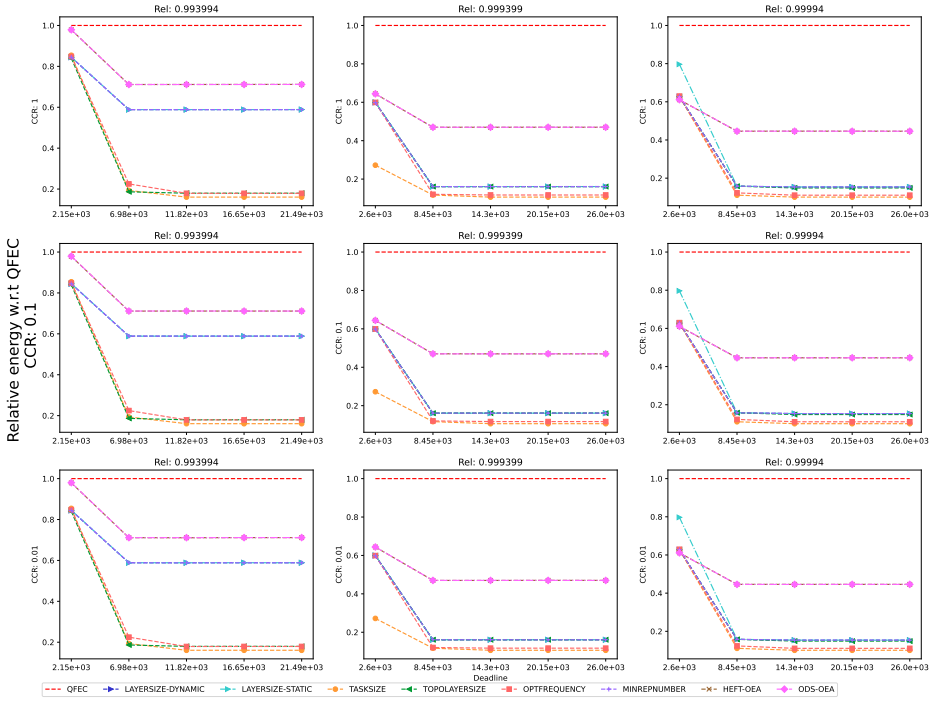

Figure 797: Performance of the different heuristics on the Montage workflow.

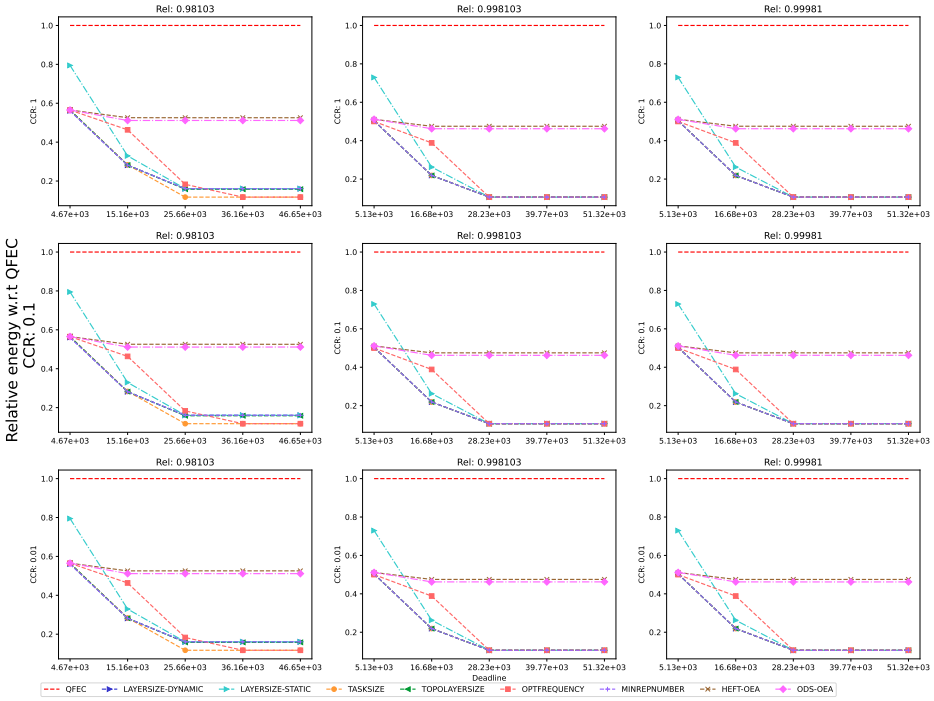

Figure 798: Performance of the different heuristics on the QR workflow.

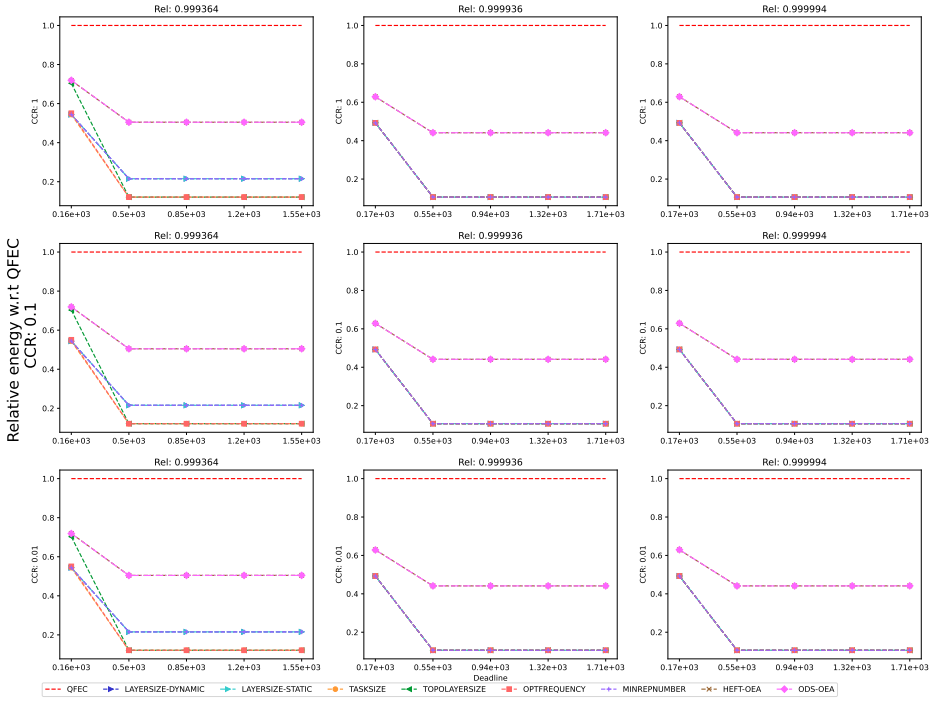

Figure 799: Performance of the different heuristics on the Seismology workflow.

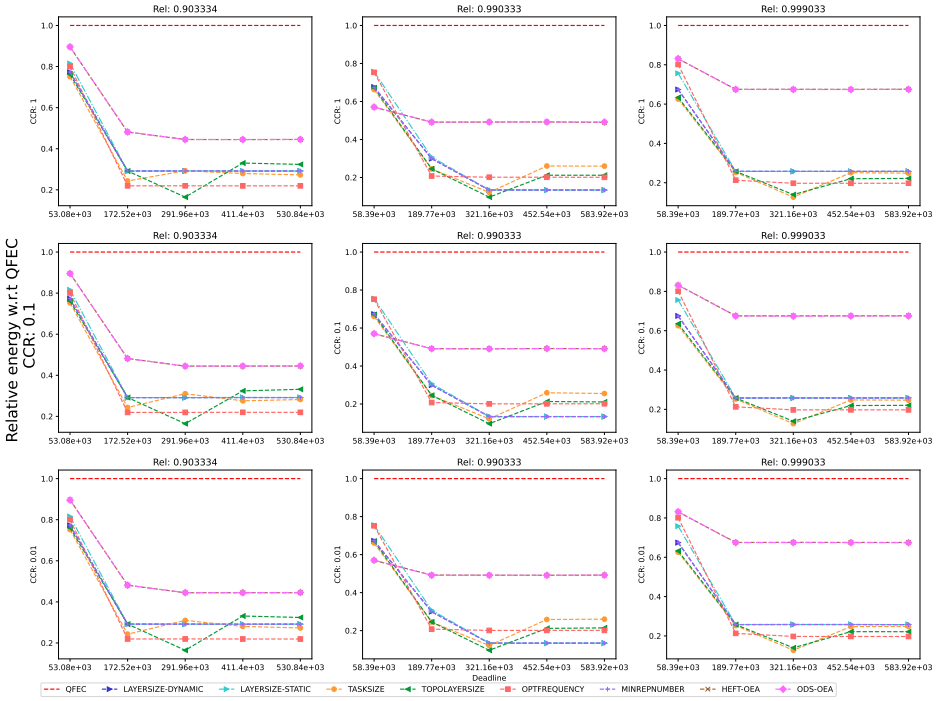

Figure 800: Performance of the different heuristics on the SoyKB workflow.

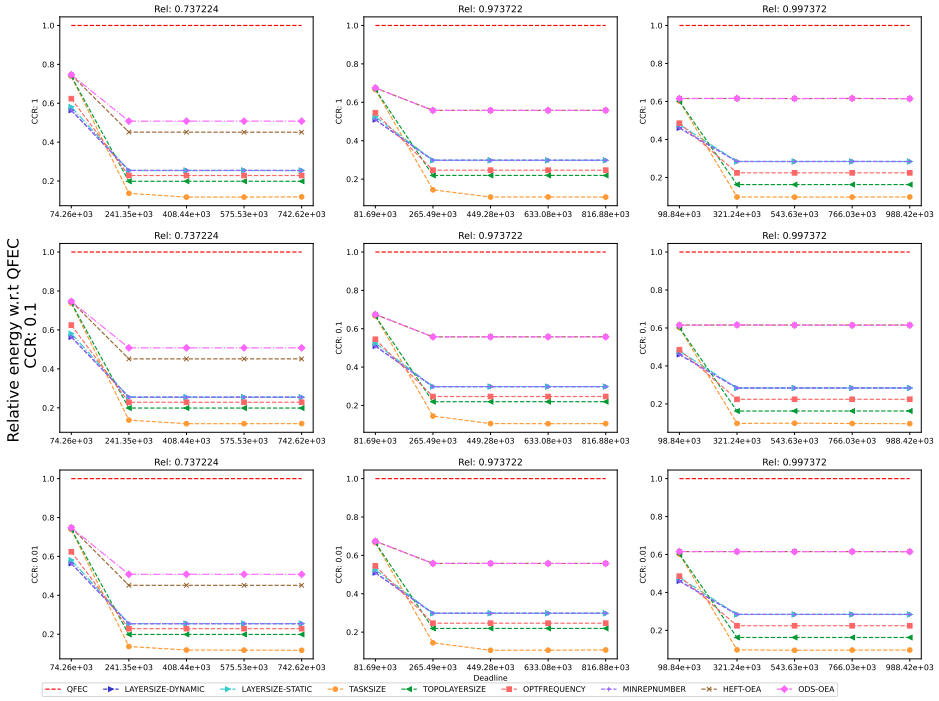

Figure 801: Performance of the different heuristics on the SRASearch.

F Study of ExtMILP when actual execution times are drawn from a truncated normal distribution

F.1 $BC/WC = 0.1$

F.1.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

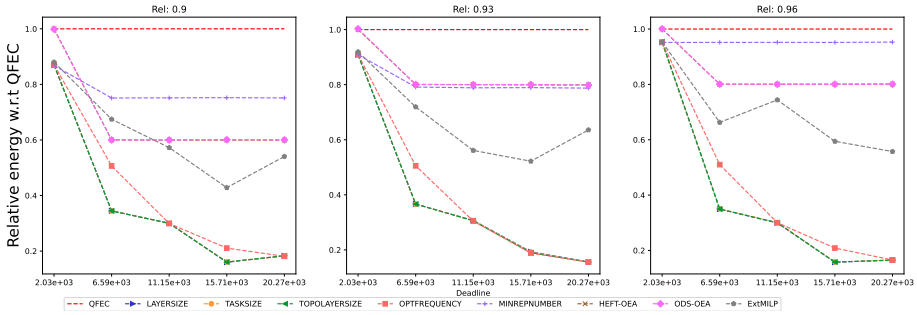

Figure 802: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

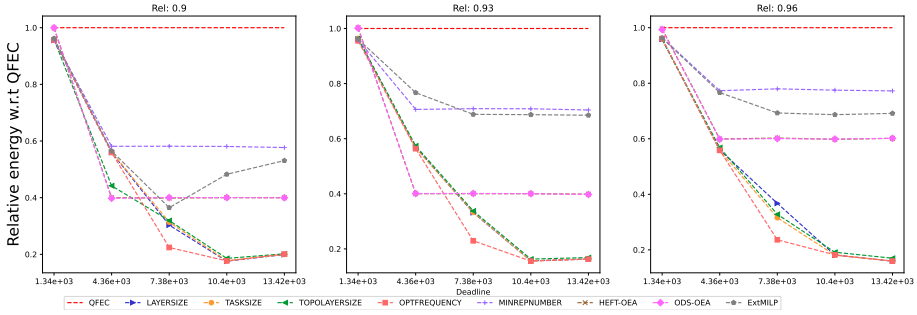

Figure 803: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

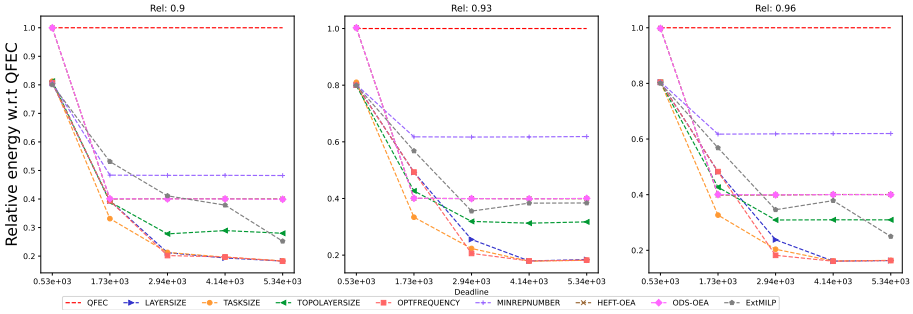

Figure 804: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

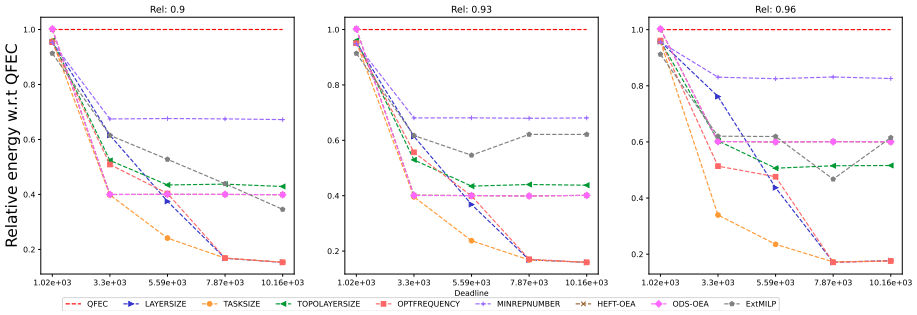

Figure 805: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

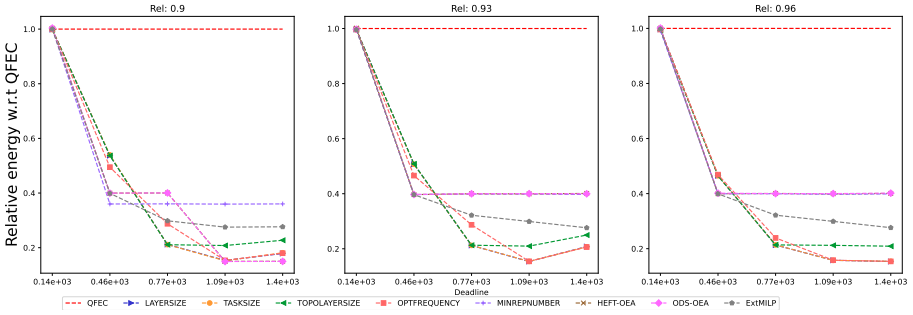

Figure 806: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

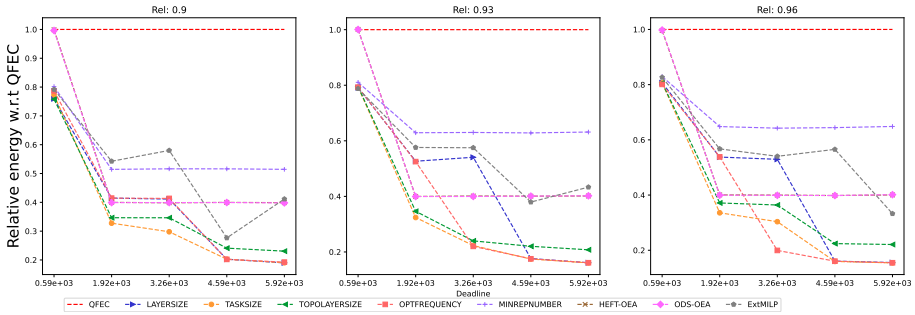

Figure 807: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

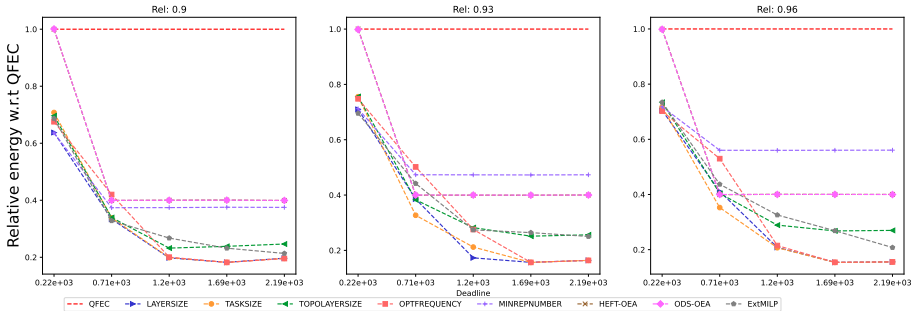

Figure 808: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

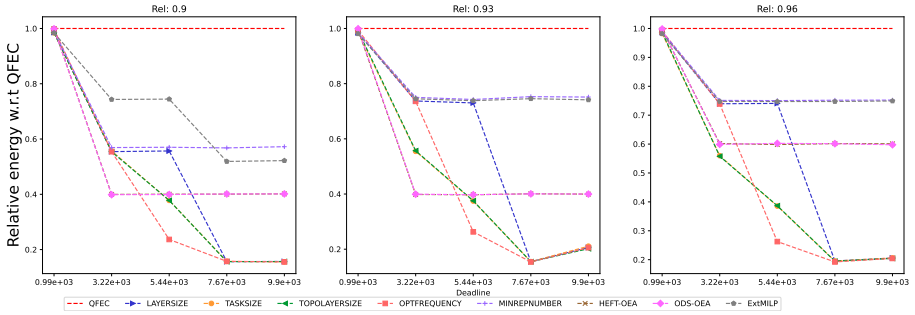

Figure 809: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

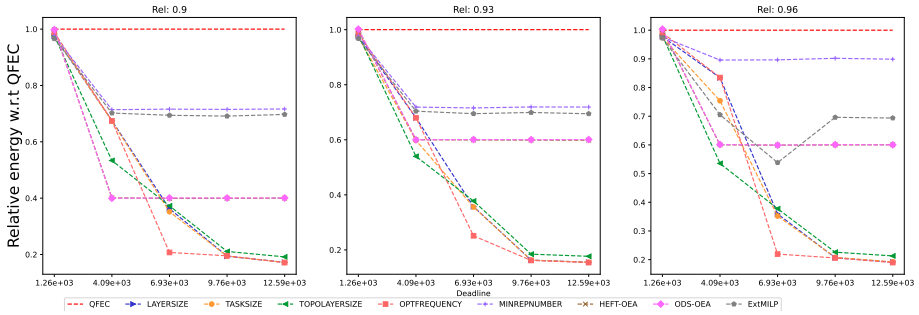

Figure 810: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

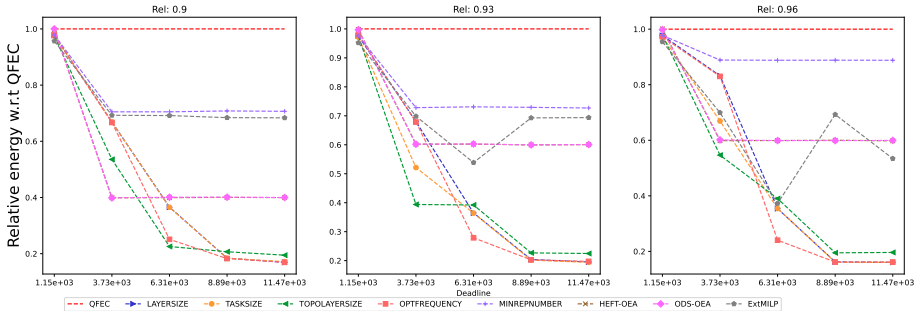

Figure 811: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

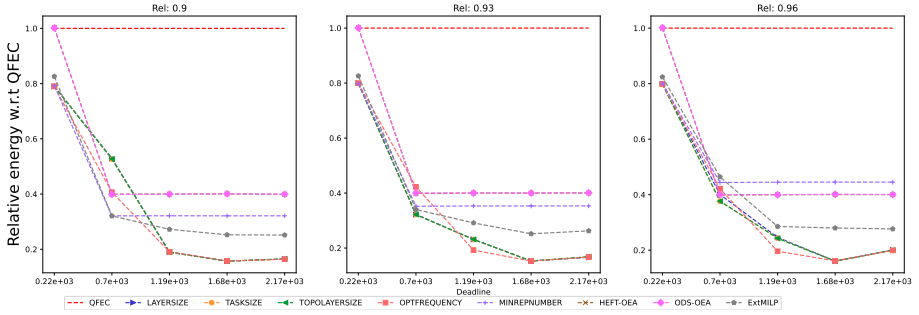

Figure 812: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

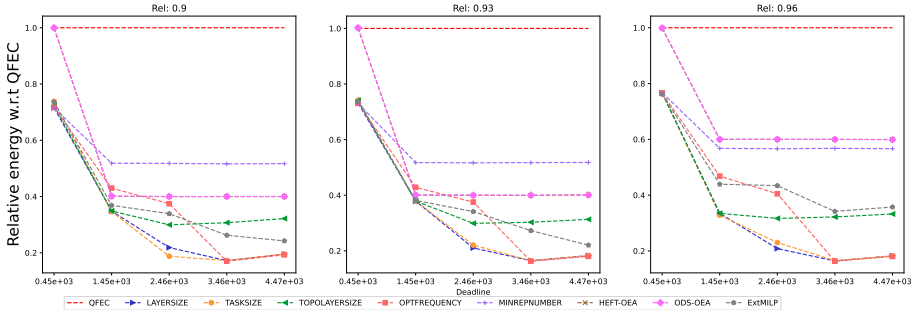

Figure 813: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

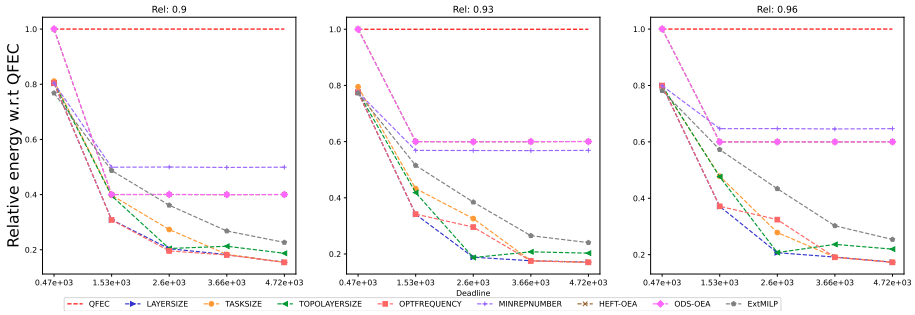

Figure 814: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

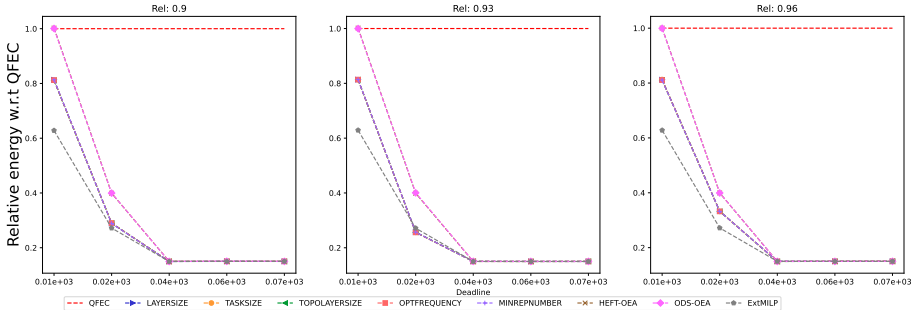

Figure 815: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

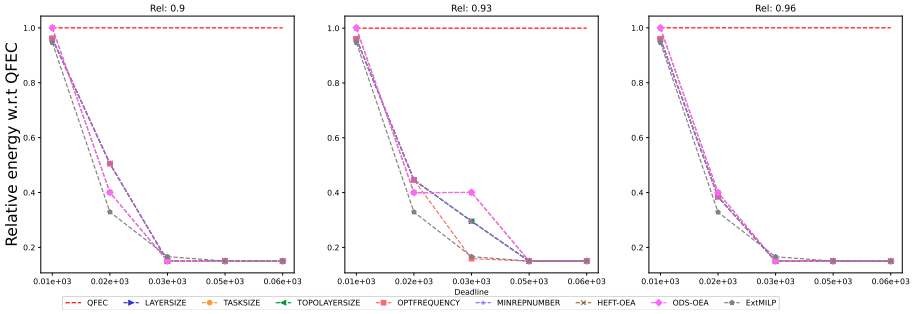

Figure 816: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

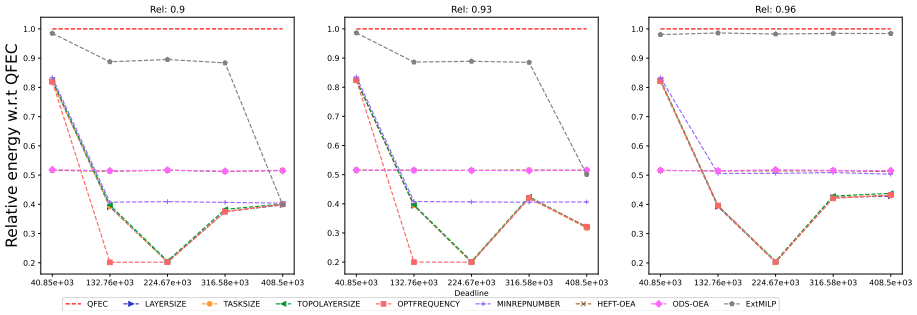

Figure 817: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

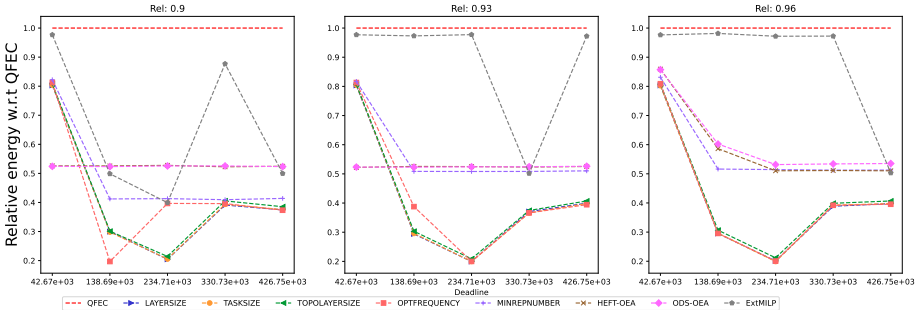

Figure 818: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

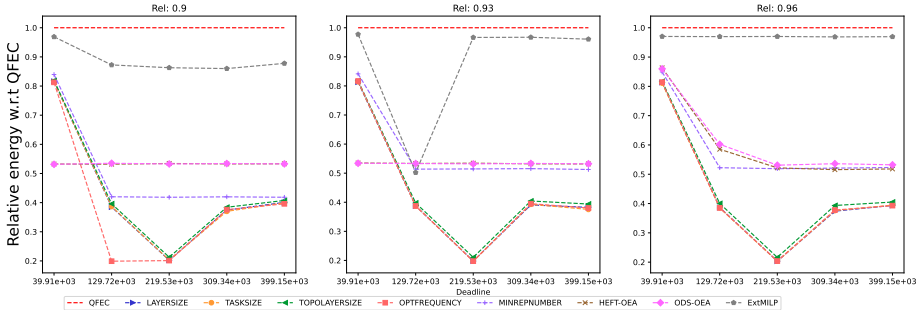

Figure 819: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

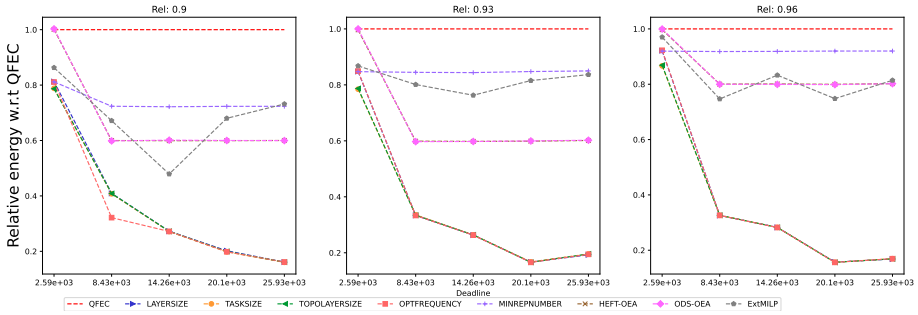

Figure 820: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

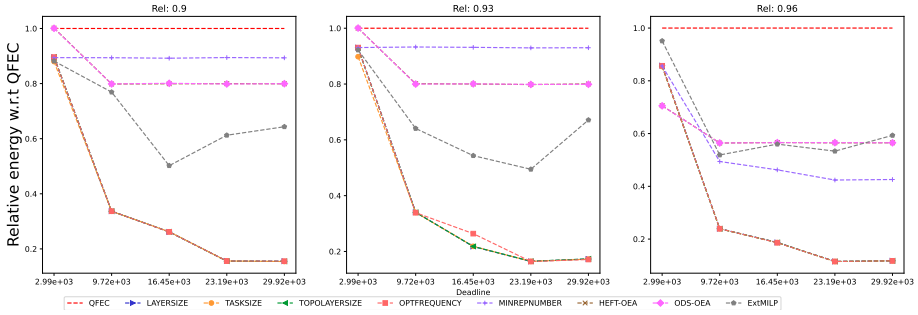

Figure 821: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.1.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

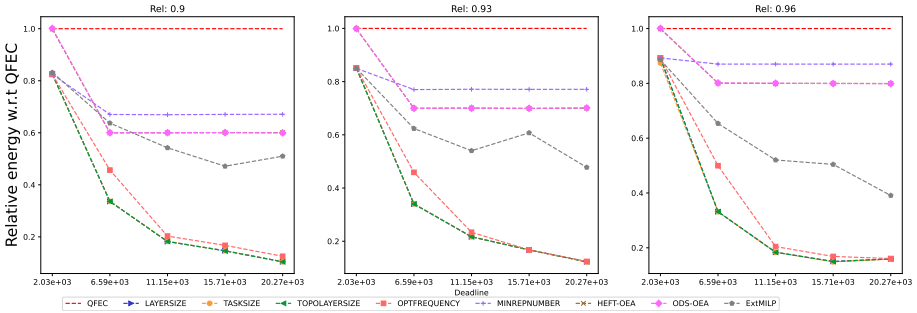

Figure 822: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

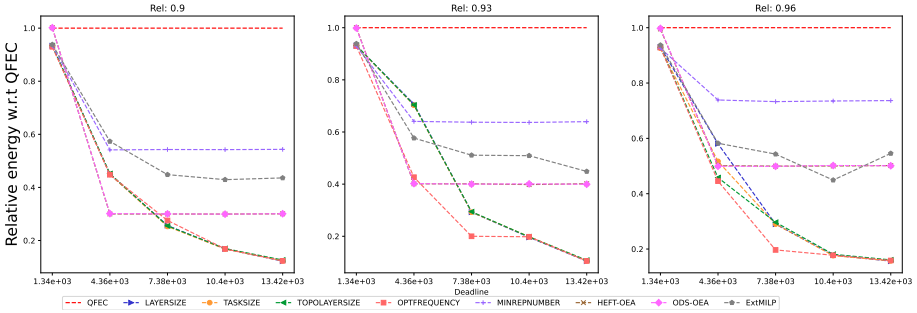

Figure 823: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

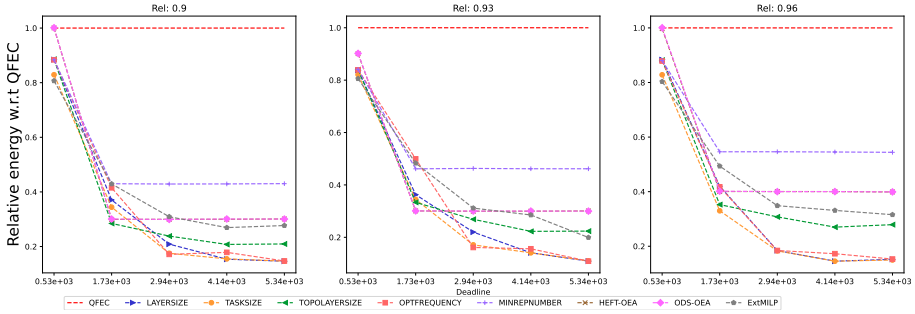

Figure 824: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

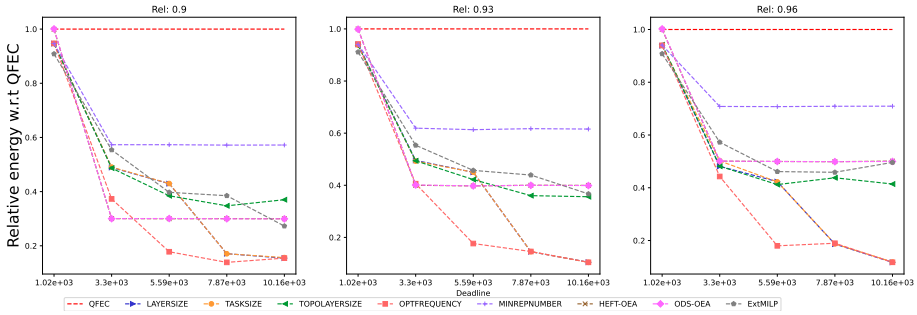

Figure 825: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

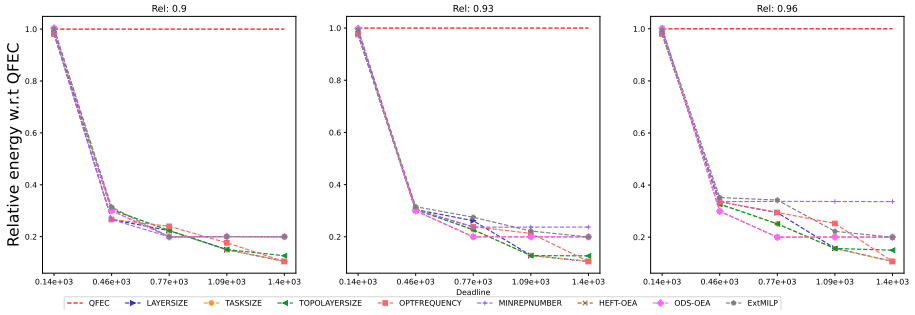

Figure 826: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

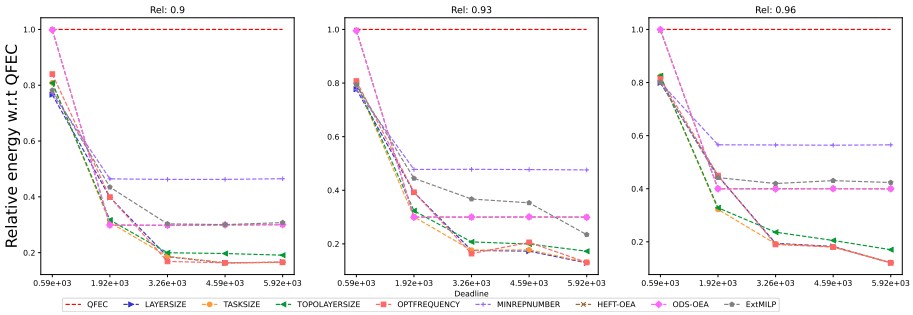

Figure 827: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

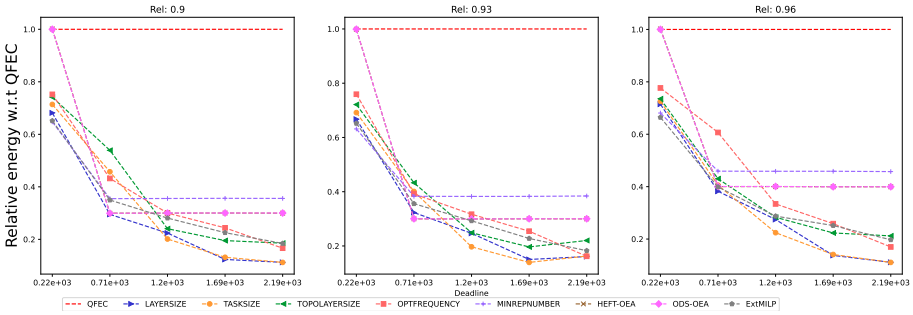

Figure 828: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

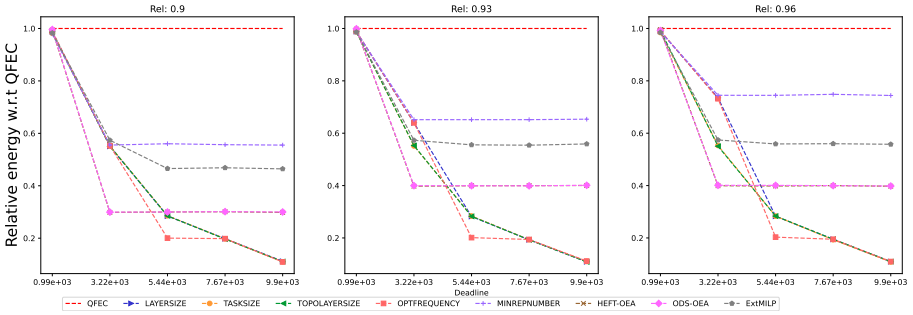

Figure 829: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

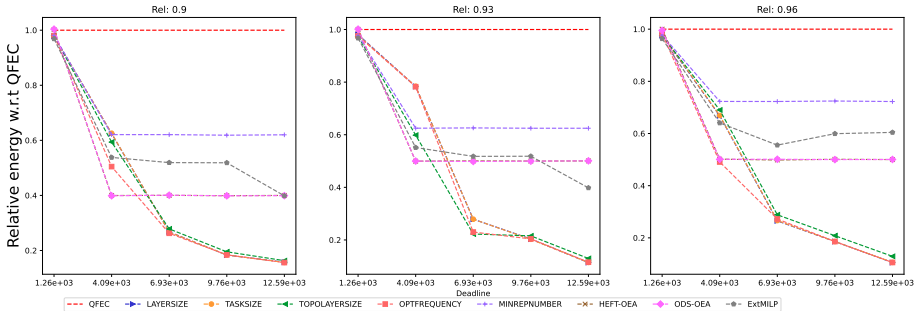

Figure 830: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

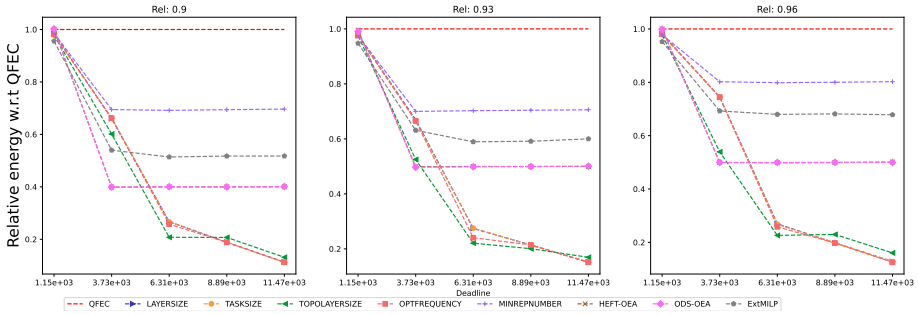

Figure 831: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

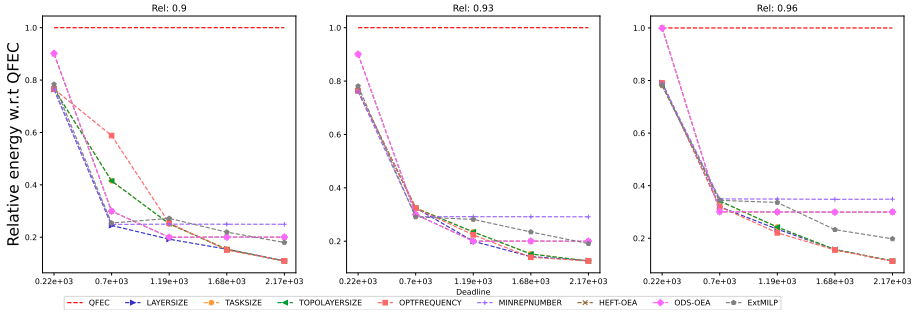

Figure 832: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

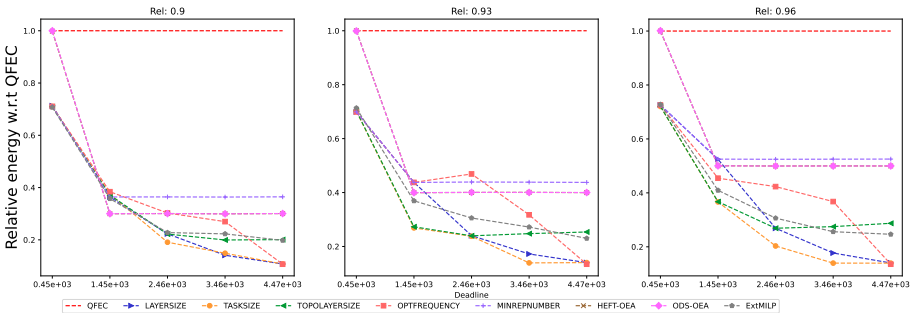

Figure 833: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

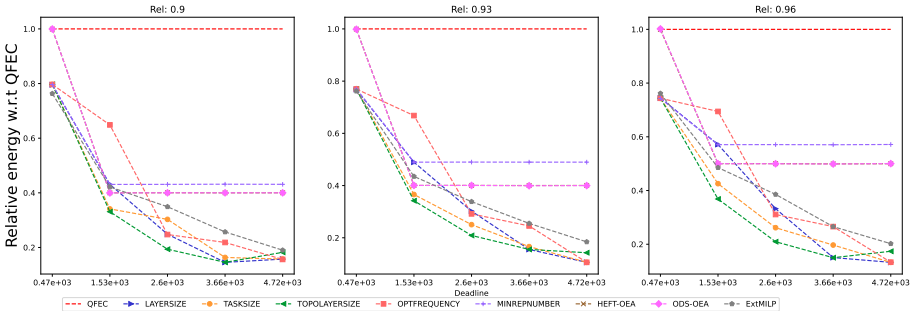

Figure 834: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

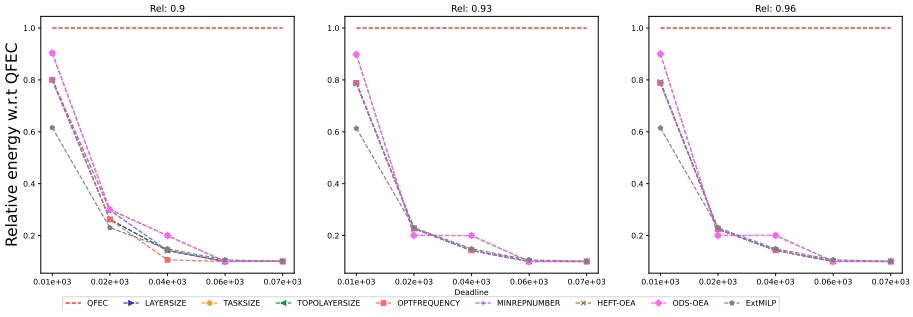

Figure 835: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

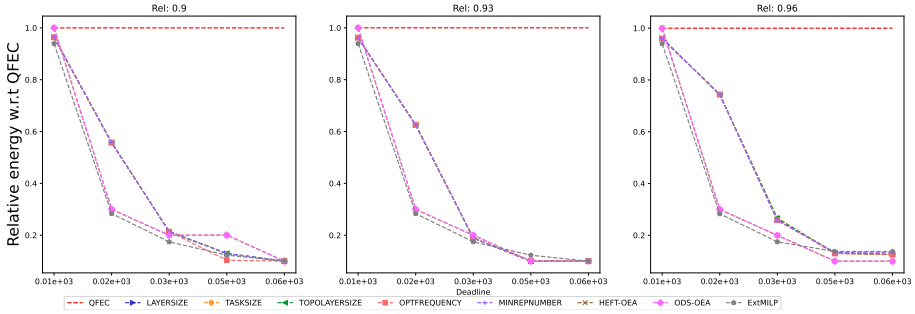

Figure 836: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

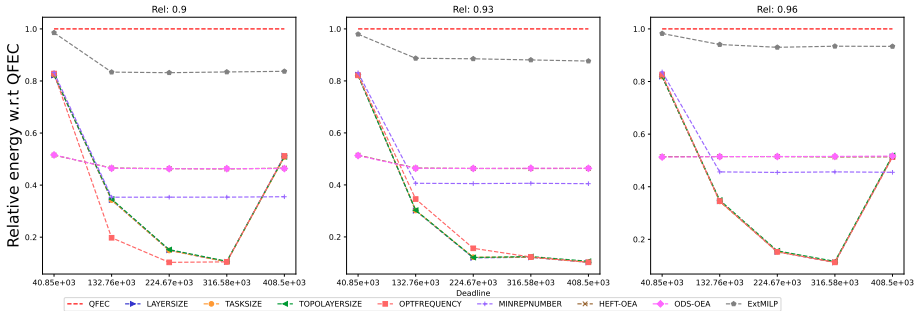

Figure 837: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

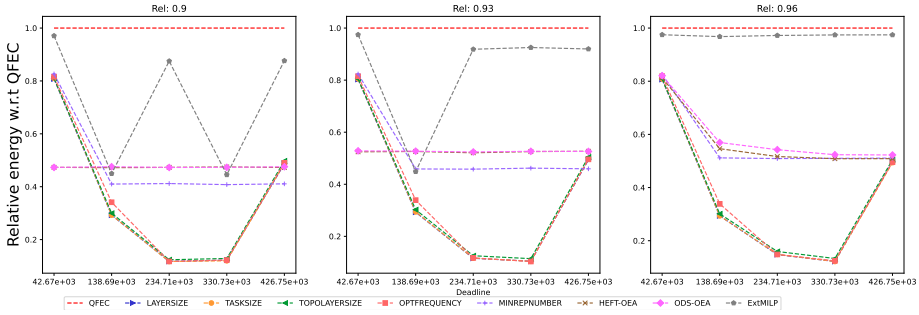

Figure 838: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

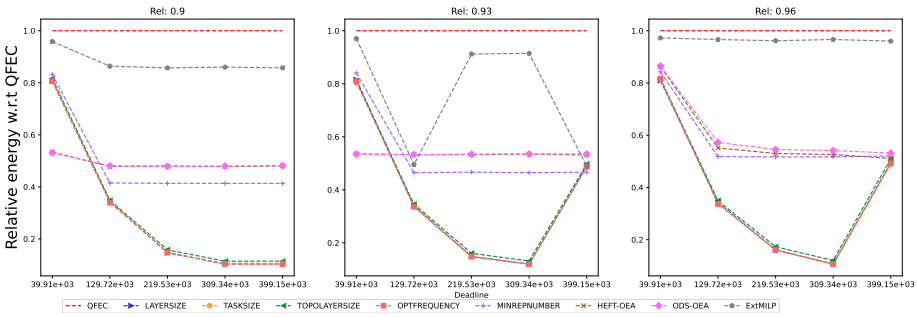

Figure 839: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

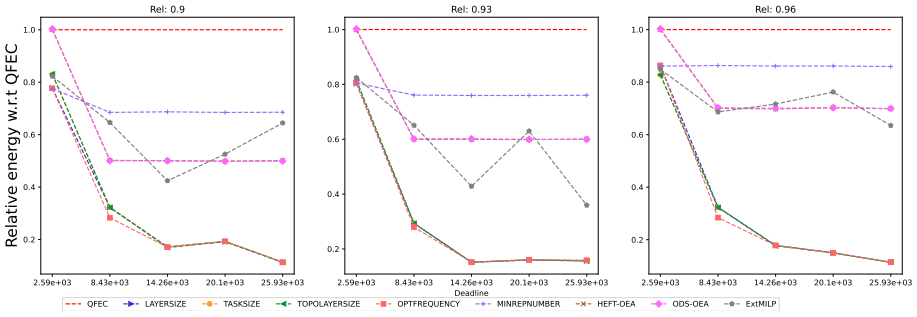

Figure 840: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

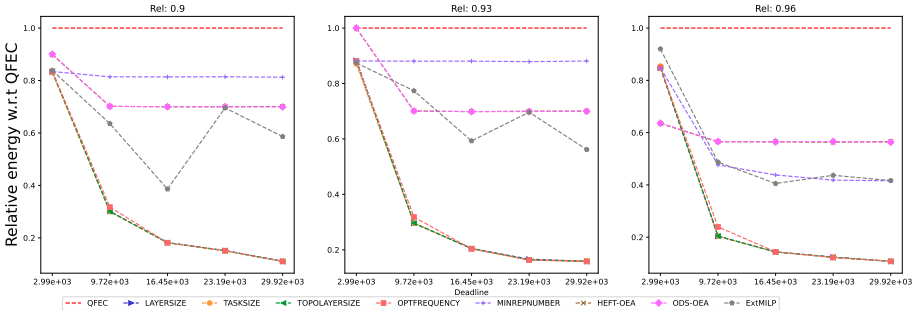

Figure 841: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.1.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

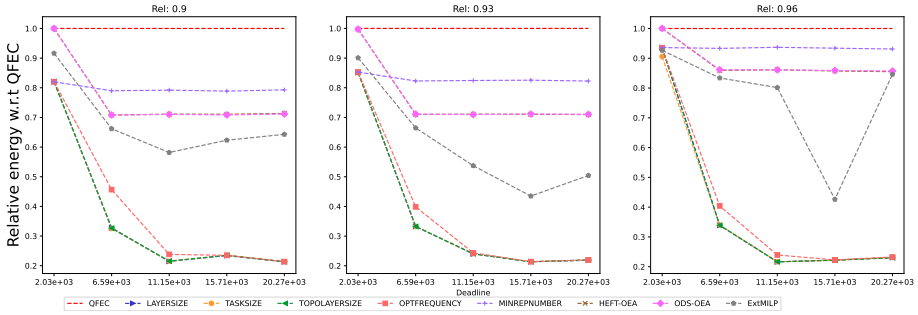

Figure 842: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

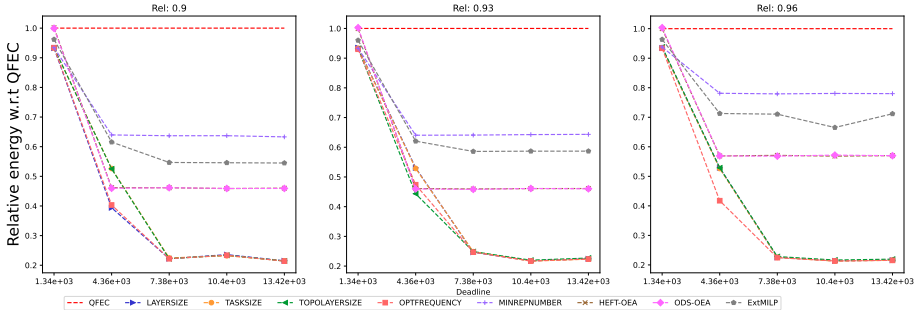

Figure 843: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

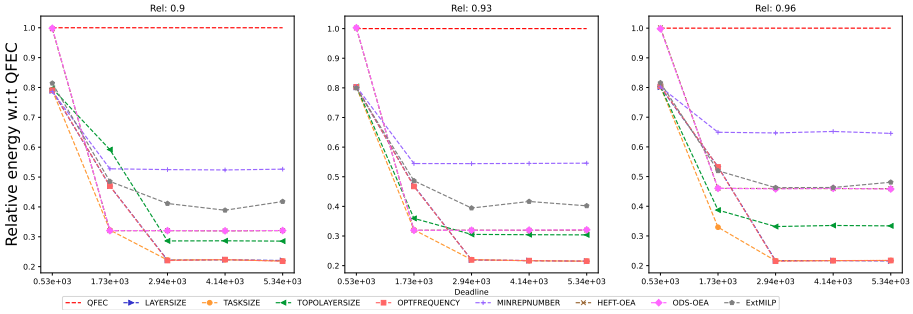

Figure 844: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

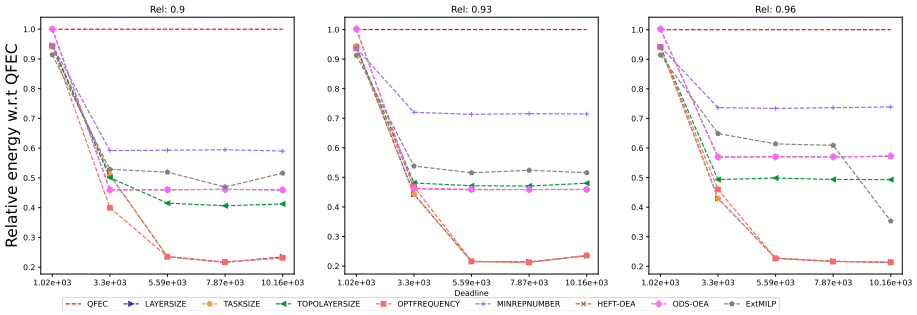

Figure 845: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

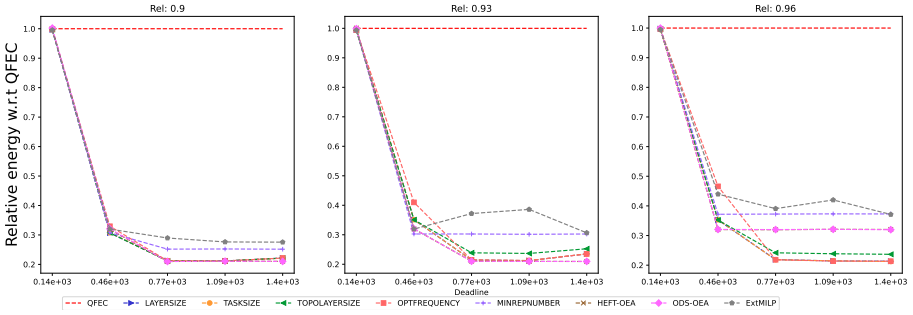

Figure 846: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

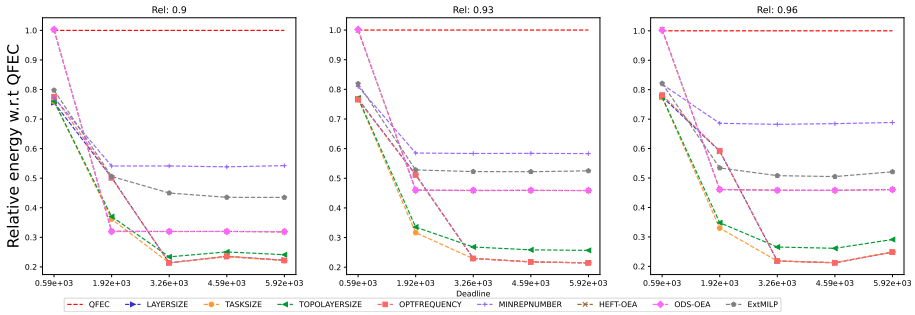

Figure 847: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

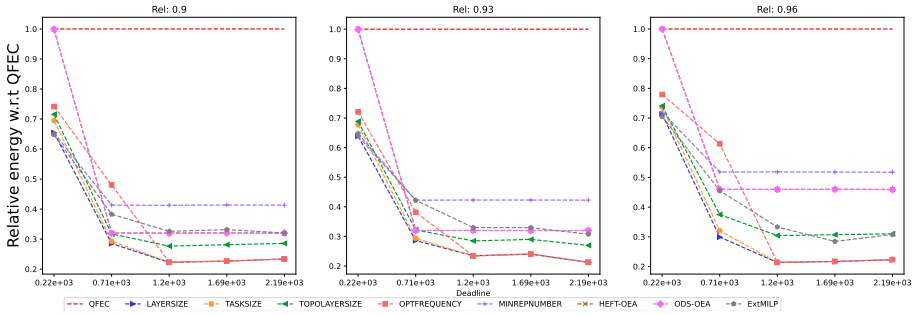

Figure 848: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

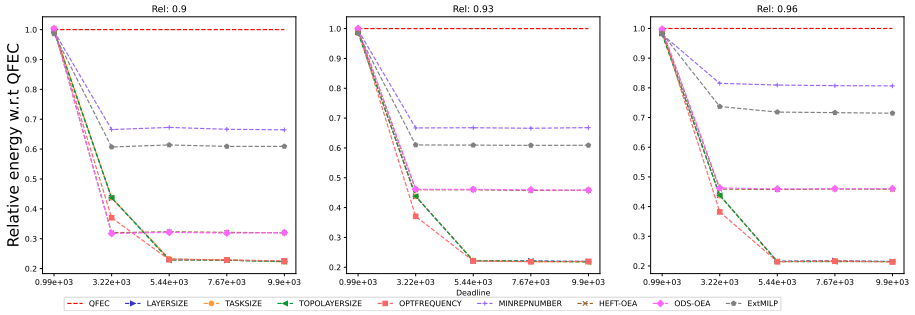

Figure 849: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

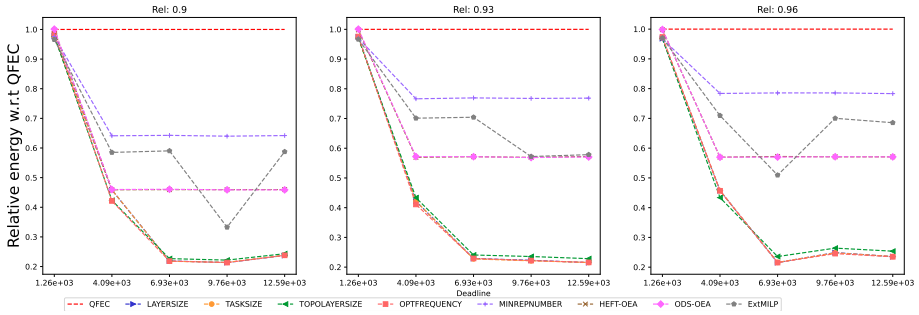

Figure 850: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

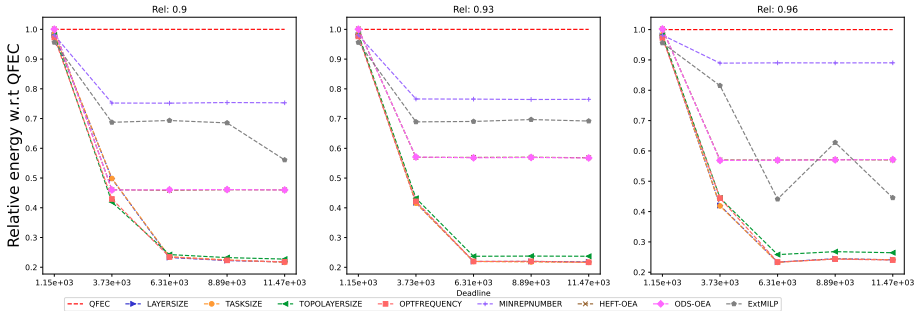

Figure 851: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

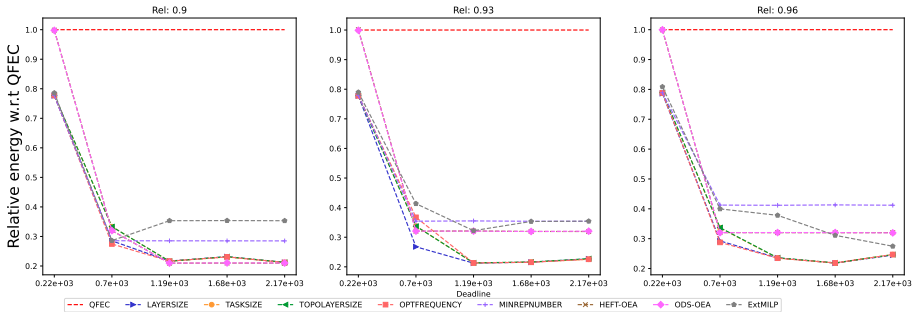

Figure 852: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

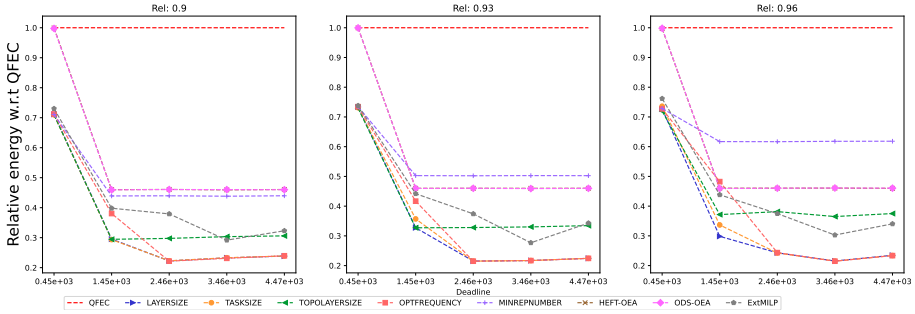

Figure 853: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

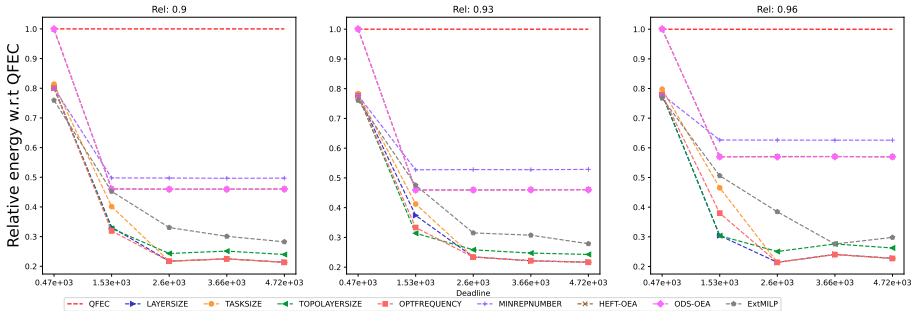

Figure 854: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

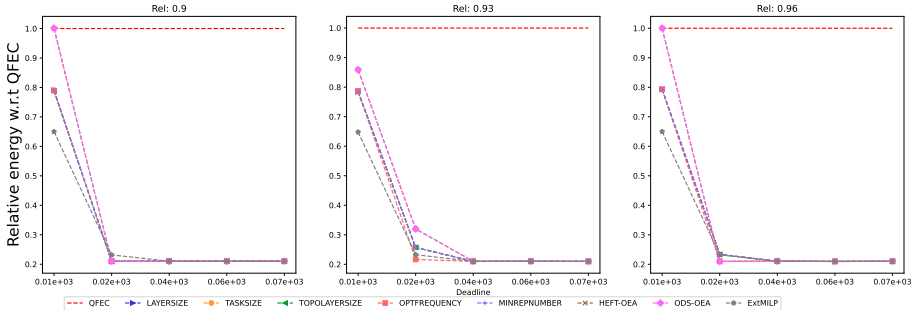

Figure 855: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

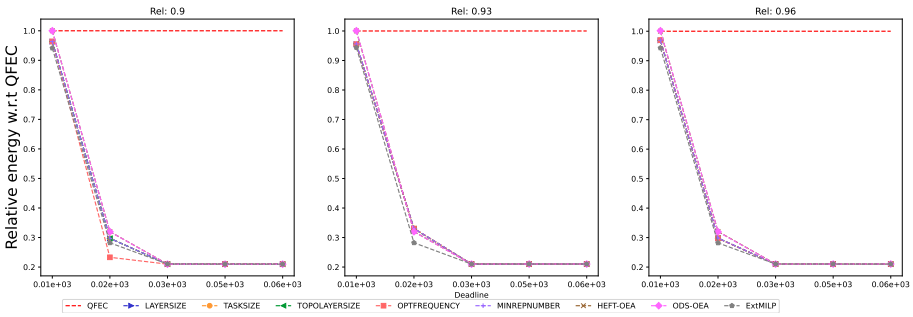

Figure 856: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

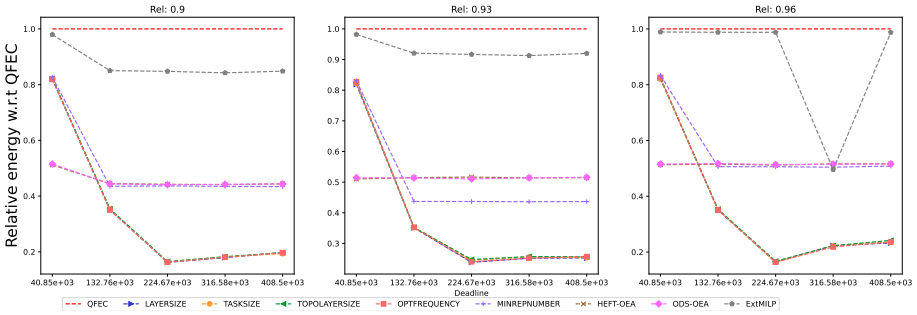

Figure 857: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

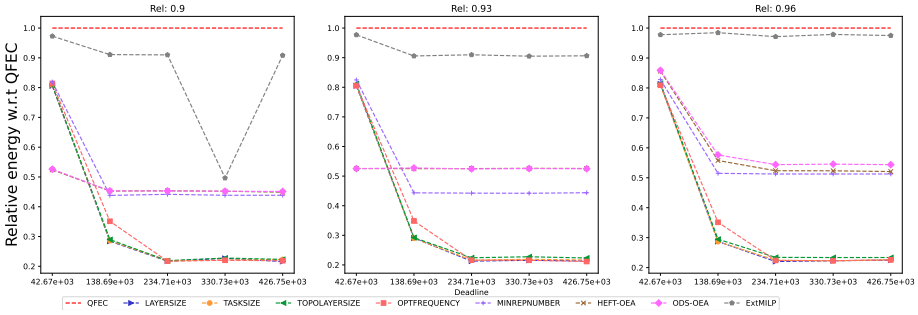

Figure 858: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

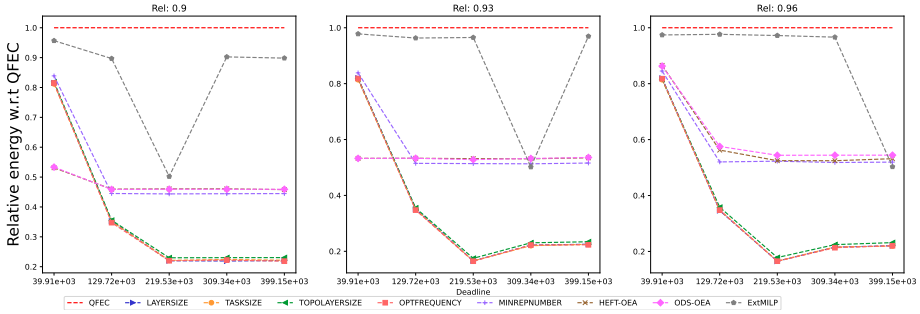

Figure 859: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

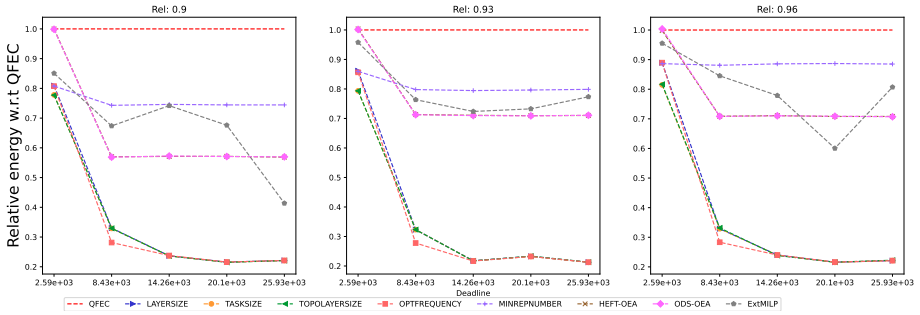

Figure 860: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

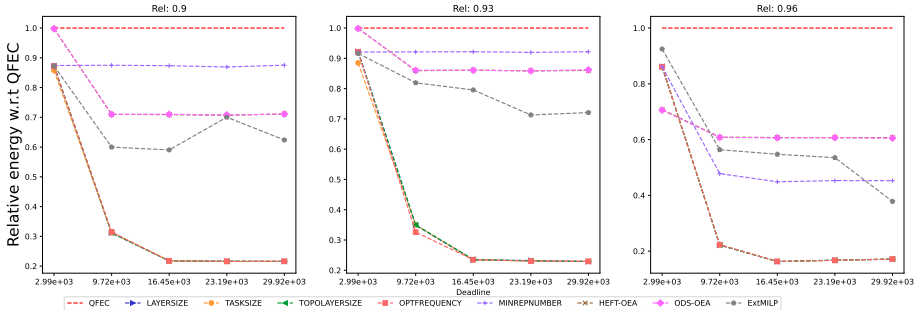

Figure 861: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.1.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

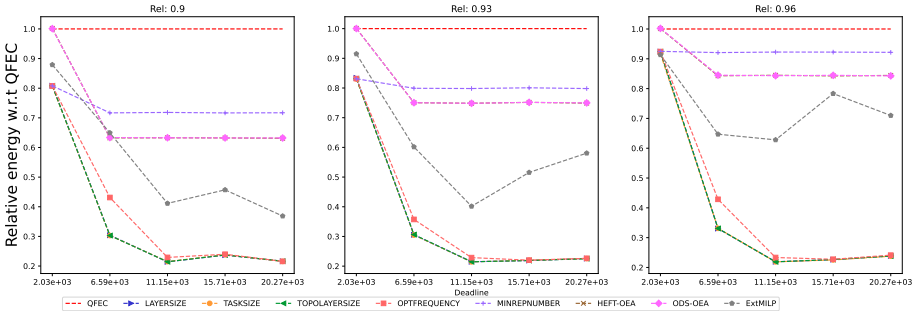

Figure 862: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

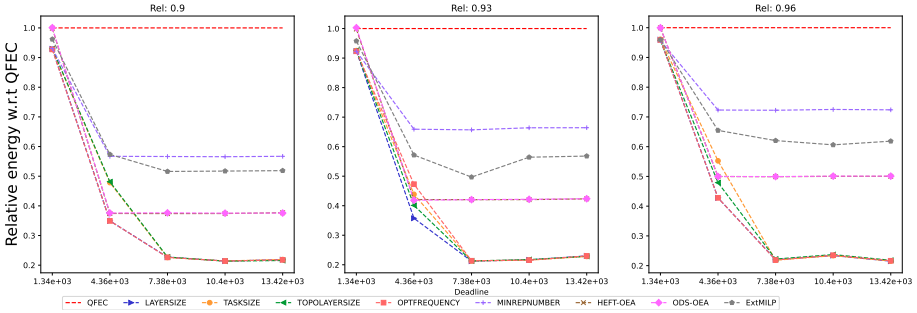

Figure 863: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

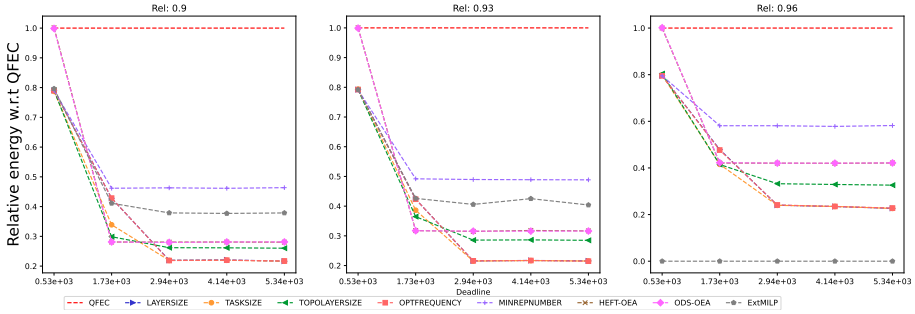

Figure 864: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

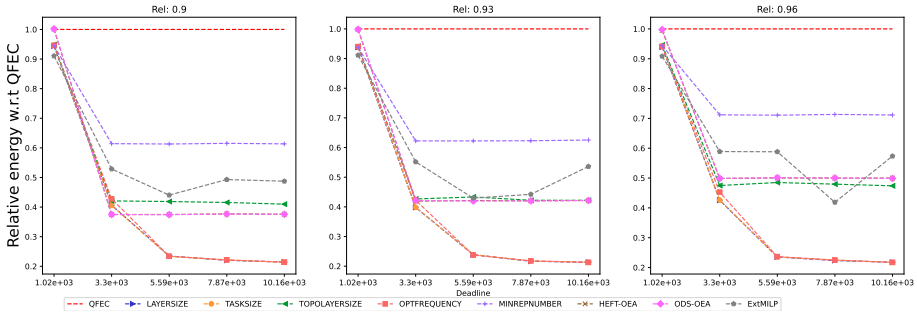

Figure 865: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

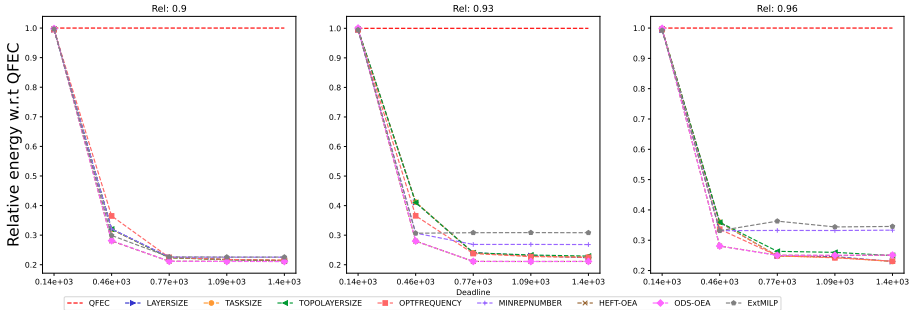

Figure 866: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

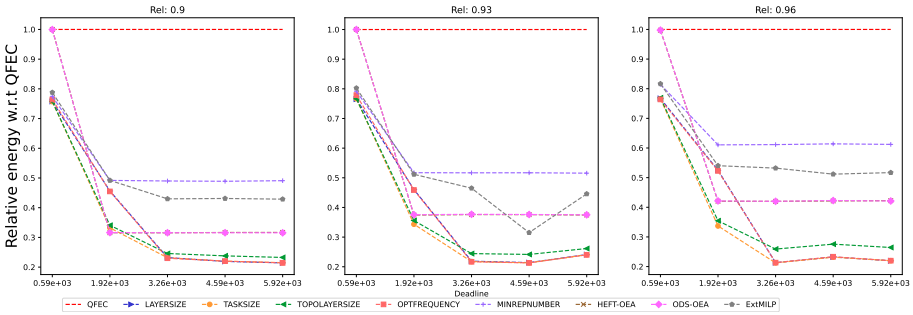

Figure 867: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

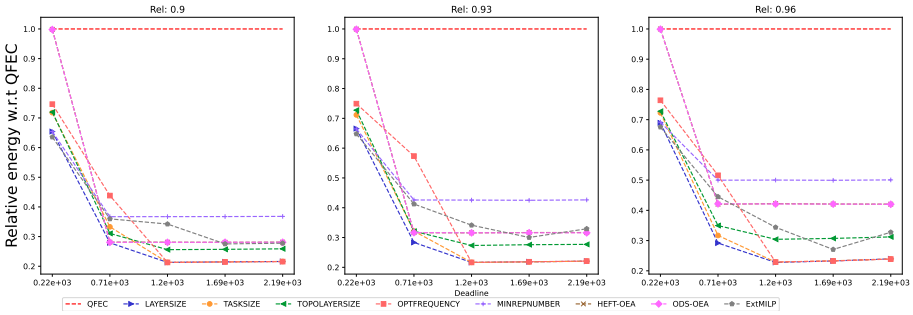

Figure 868: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

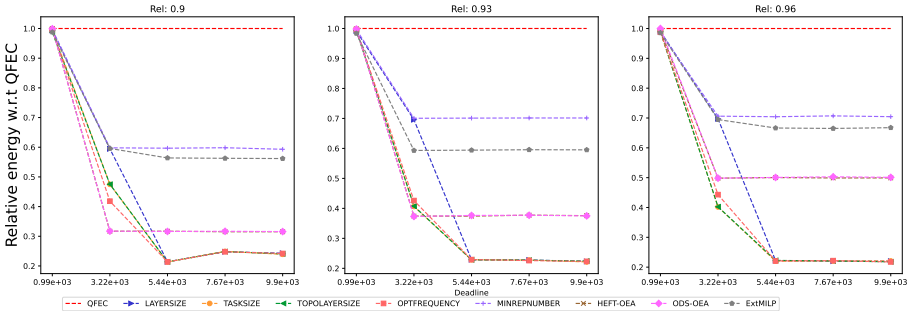

Figure 869: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

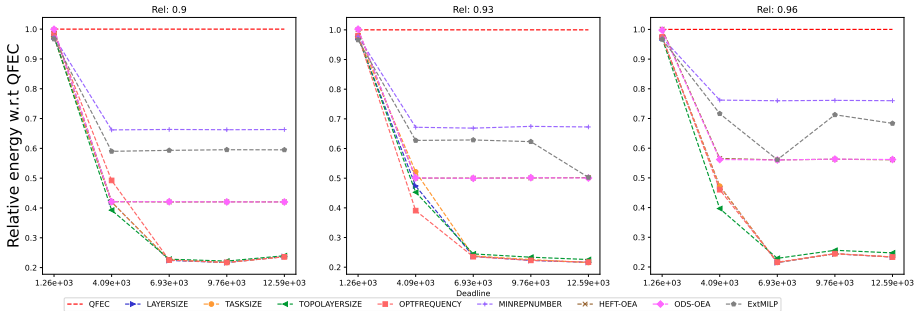

Figure 870: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

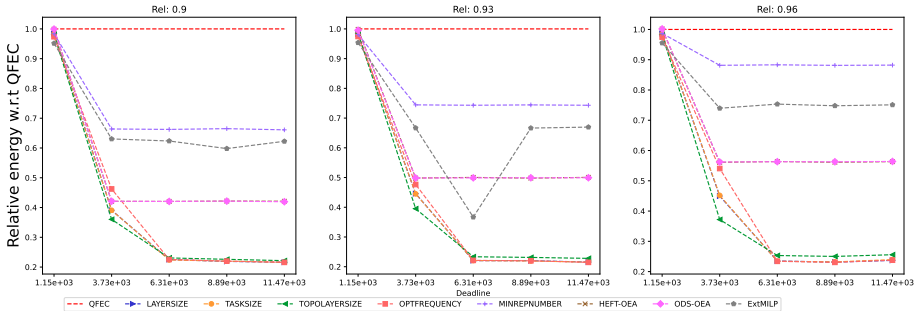

Figure 871: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

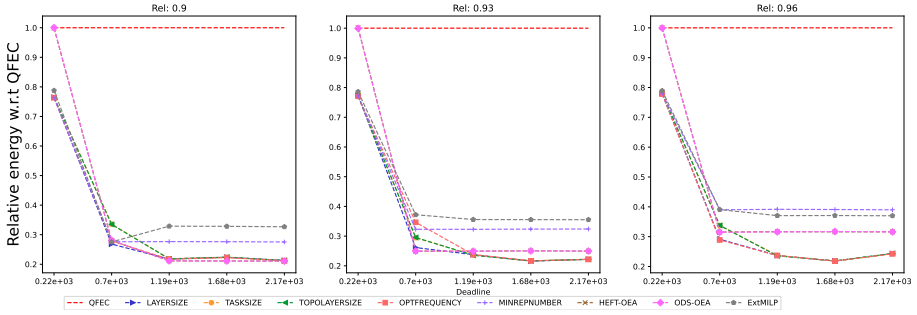

Figure 872: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

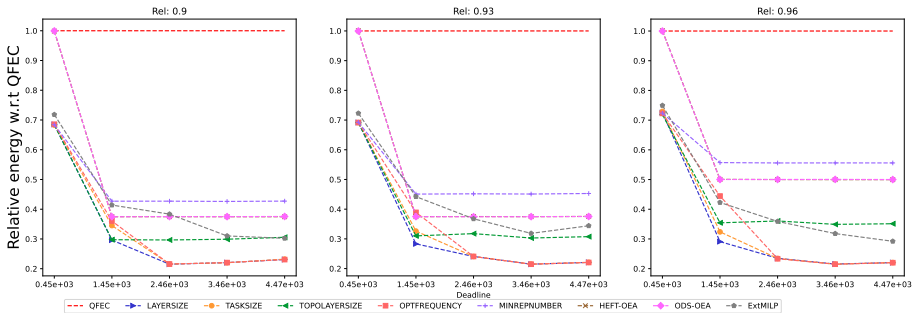

Figure 873: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

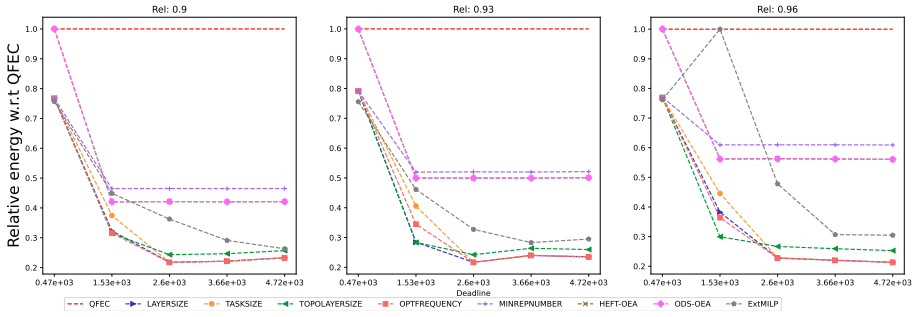

Figure 874: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

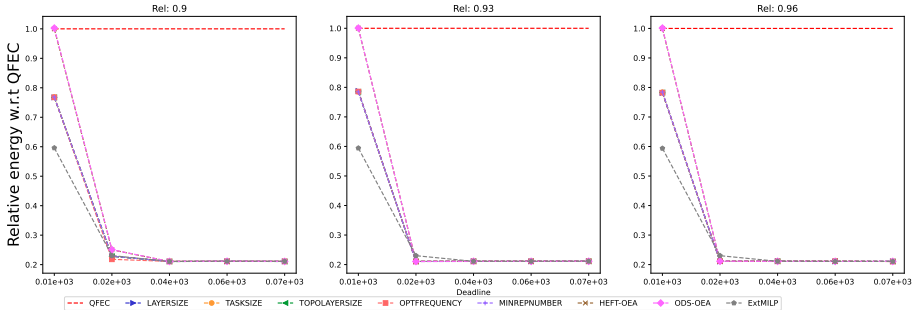

Figure 875: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

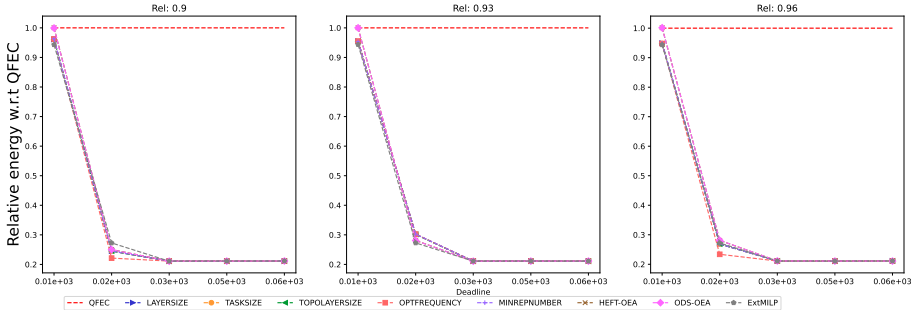

Figure 876: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

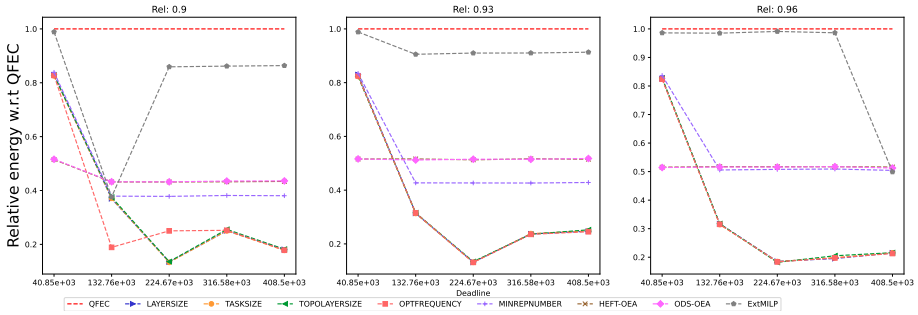

Figure 877: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

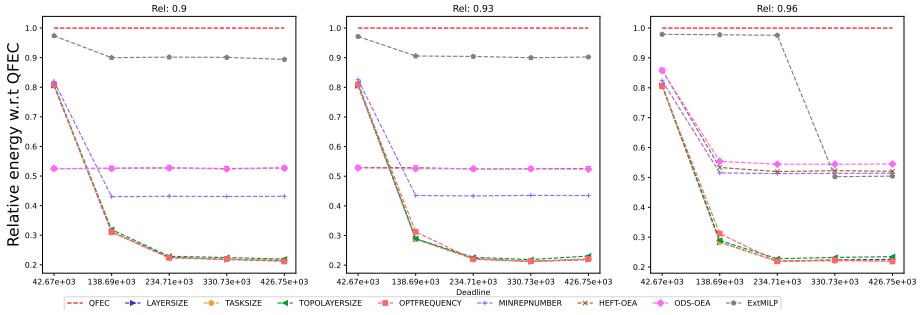

Figure 878: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

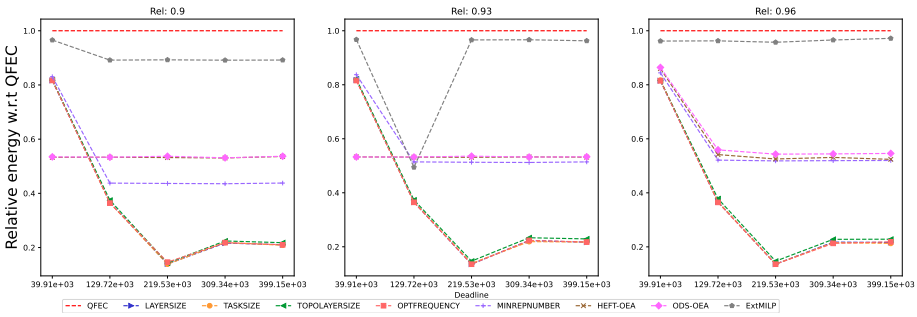

Figure 879: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

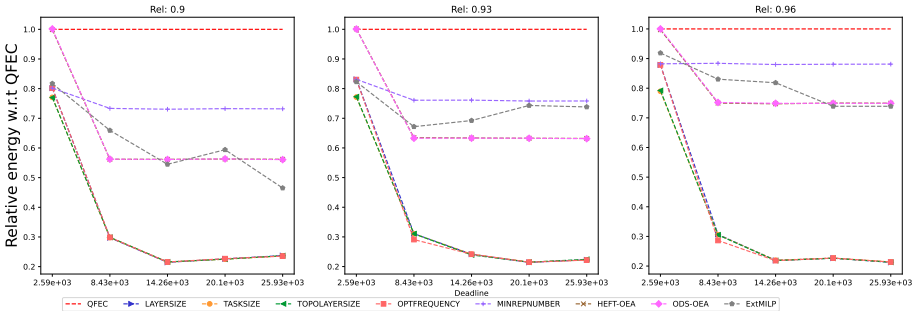

Figure 880: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

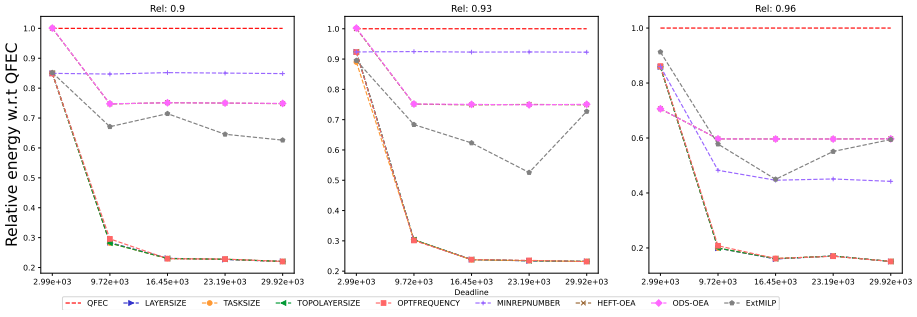

Figure 881: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.2 $BC/WC = 0.2$

F.2.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

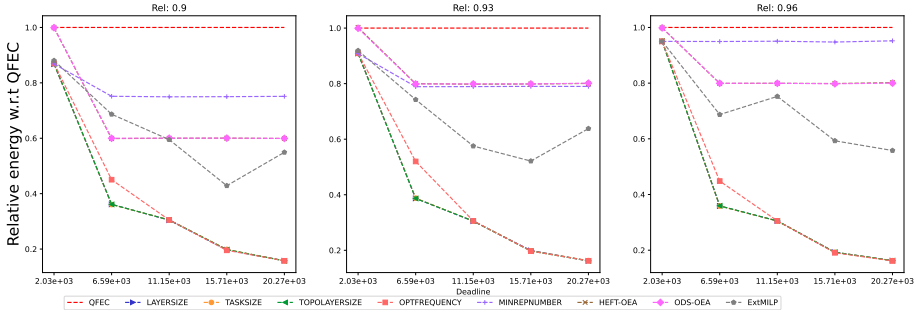

Figure 882: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

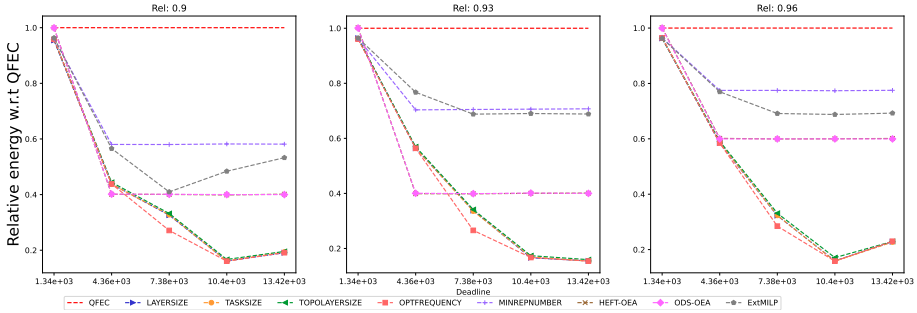

Figure 883: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

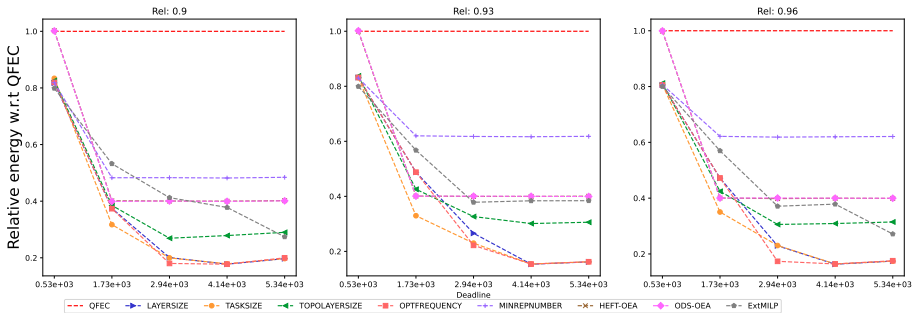

Figure 884: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

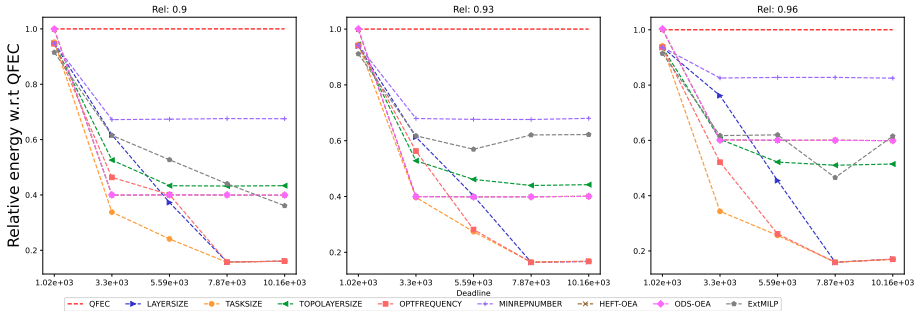

Figure 885: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

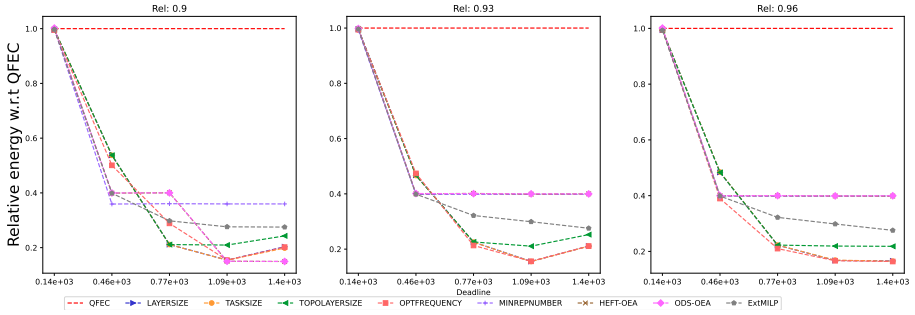

Figure 886: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

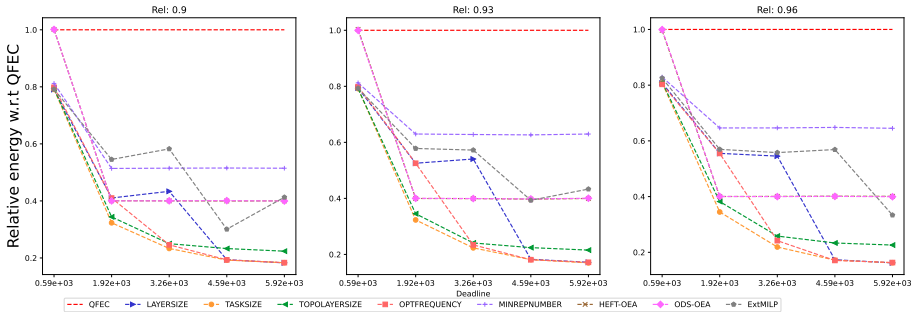

Figure 887: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

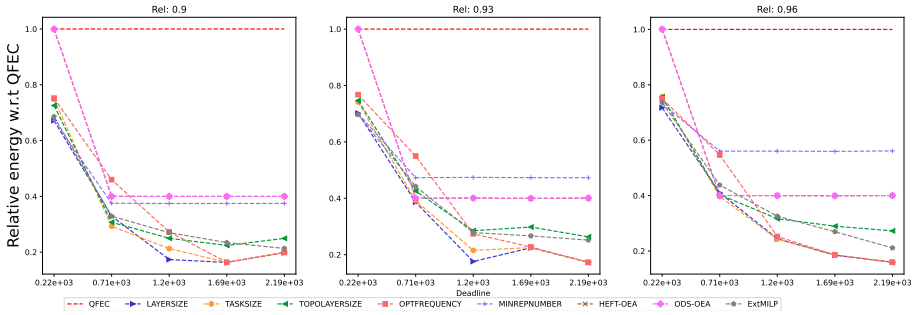

Figure 888: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

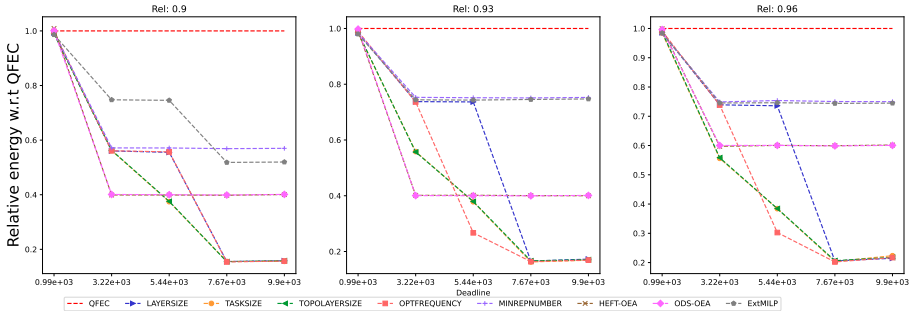

Figure 889: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

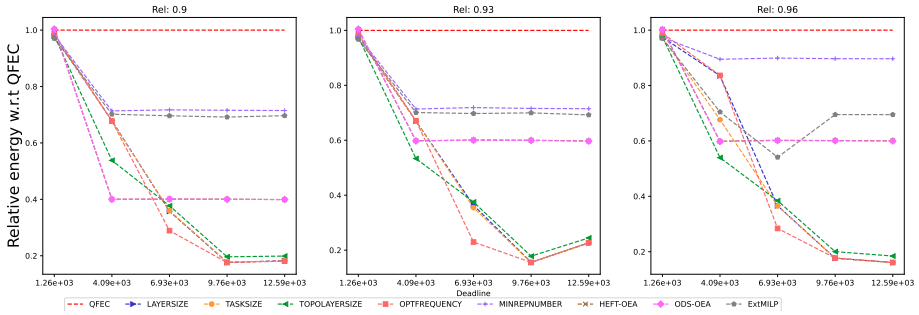

Figure 890: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

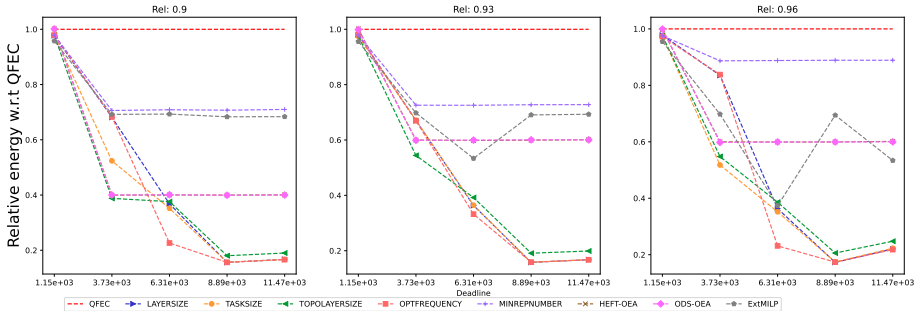

Figure 891: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

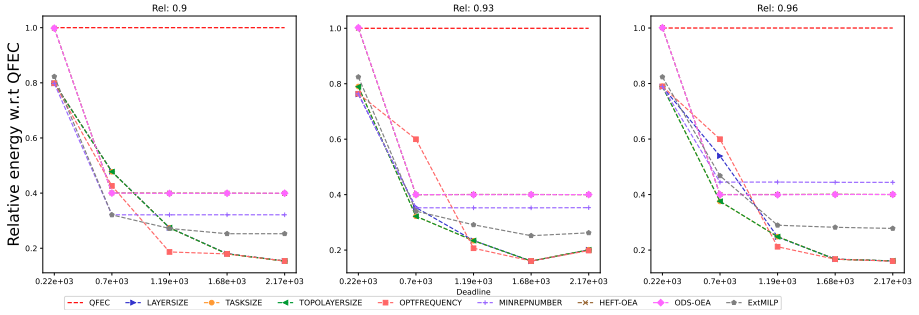

Figure 892: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

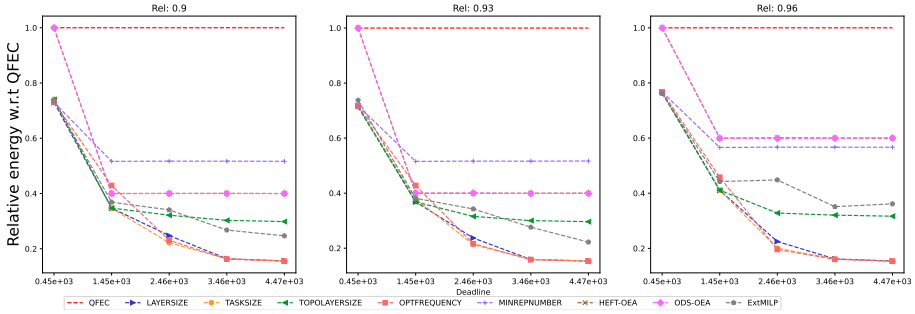

Figure 893: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

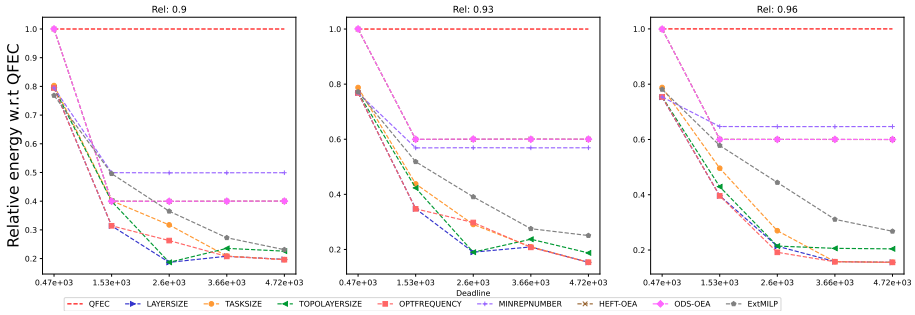

Figure 894: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

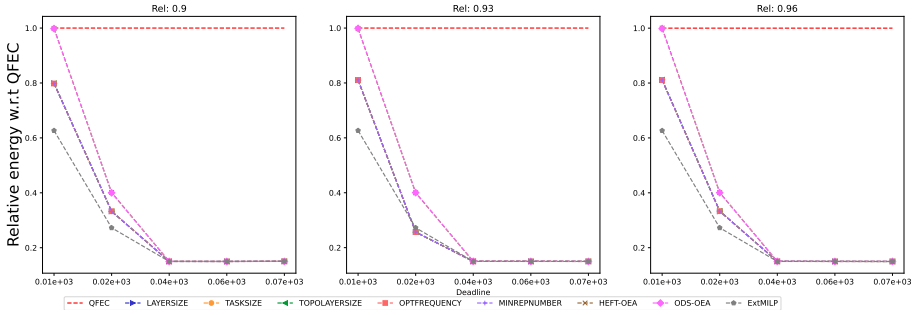

Figure 895: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

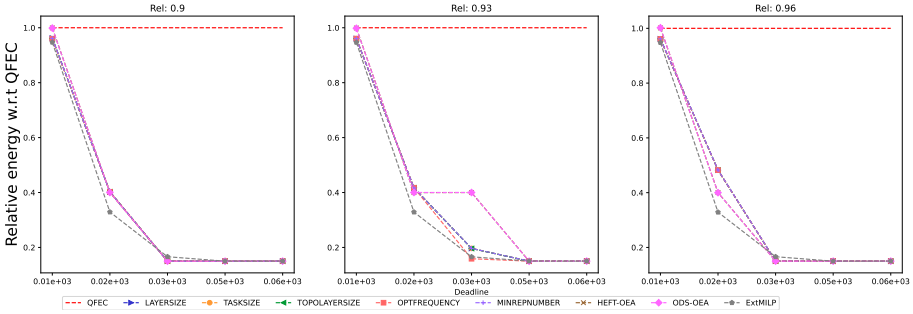

Figure 896: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

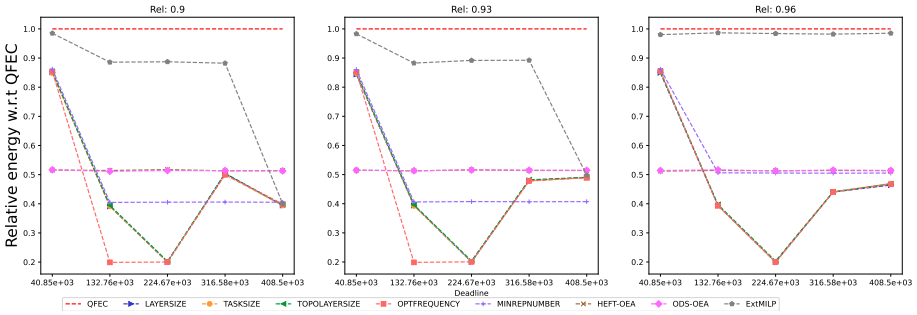

Figure 897: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

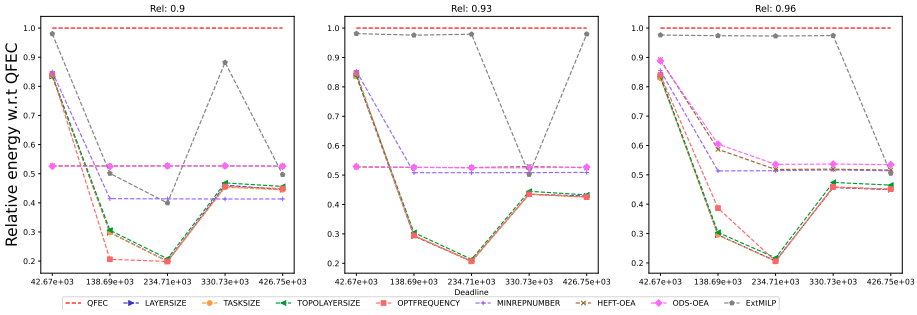

Figure 898: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

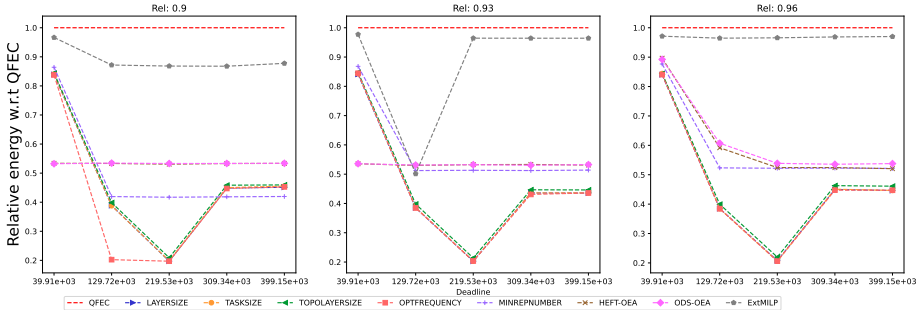

Figure 899: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

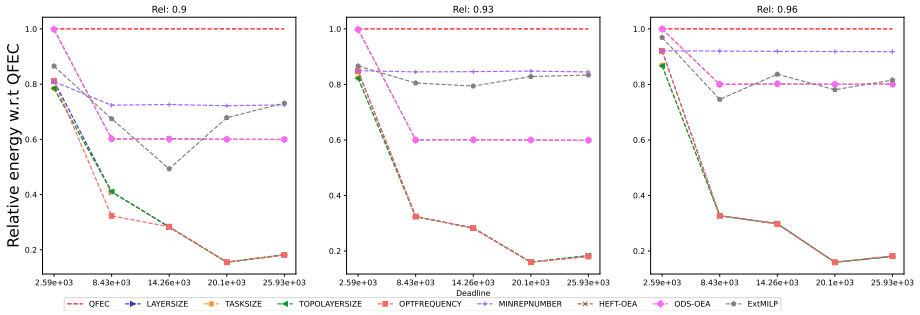

Figure 900: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

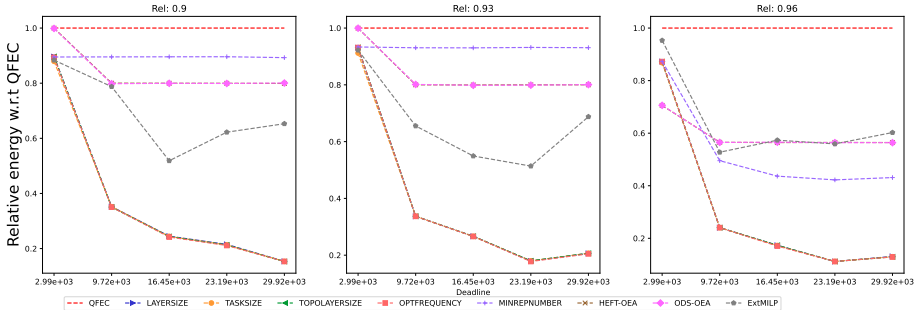

Figure 901: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.2.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

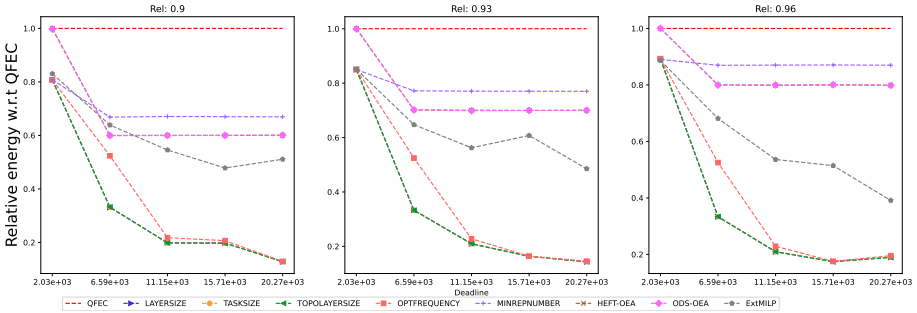

Figure 902: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

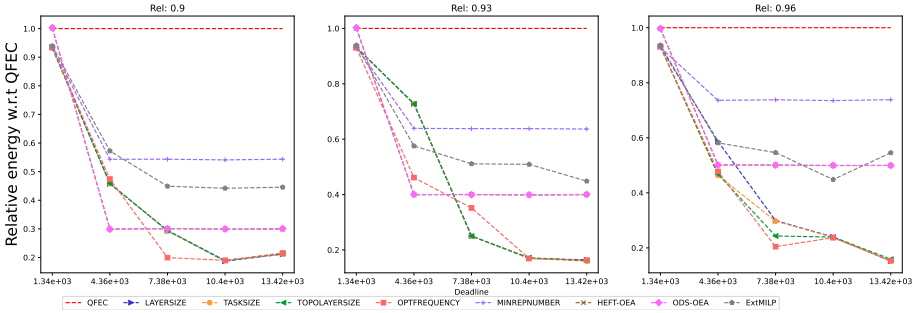

Figure 903: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

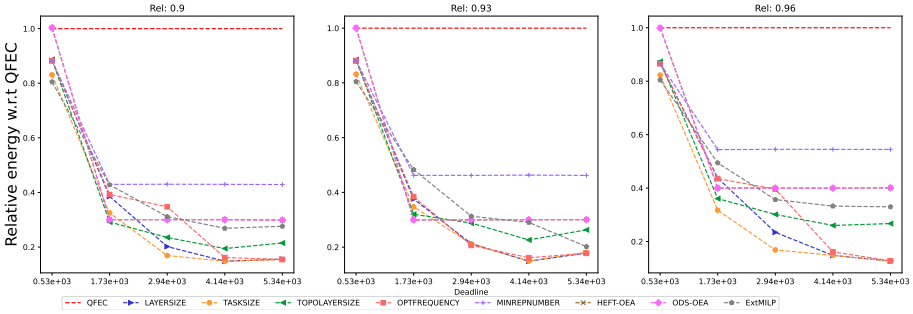

Figure 904: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

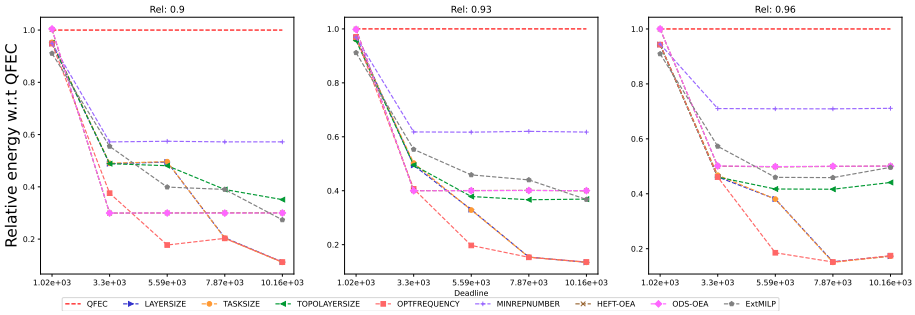

Figure 905: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

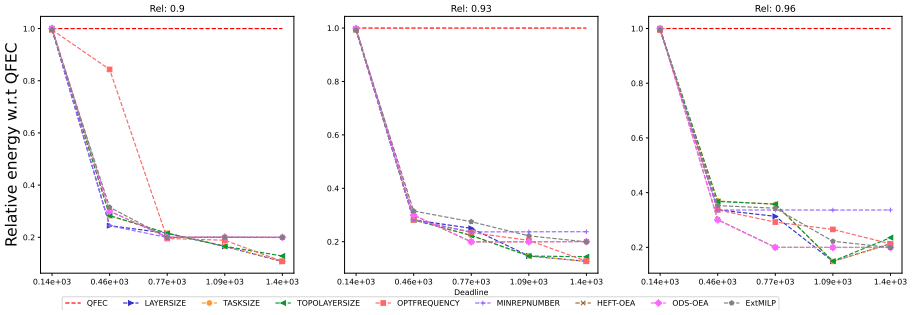

Figure 906: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

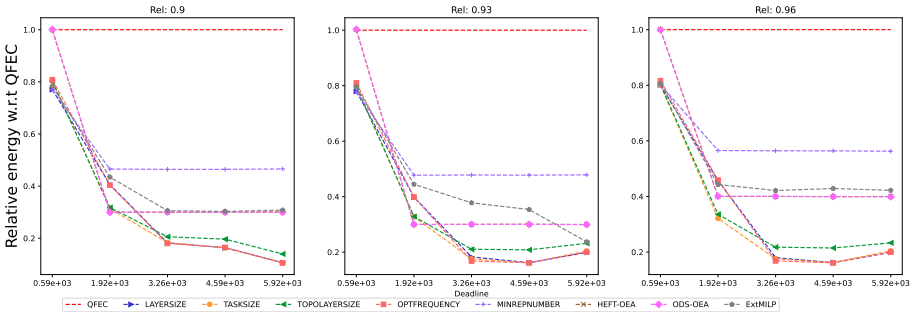

Figure 907: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

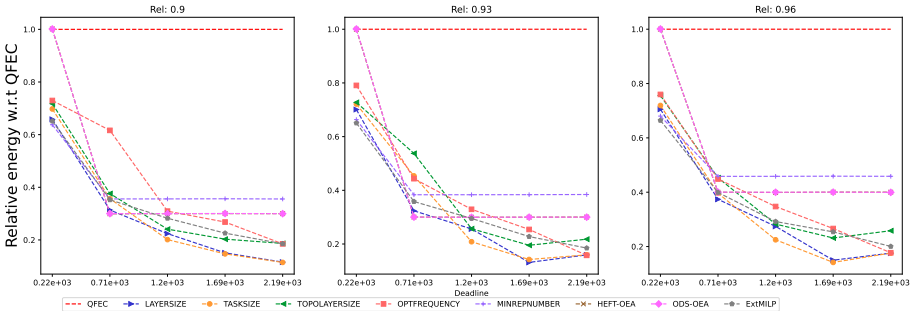

Figure 908: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

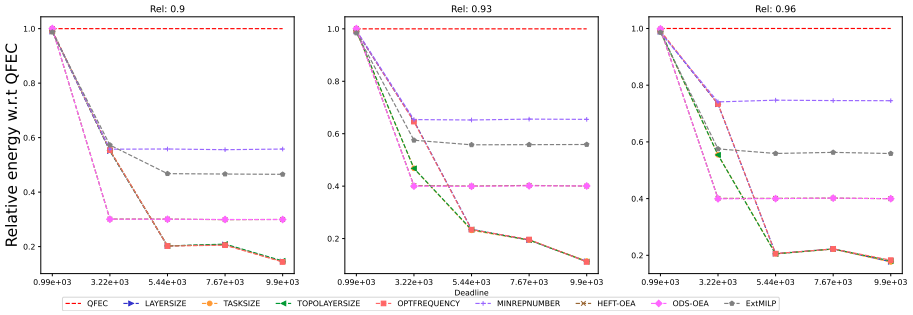

Figure 909: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

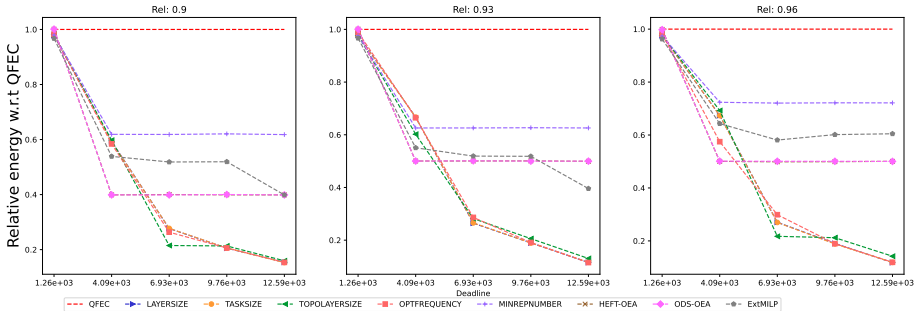

Figure 910: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

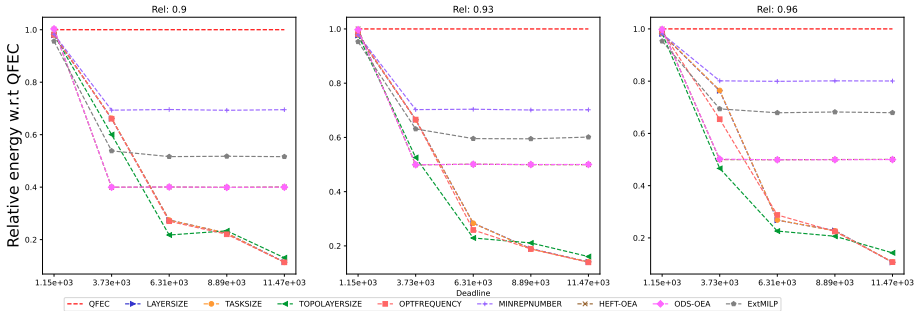

Figure 911: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

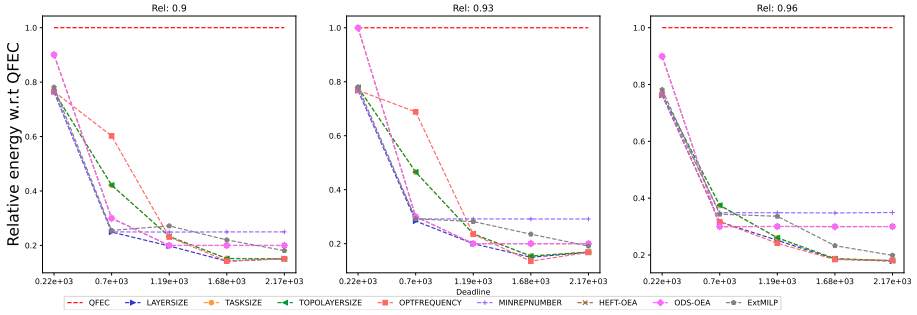

Figure 912: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

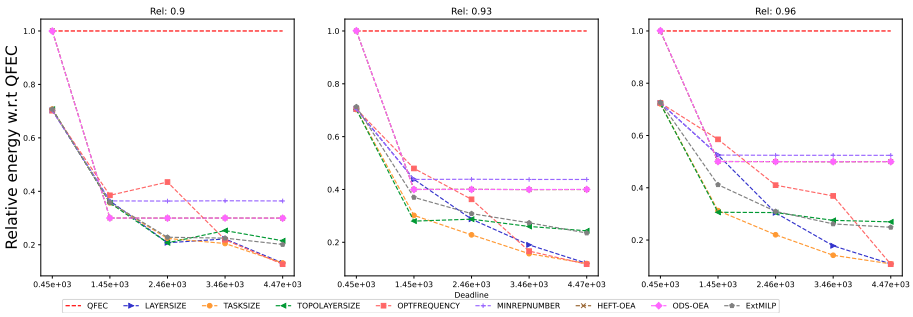

Figure 913: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

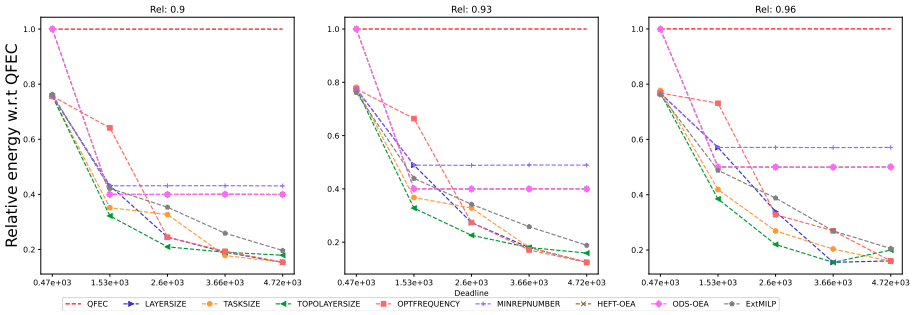

Figure 914: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

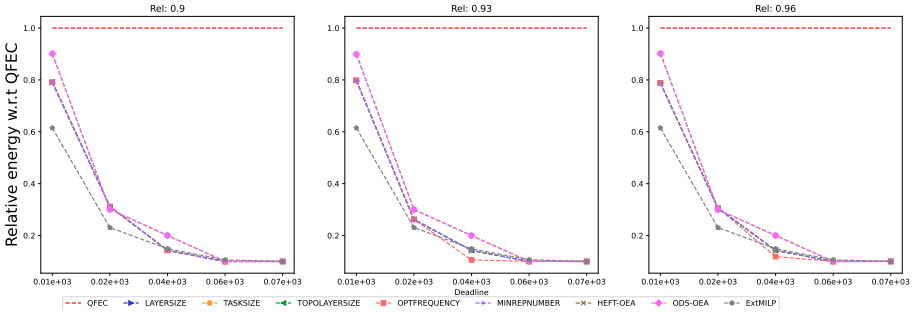

Figure 915: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

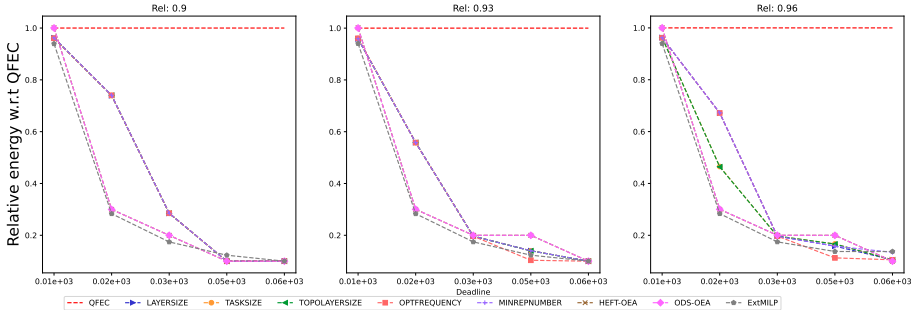

Figure 916: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

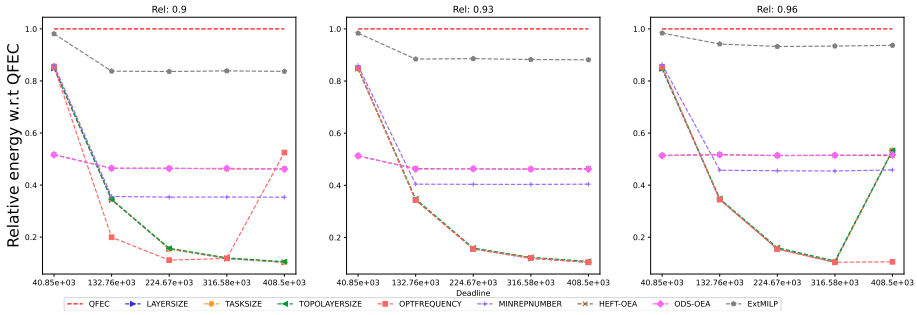

Figure 917: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

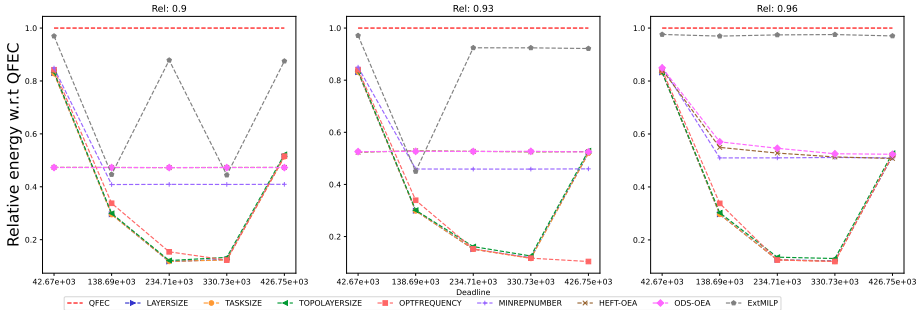

Figure 918: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

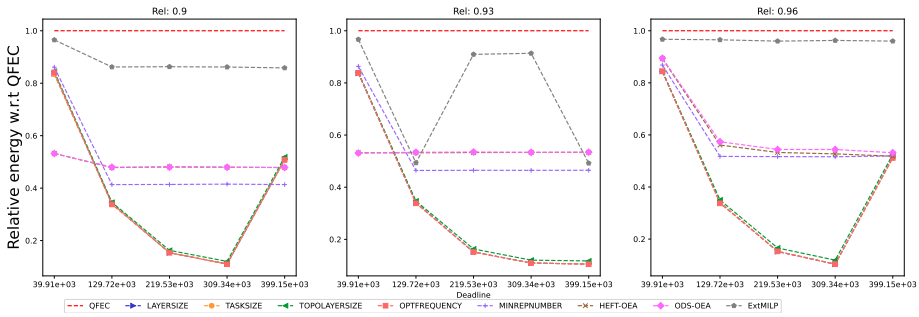

Figure 919: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

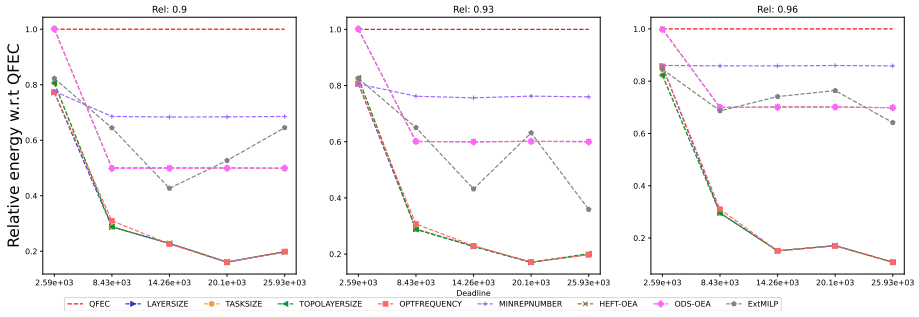

Figure 920: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

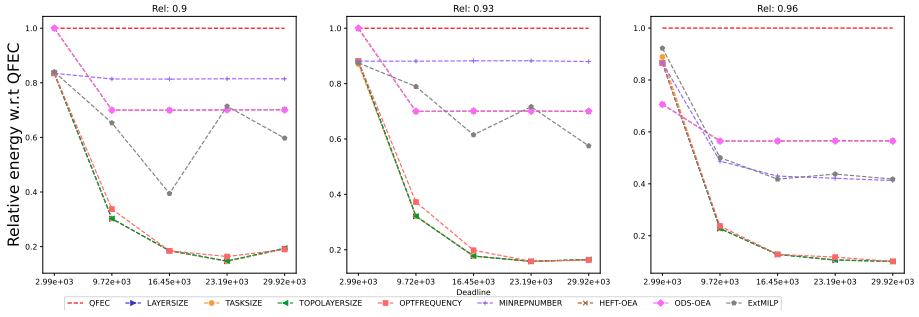

Figure 921: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.2.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

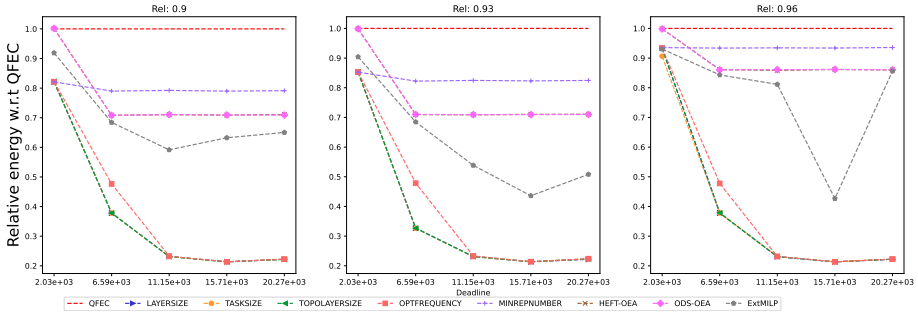

Figure 922: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

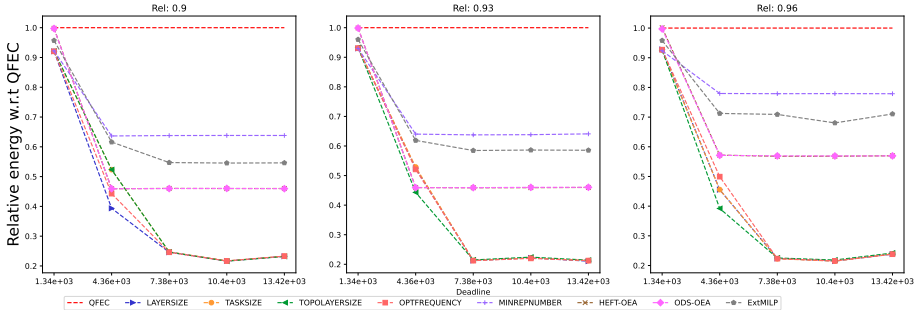

Figure 923: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

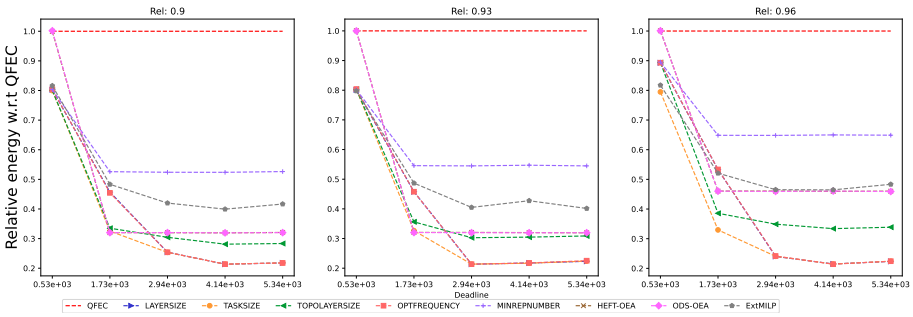

Figure 924: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

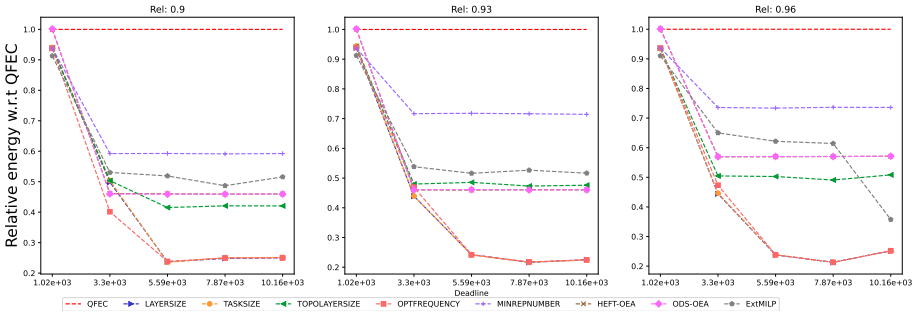

Figure 925: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

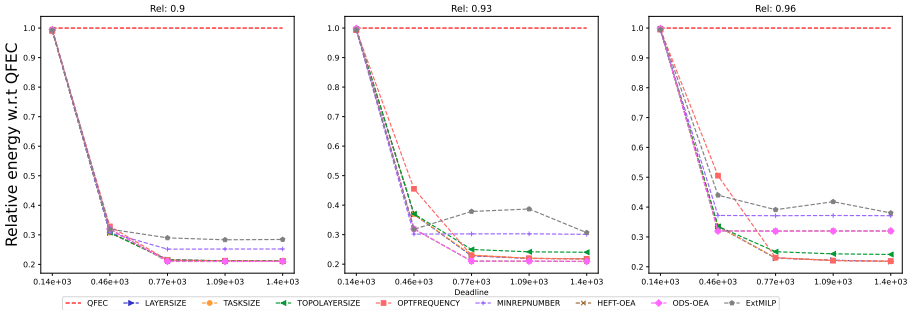

Figure 926: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

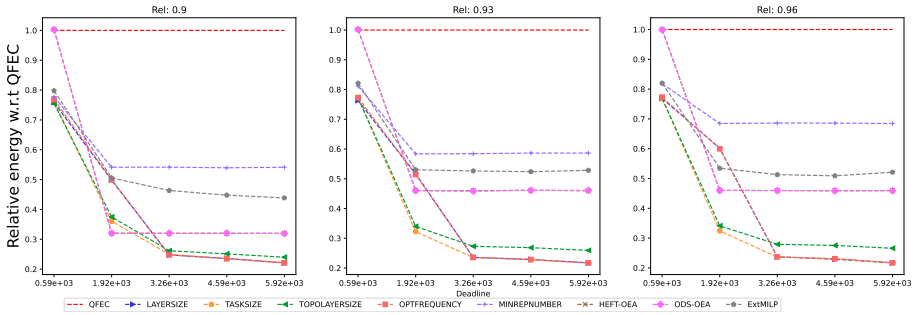

Figure 927: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

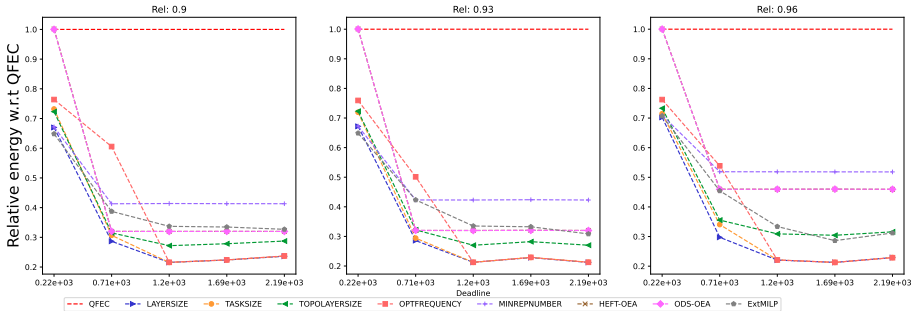

Figure 928: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

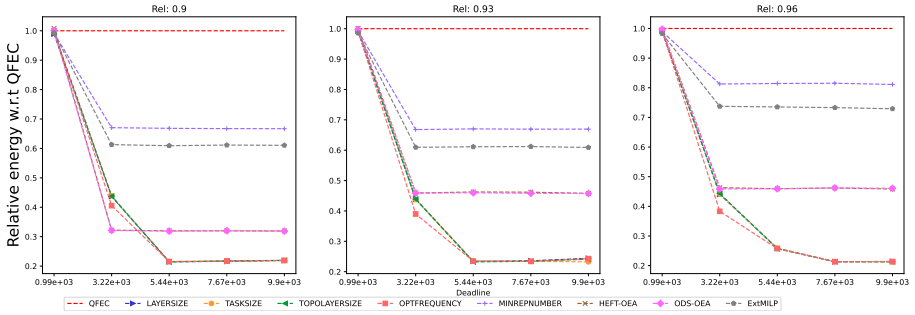

Figure 929: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

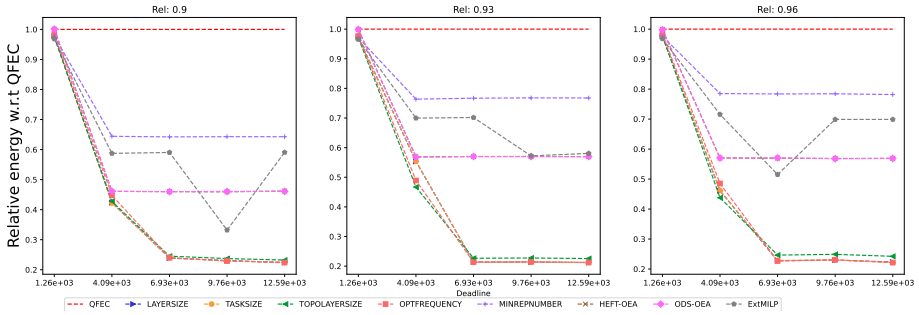

Figure 930: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

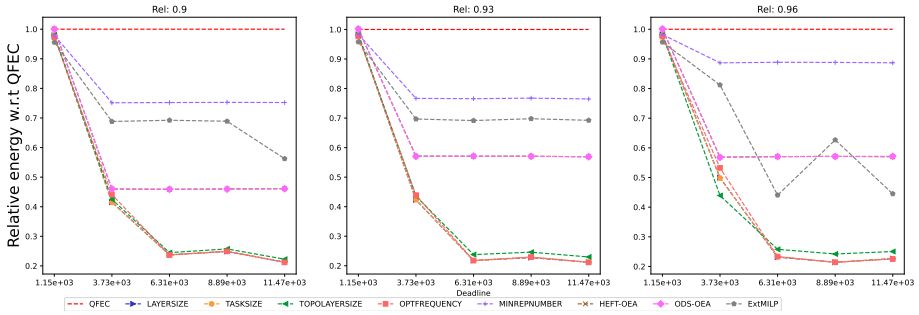

Figure 931: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

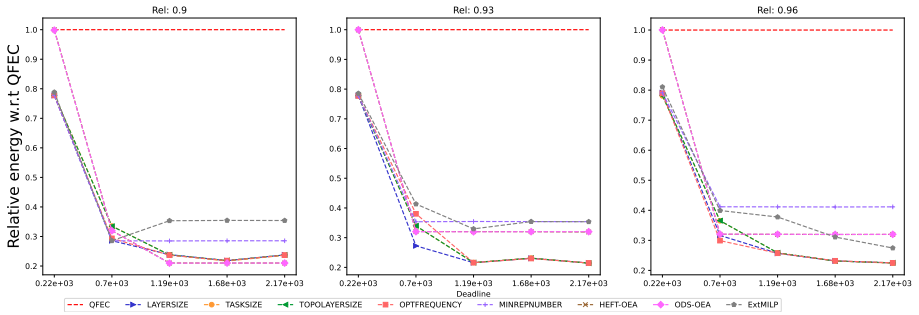

Figure 932: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

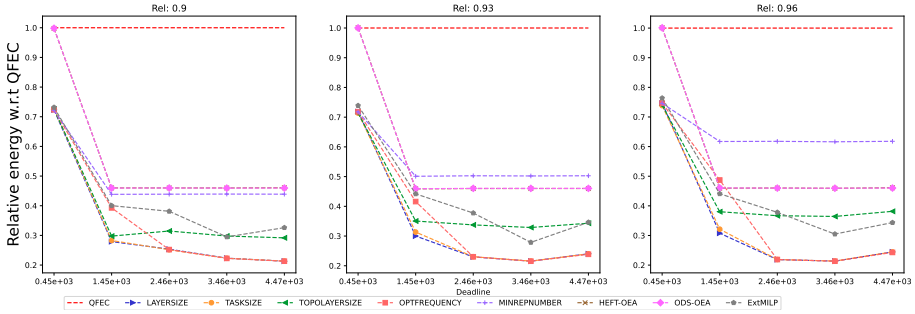

Figure 933: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

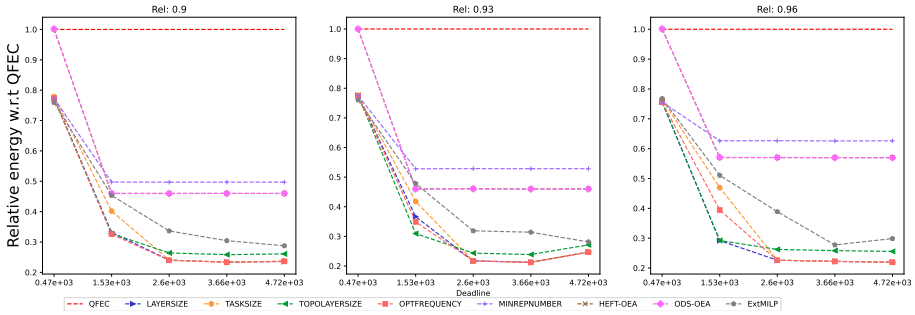

Figure 934: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

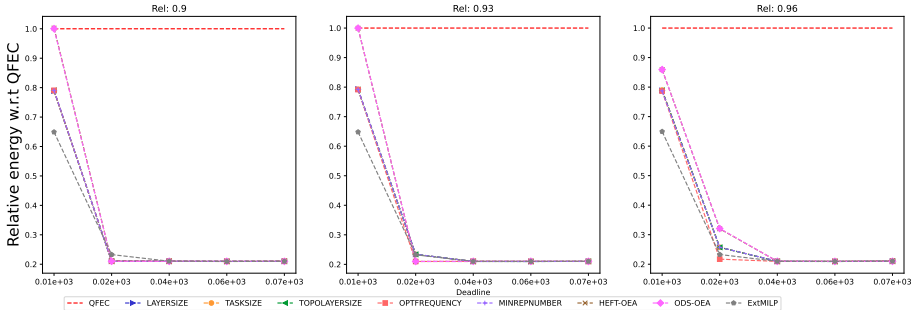

Figure 935: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

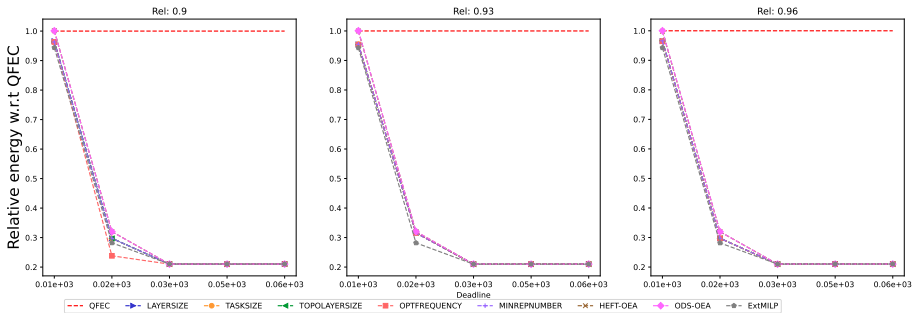

Figure 936: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

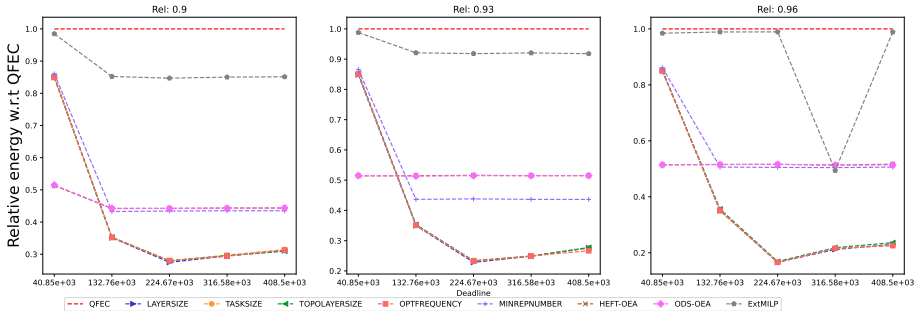

Figure 937: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

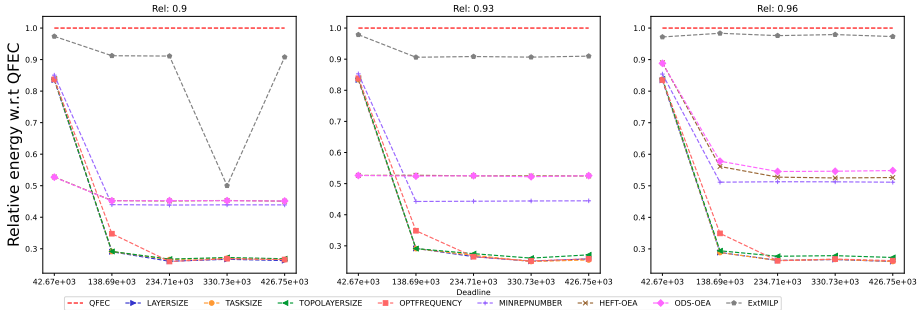

Figure 938: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

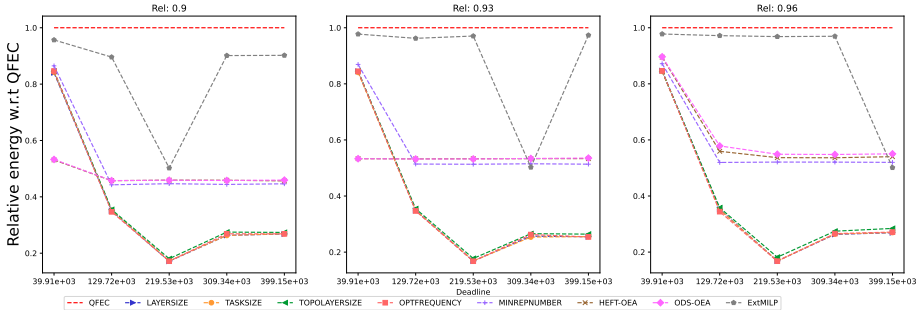

Figure 939: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

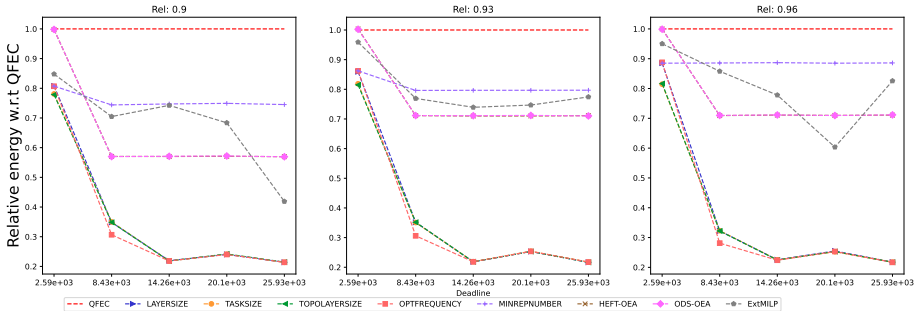

Figure 940: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

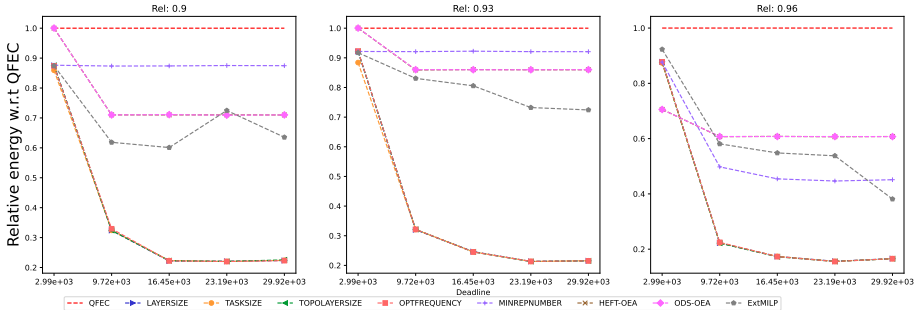

Figure 941: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.2.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

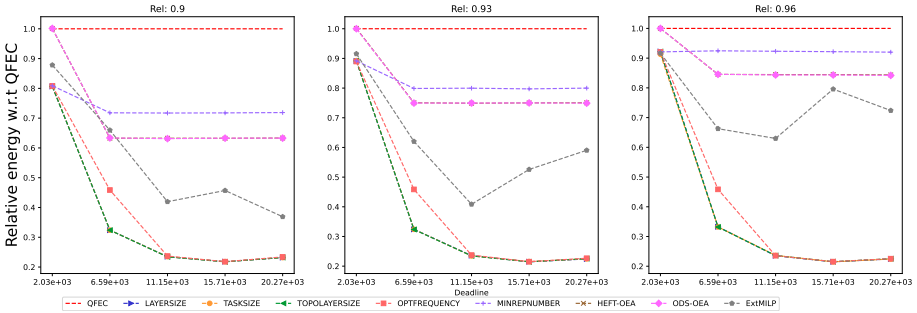

Figure 942: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

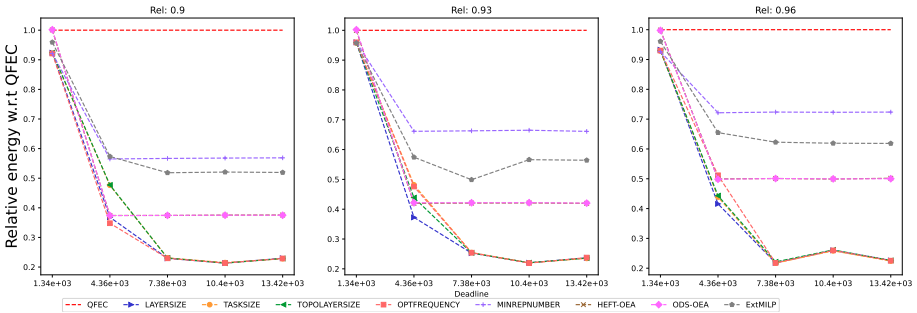

Figure 943: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

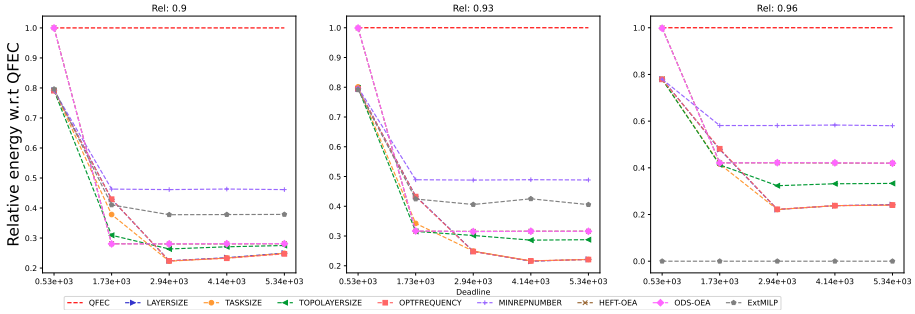

Figure 944: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

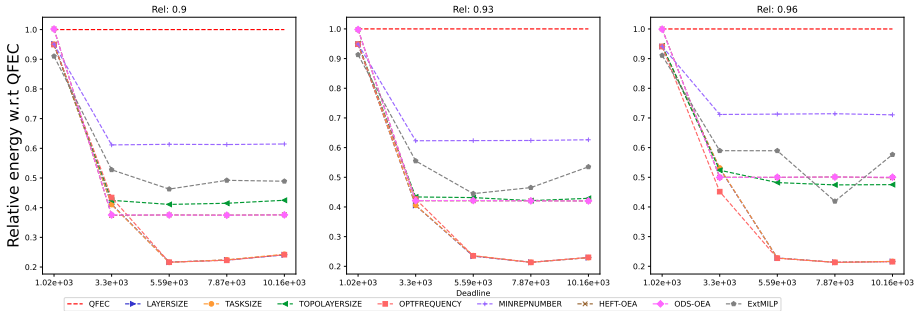

Figure 945: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

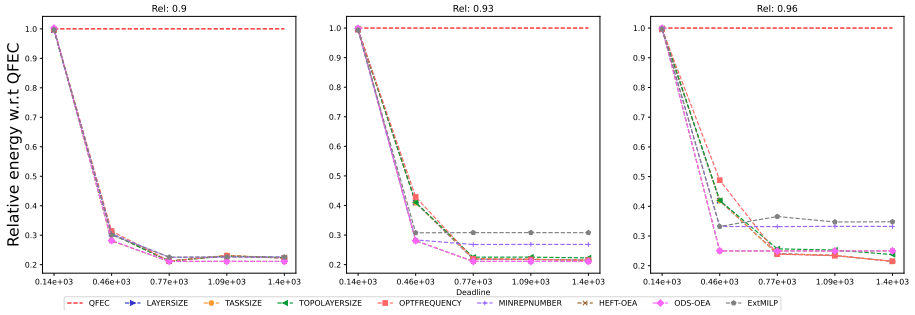

Figure 946: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

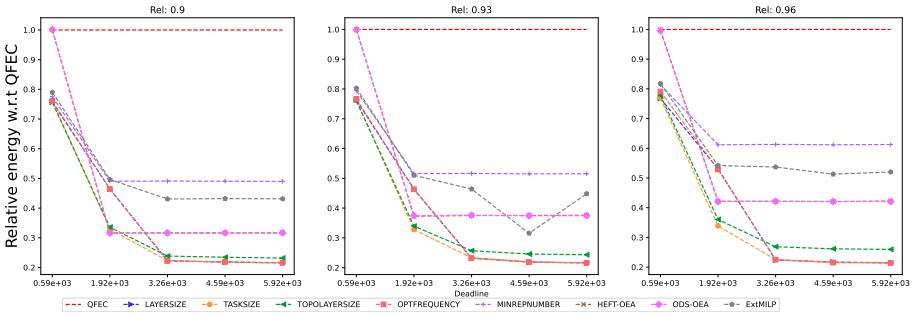

Figure 947: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

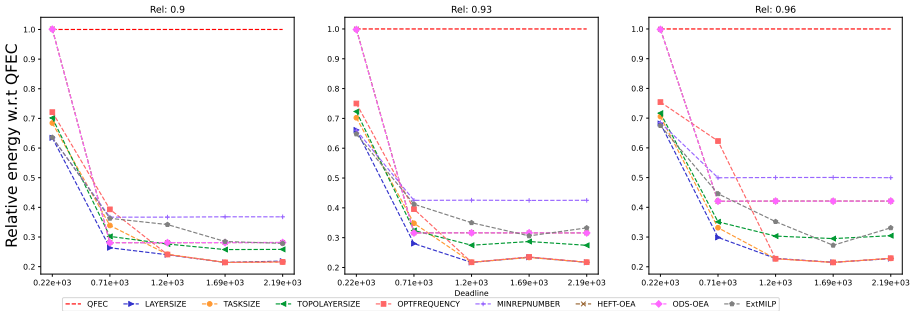

Figure 948: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

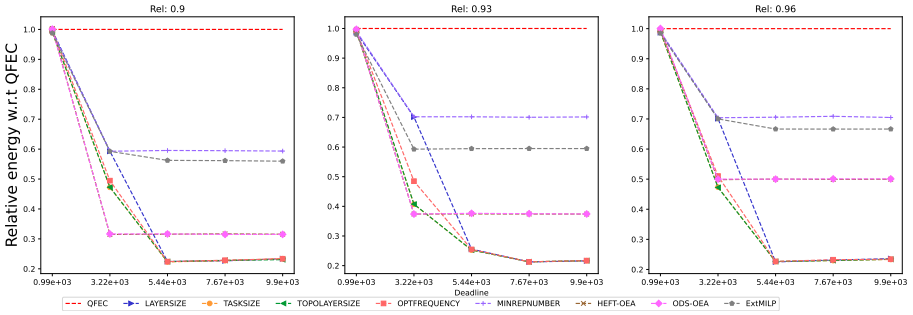

Figure 949: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

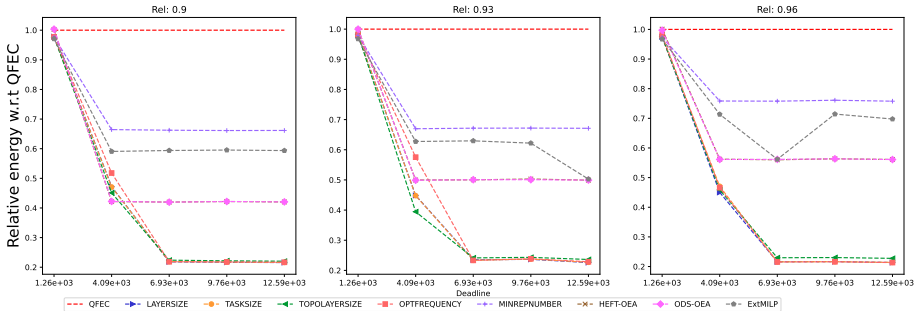

Figure 950: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

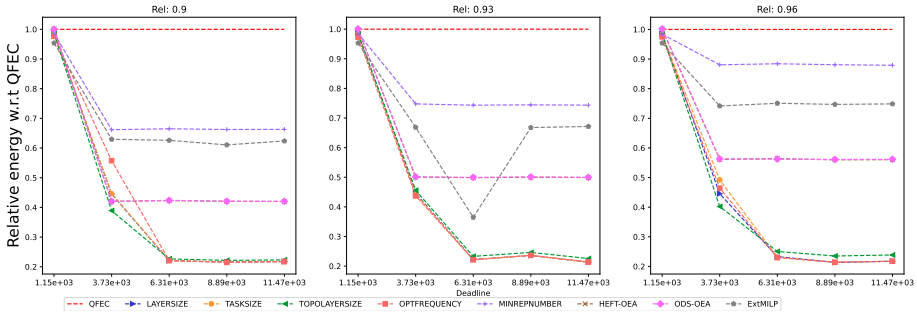

Figure 951: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

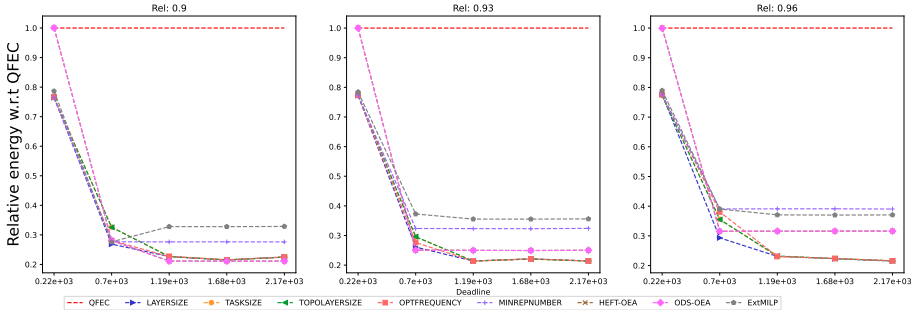

Figure 952: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

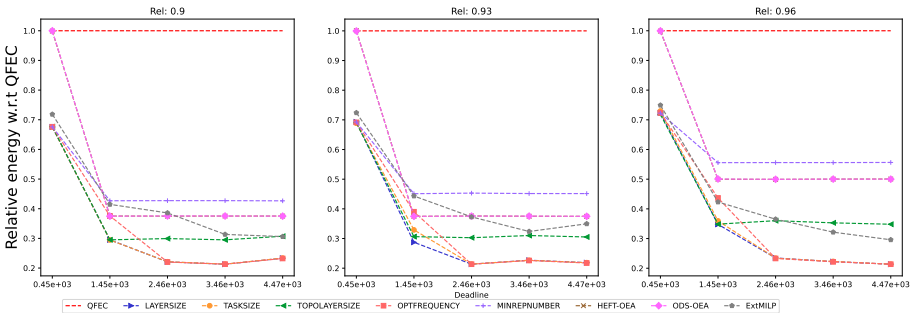

Figure 953: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

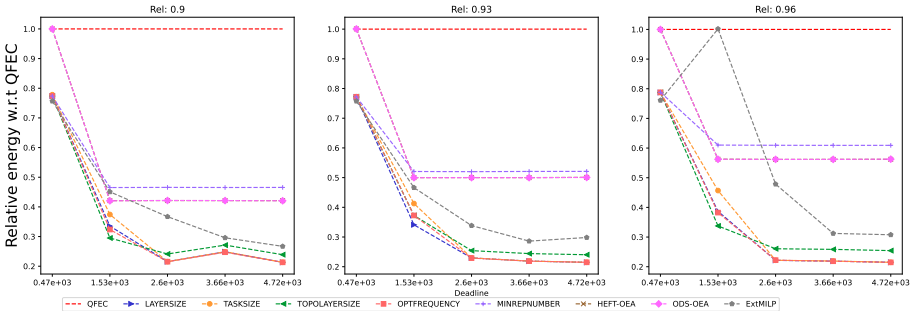

Figure 954: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

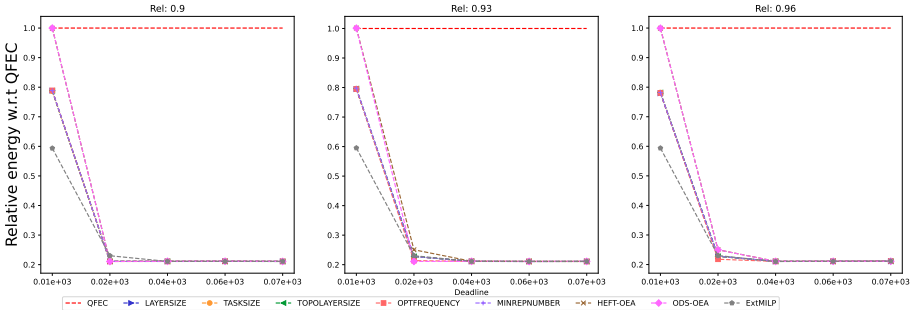

Figure 955: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

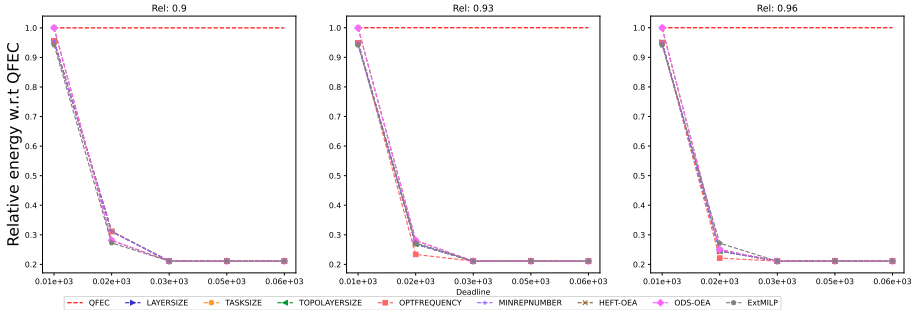

Figure 956: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

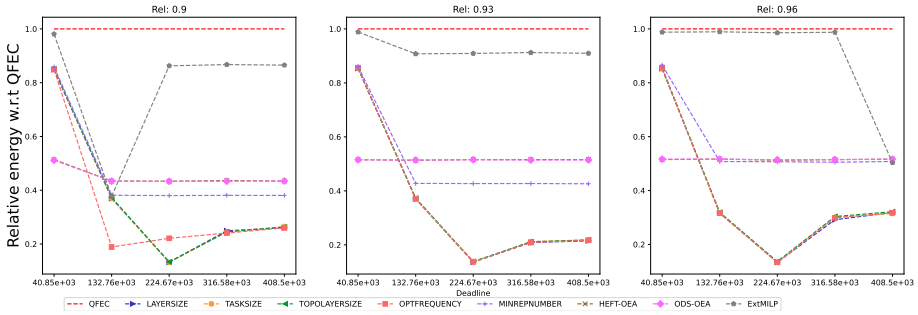

Figure 957: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

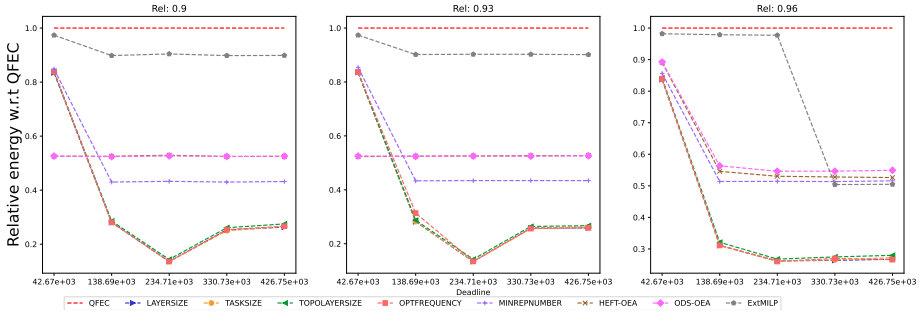

Figure 958: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

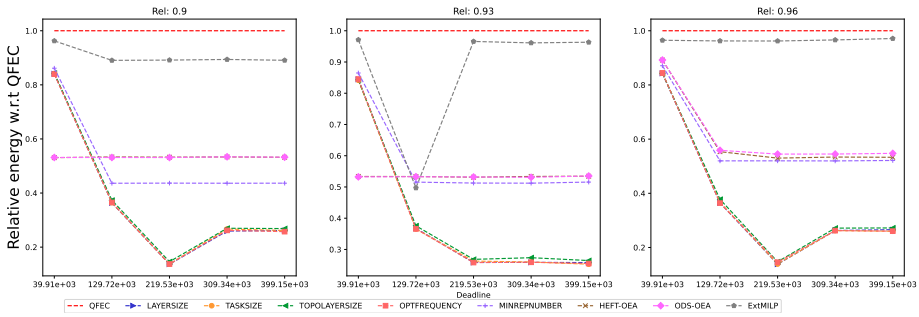

Figure 959: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

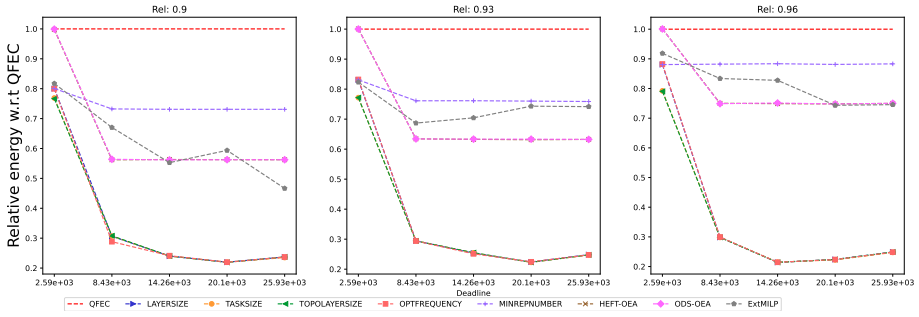

Figure 960: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

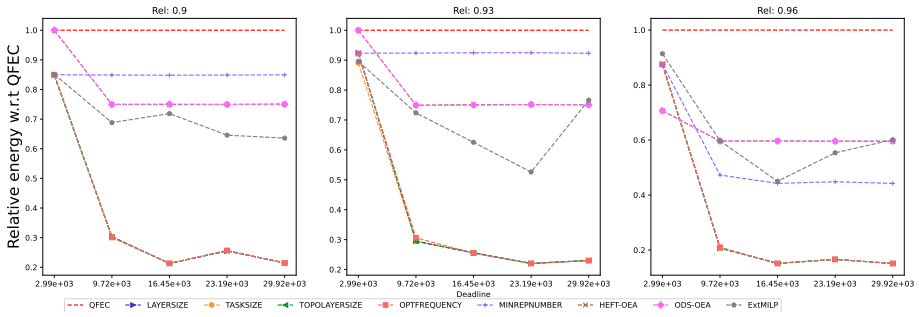

Figure 961: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.3 $BC/WC = 0.3$

F.3.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

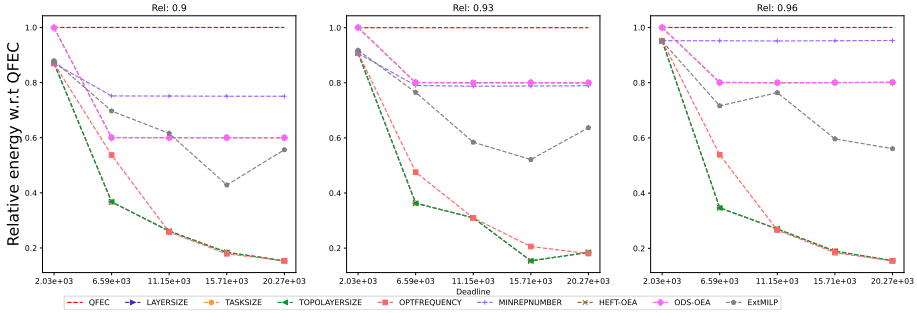

Figure 962: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

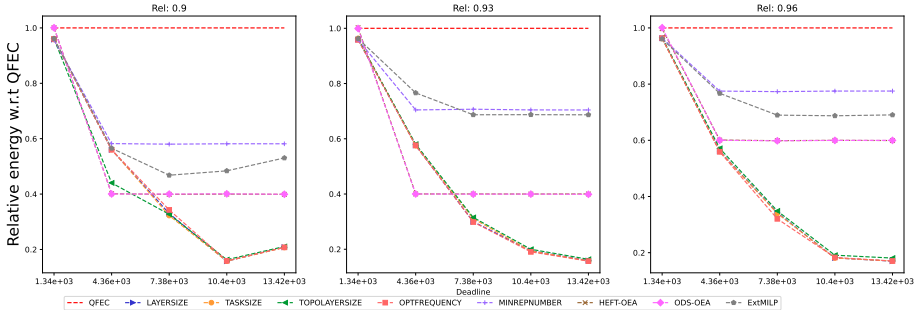

Figure 963: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

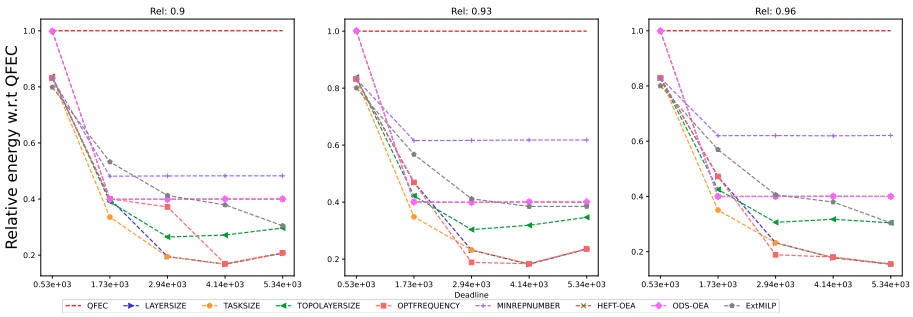

Figure 964: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

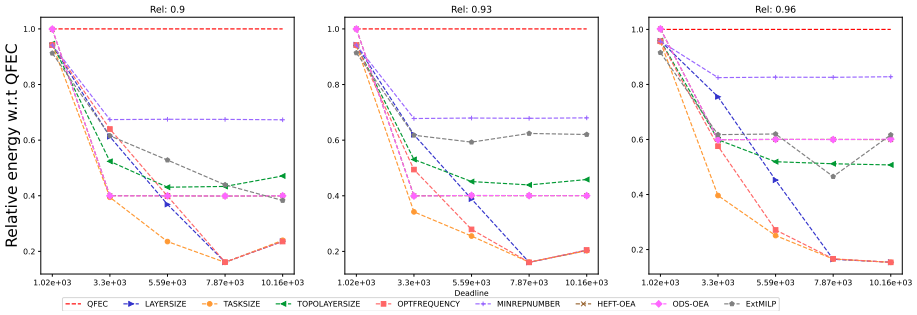

Figure 965: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

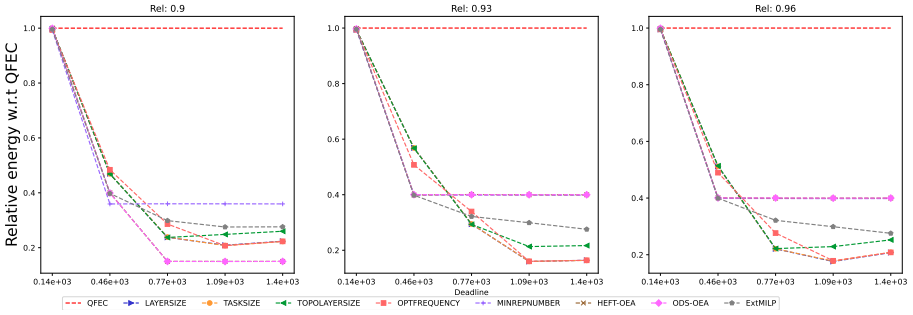

Figure 966: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

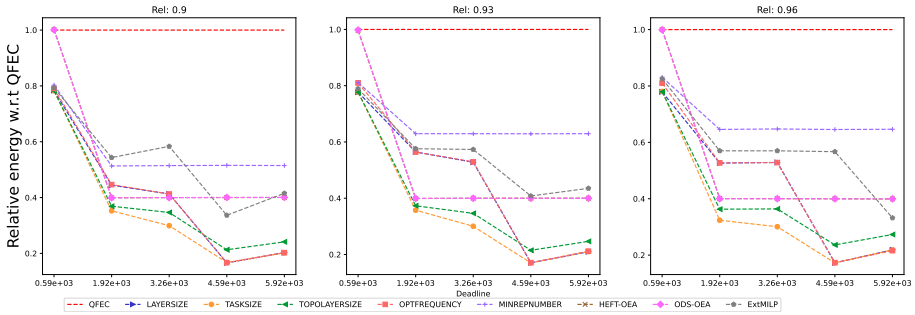

Figure 967: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

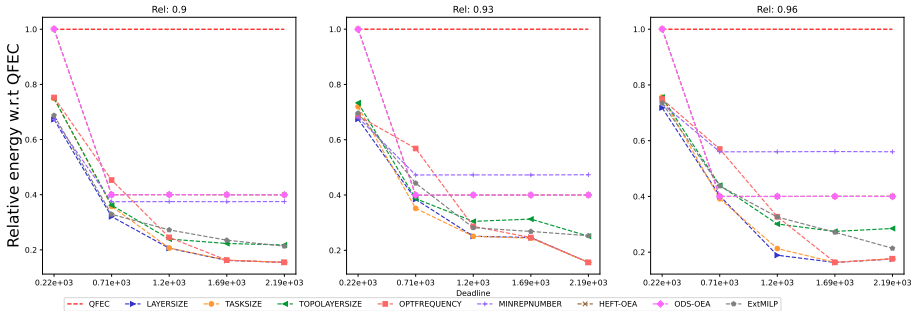

Figure 968: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

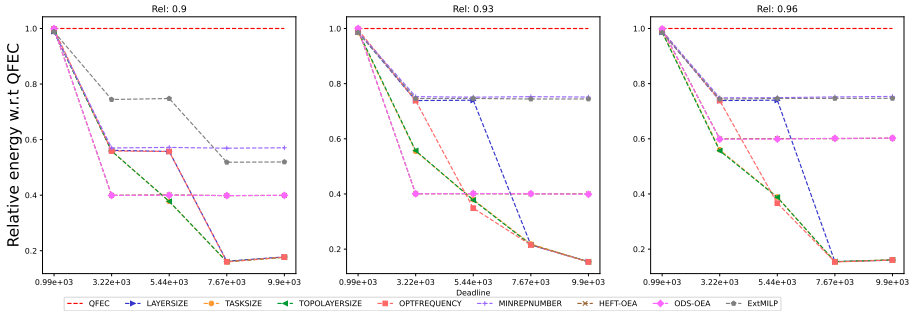

Figure 969: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

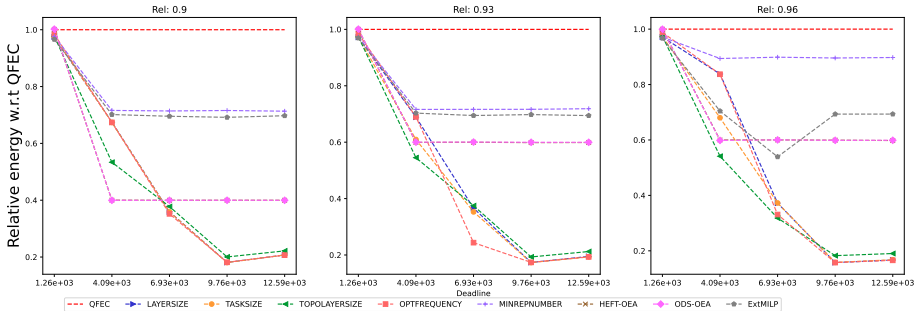

Figure 970: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

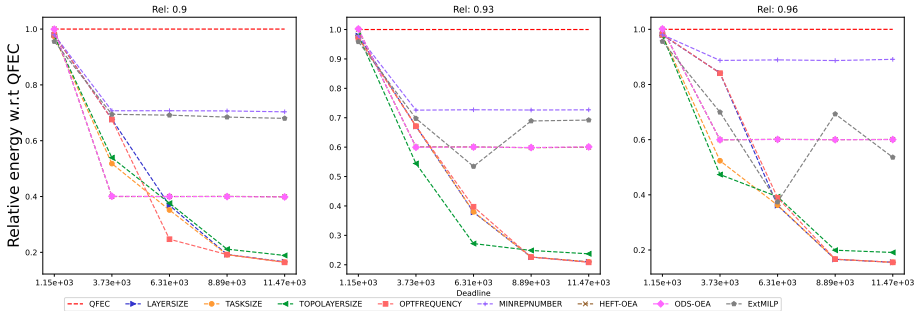

Figure 971: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

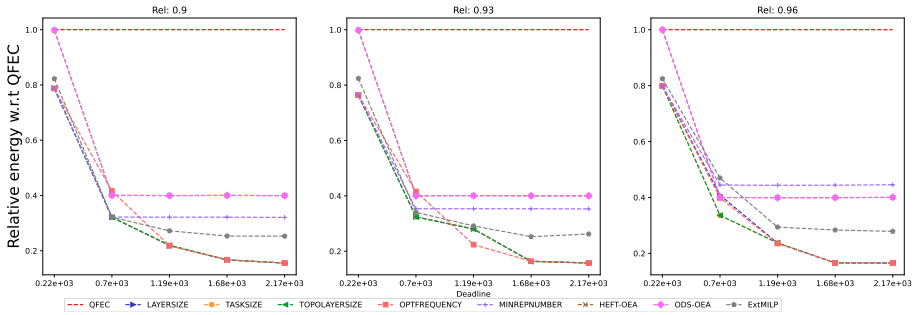

Figure 972: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

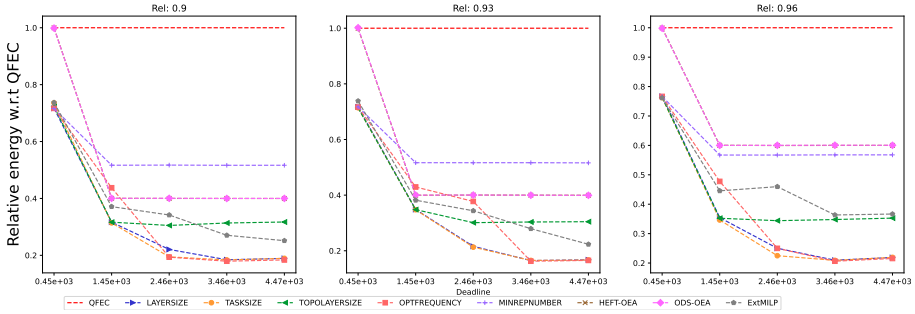

Figure 973: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

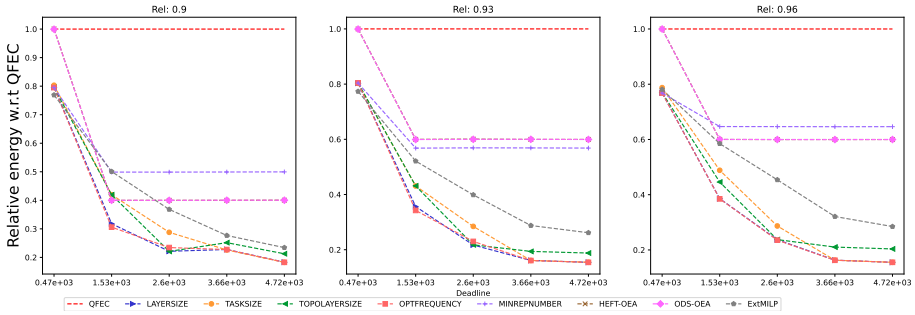

Figure 974: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

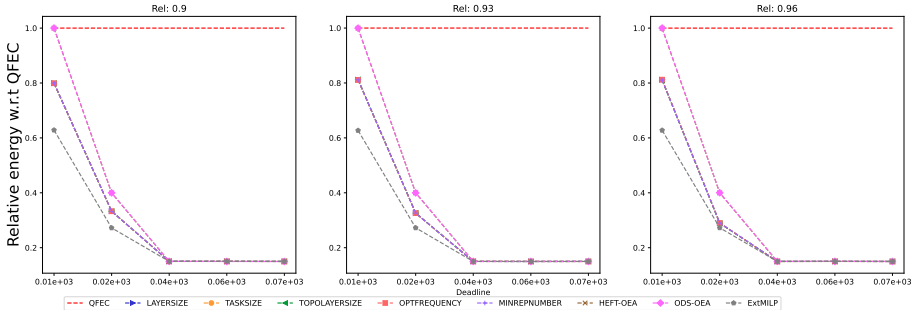

Figure 975: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

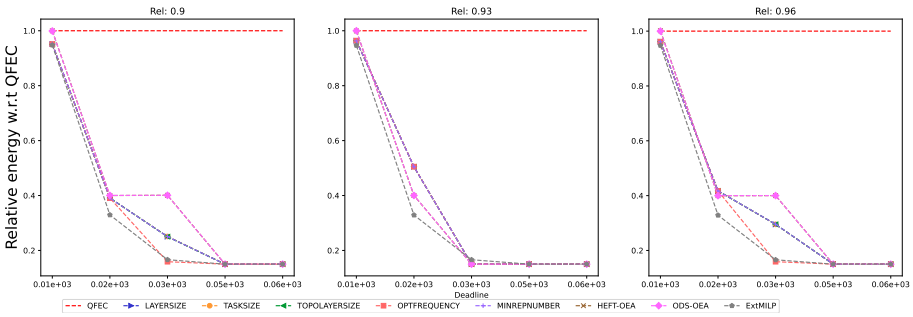

Figure 976: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

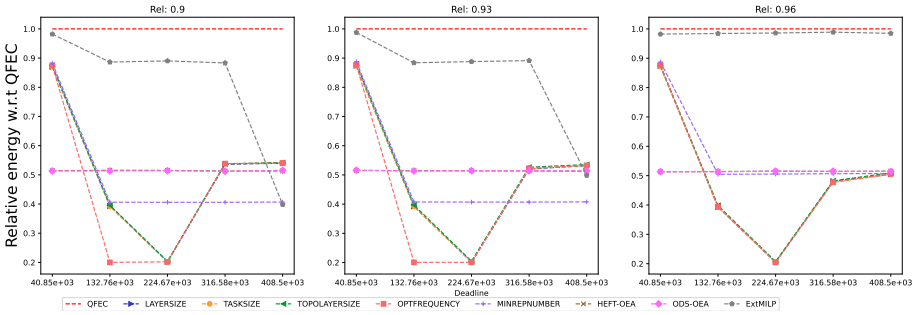

Figure 977: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

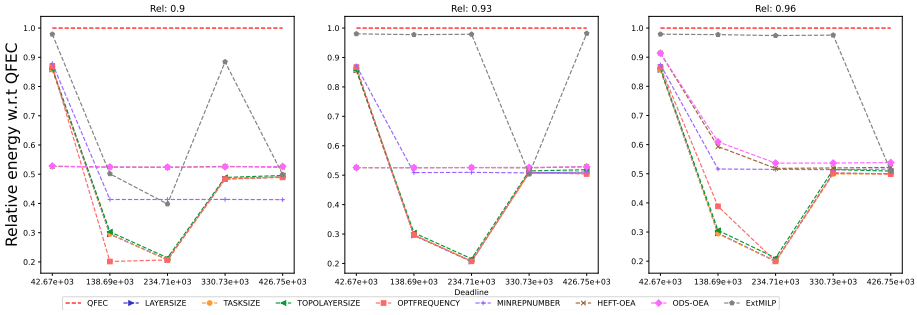

Figure 978: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

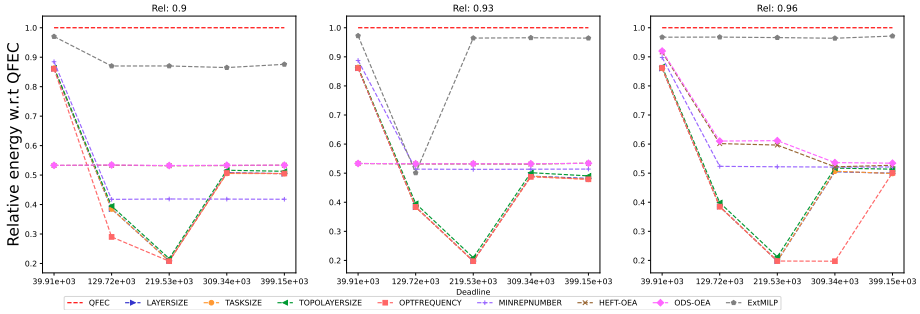

Figure 979: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

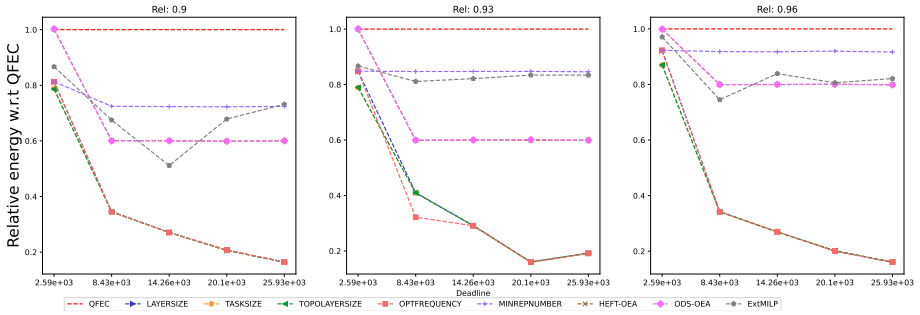

Figure 980: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

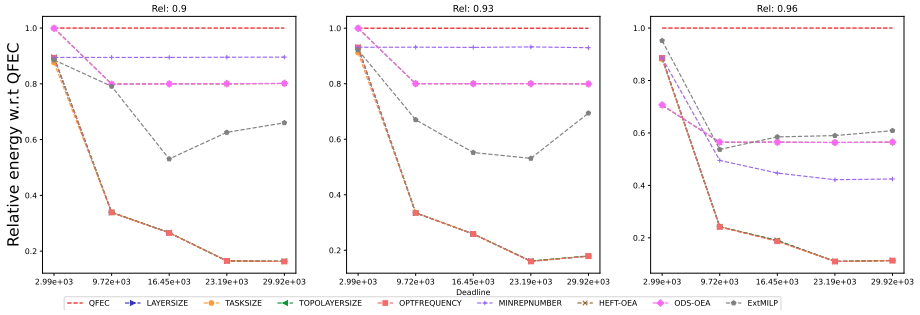

Figure 981: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.3.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

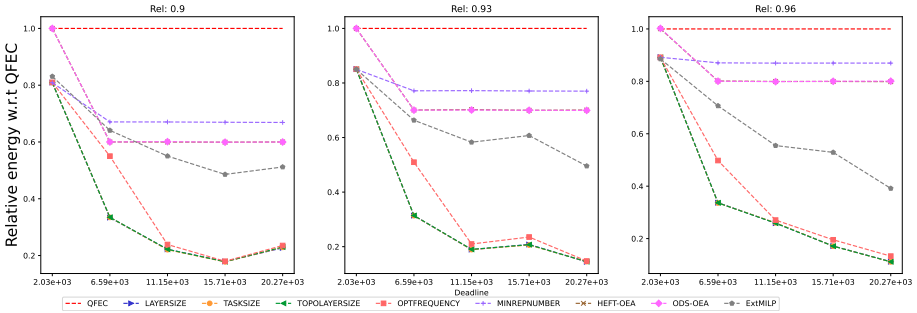

Figure 982: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

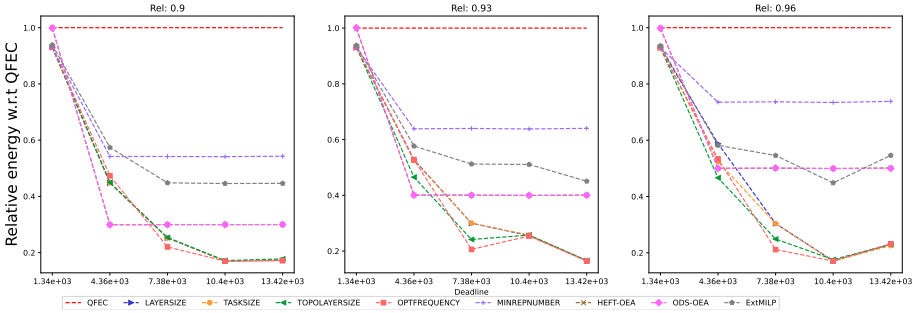

Figure 983: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

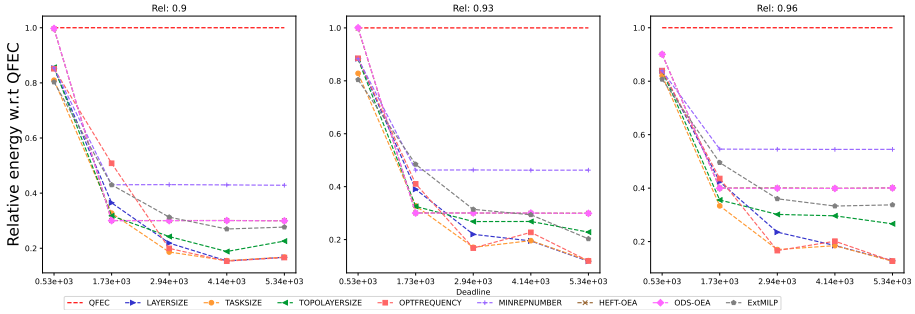

Figure 984: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

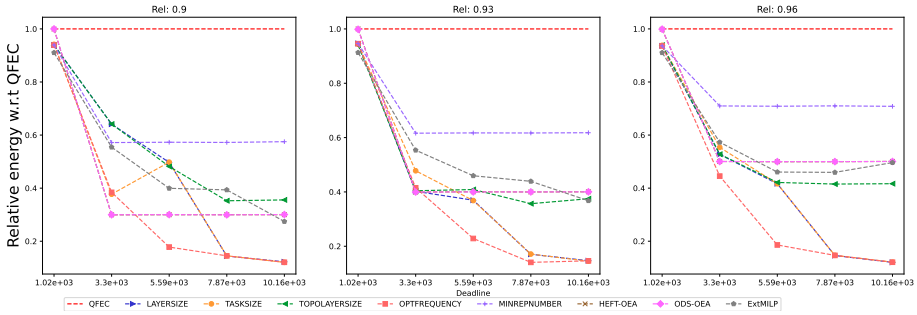

Figure 985: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

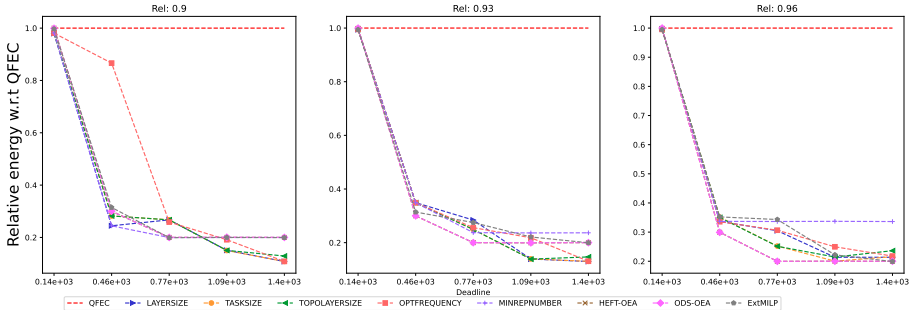

Figure 986: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

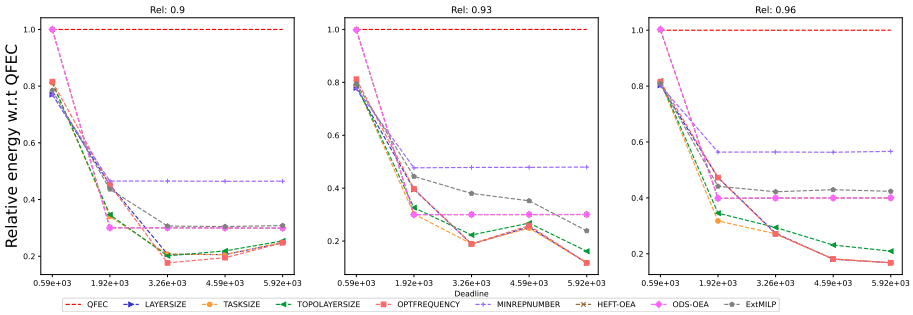

Figure 987: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

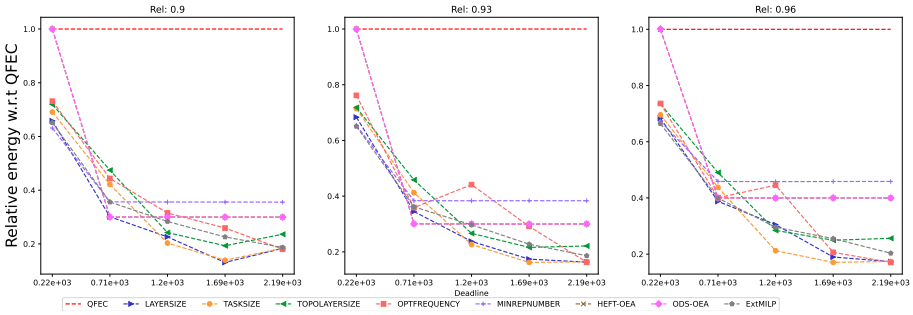

Figure 988: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

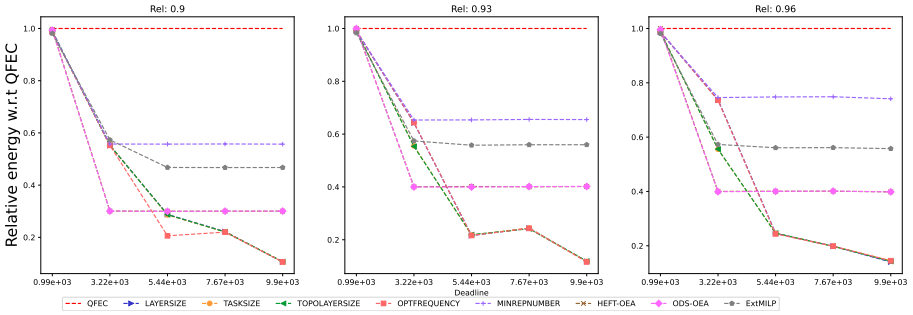

Figure 989: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

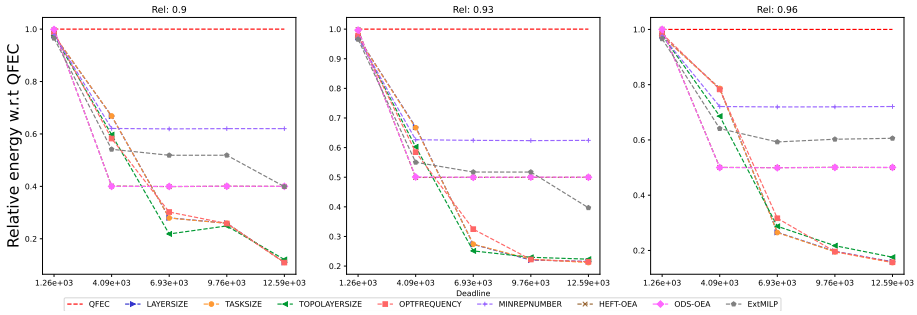

Figure 990: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

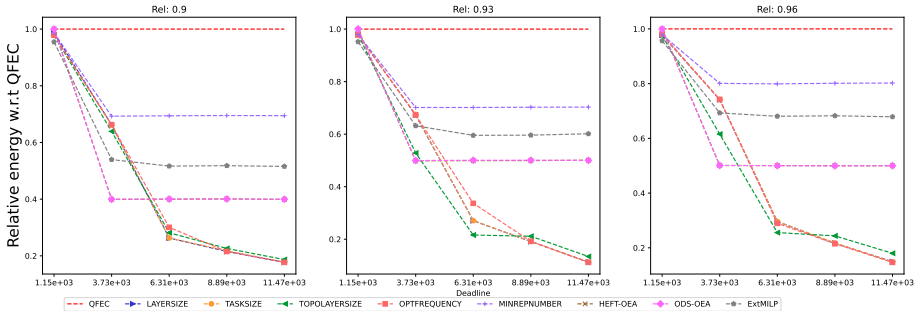

Figure 991: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

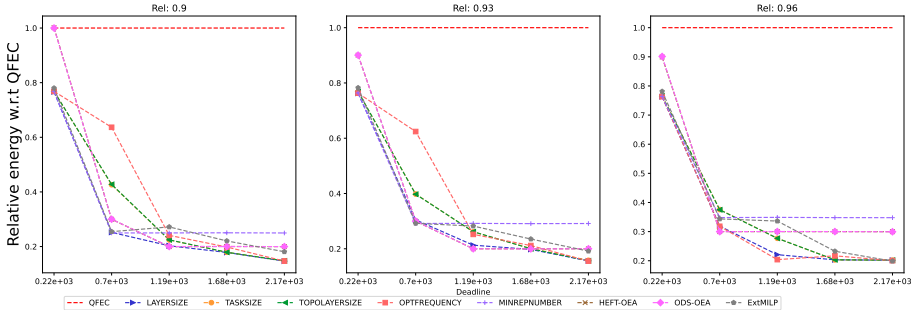

Figure 992: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

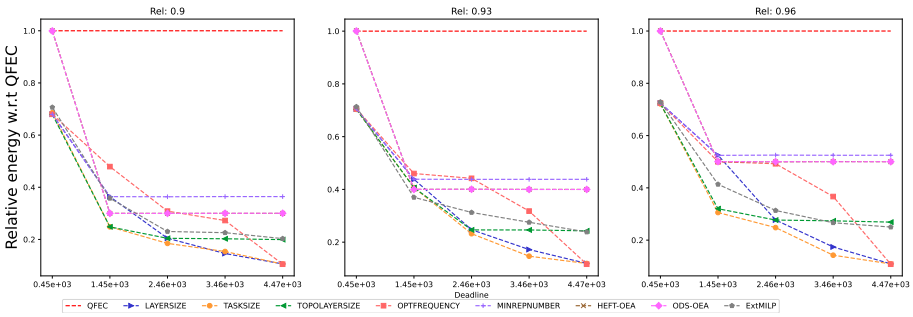

Figure 993: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

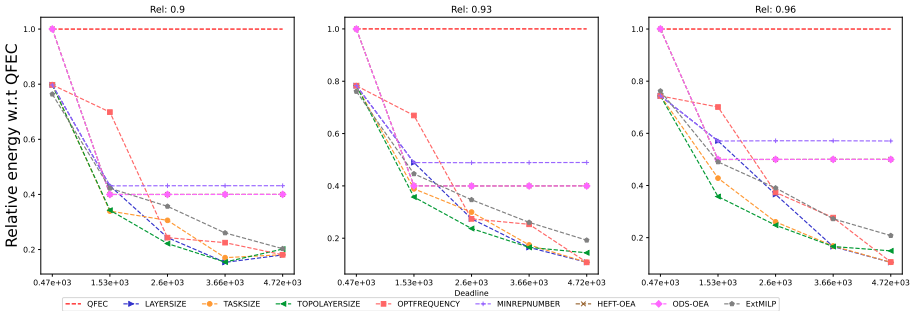

Figure 994: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

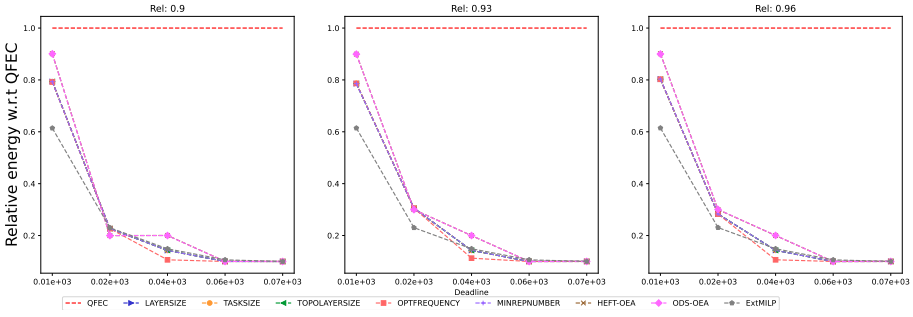

Figure 995: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

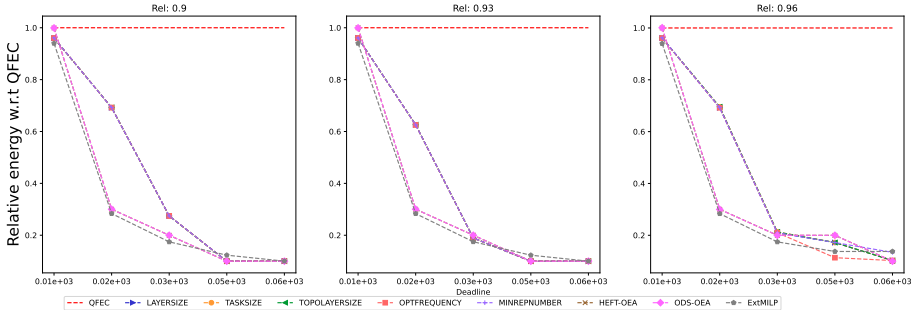

Figure 996: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

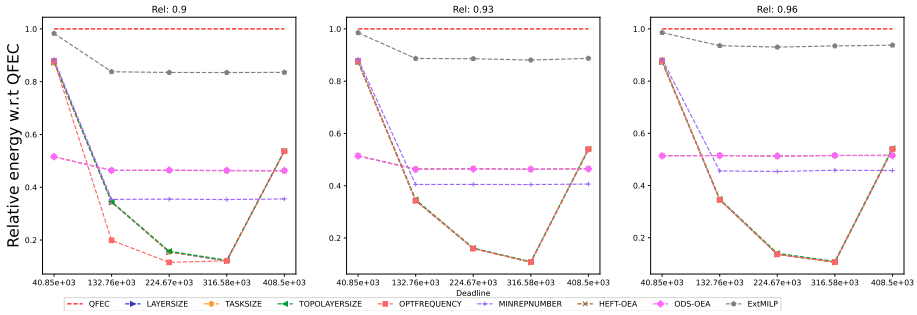

Figure 997: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

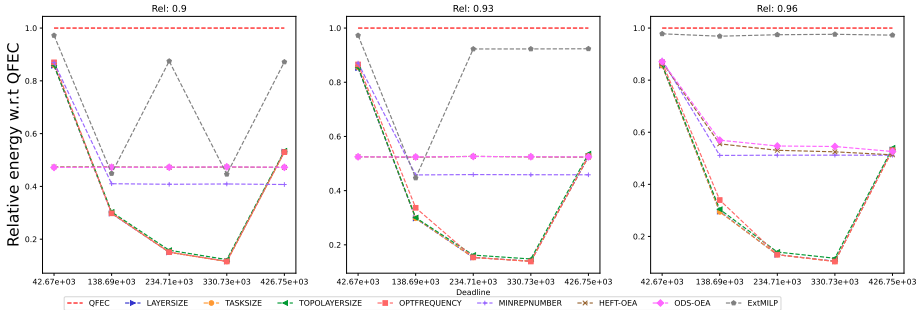

Figure 998: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

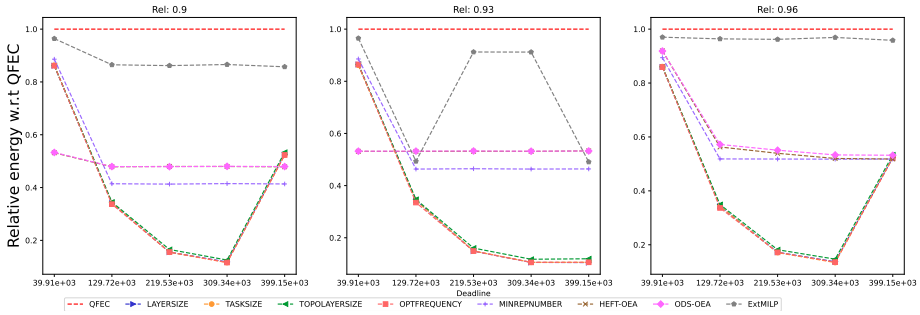

Figure 999: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

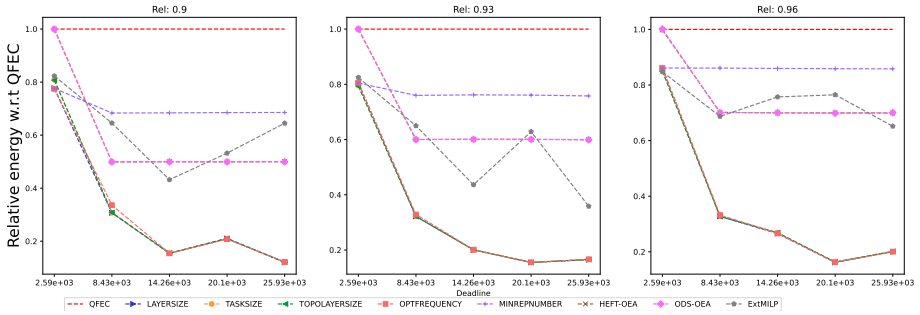

Figure 1000: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

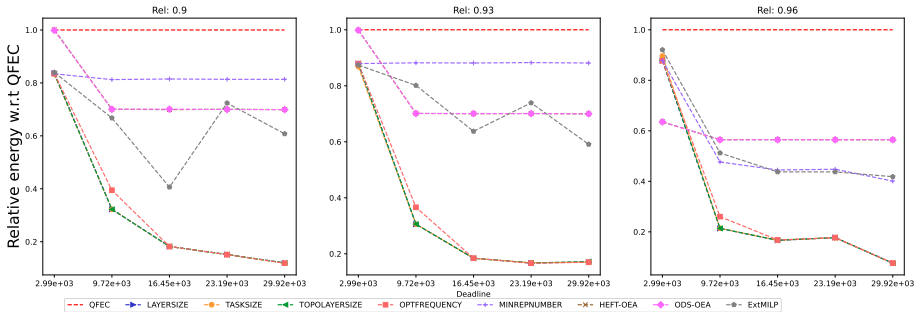

Figure 1001: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.3.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

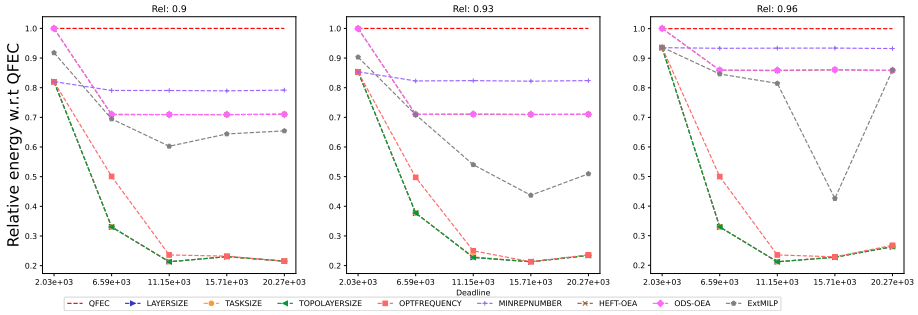

Figure 1002: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

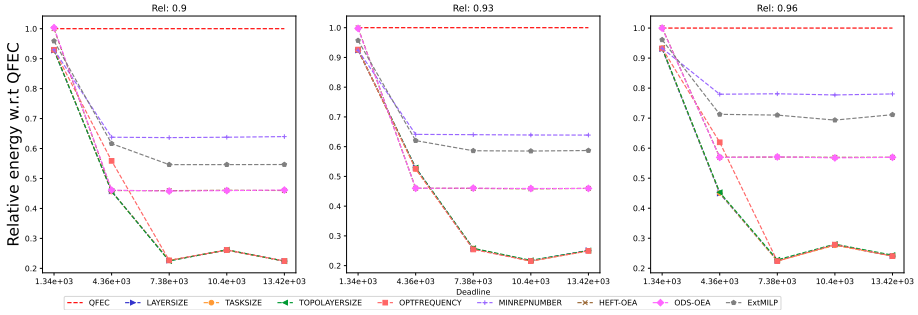

Figure 1003: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

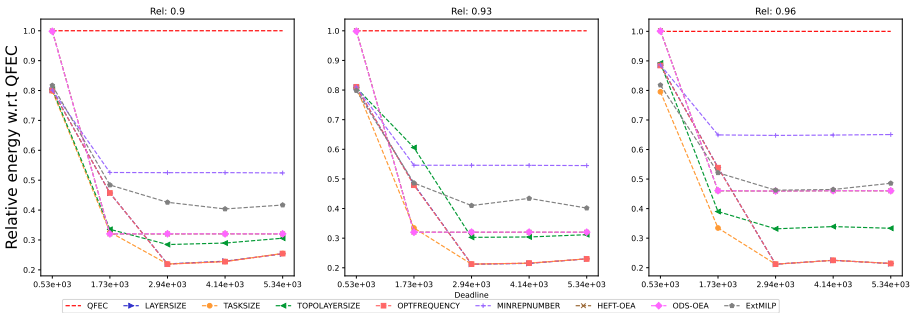

Figure 1004: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

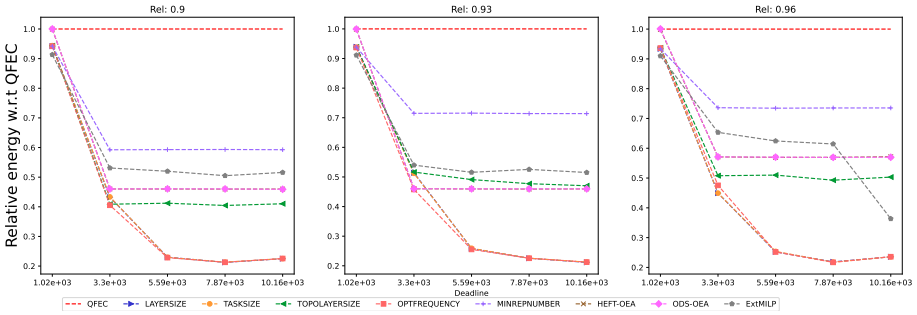

Figure 1005: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

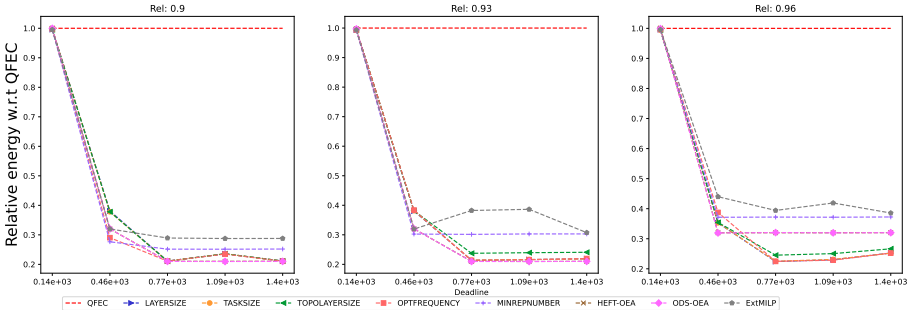

Figure 1006: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

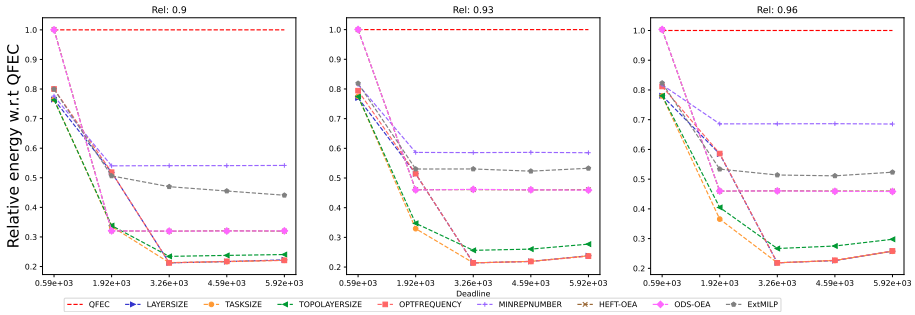

Figure 1007: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

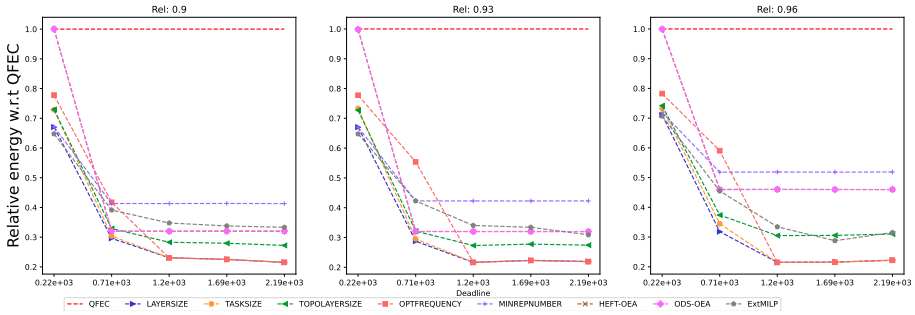

Figure 1008: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

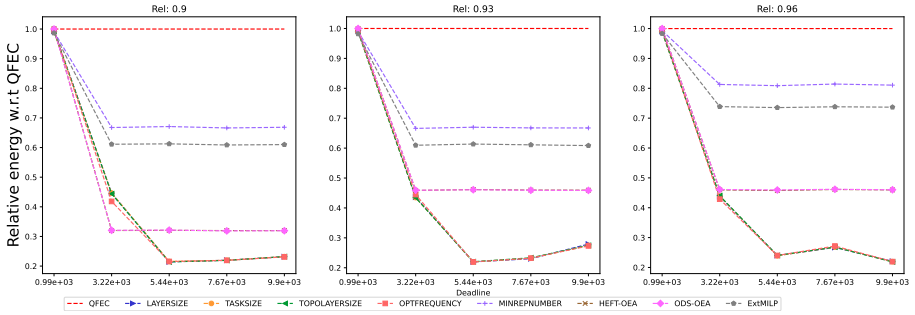

Figure 1009: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

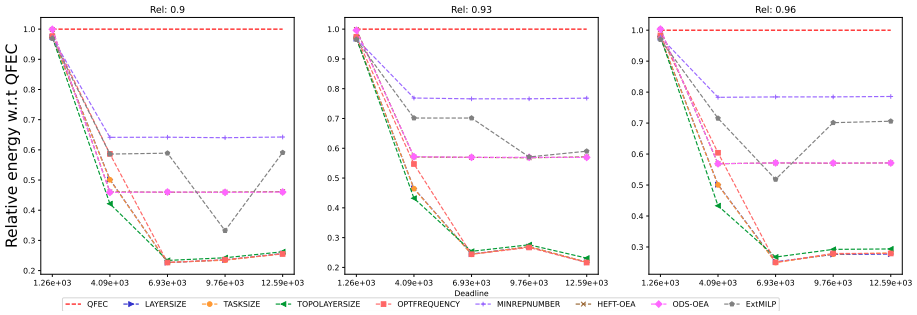

Figure 1010: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

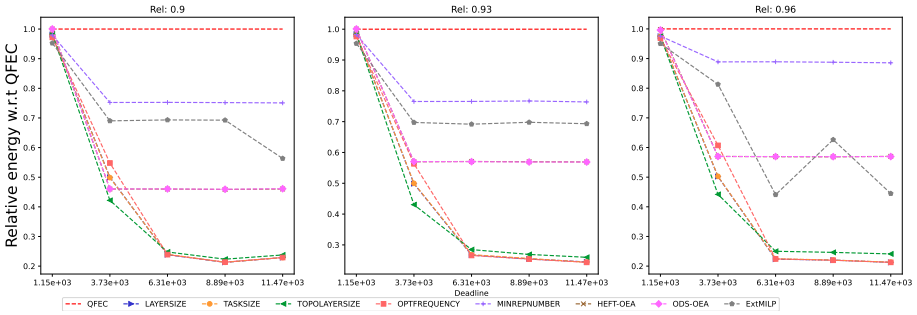

Figure 1011: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

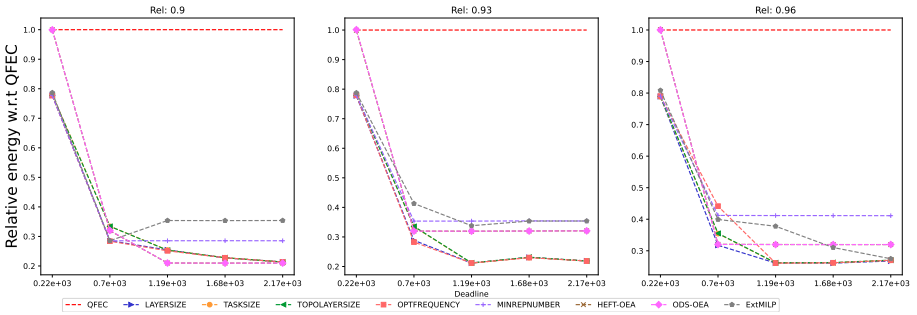

Figure 1012: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

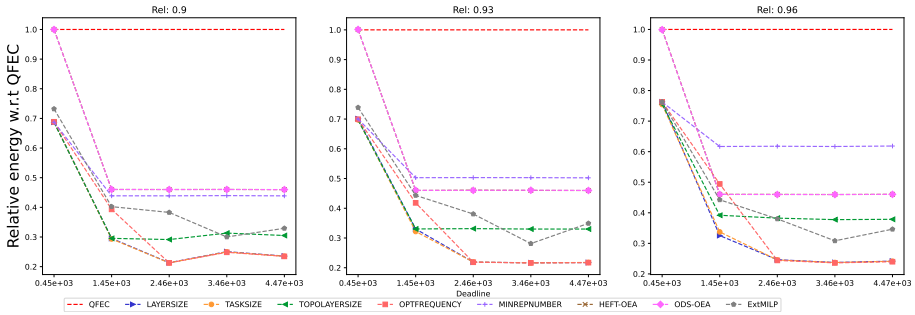

Figure 1013: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

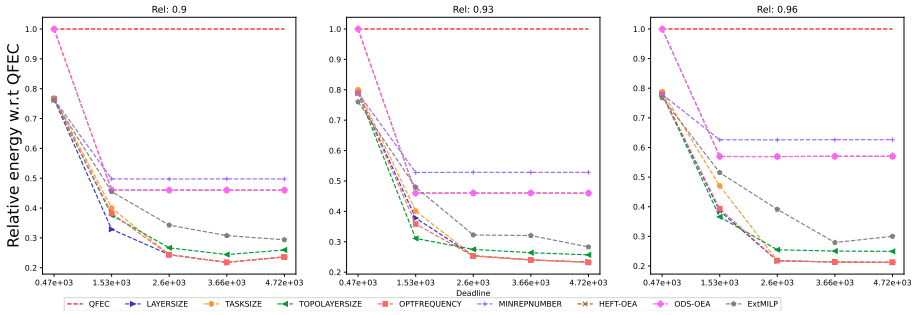

Figure 1014: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

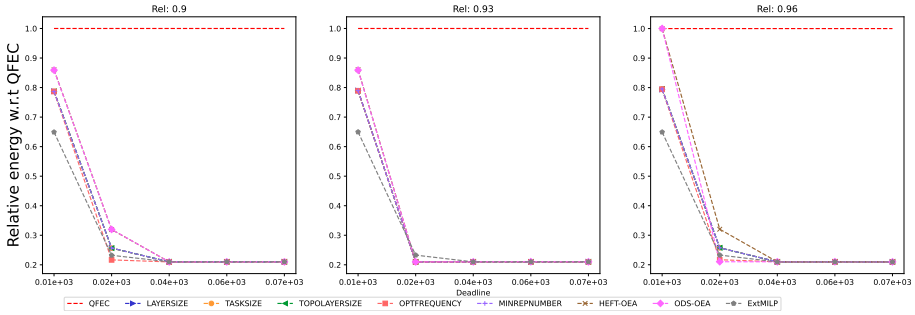

Figure 1015: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

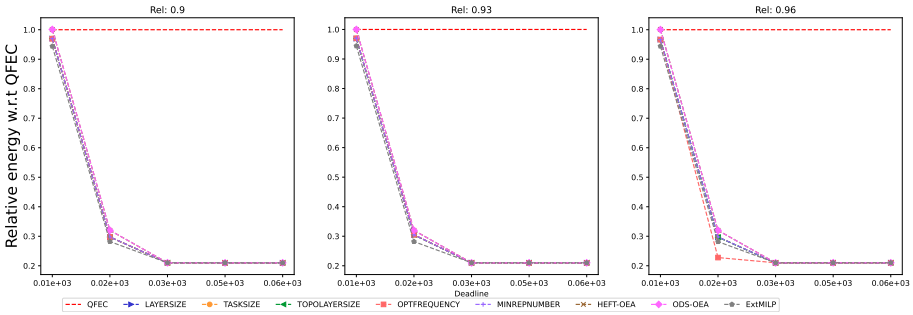

Figure 1016: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

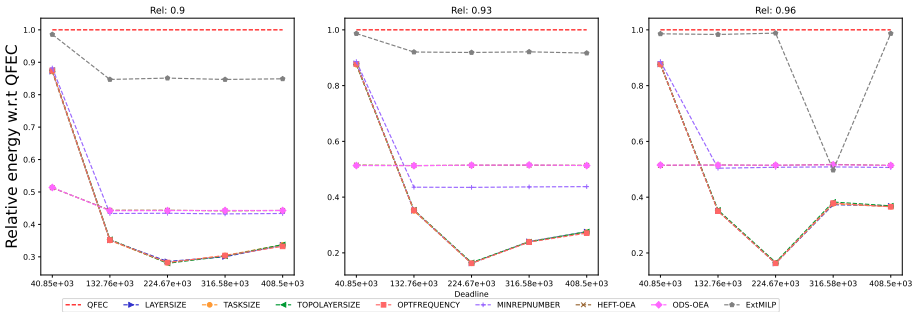

Figure 1017: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

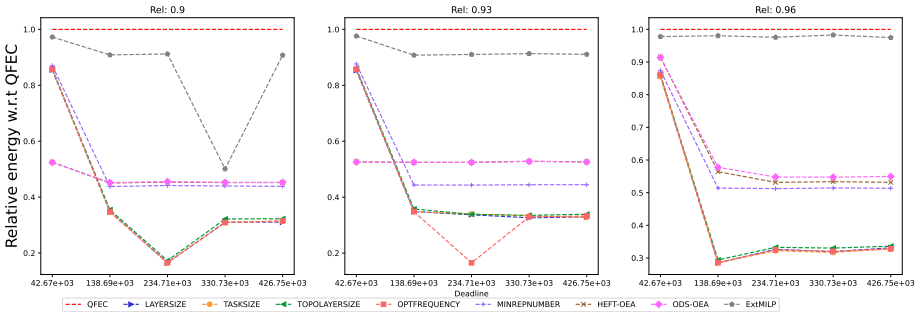

Figure 1018: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

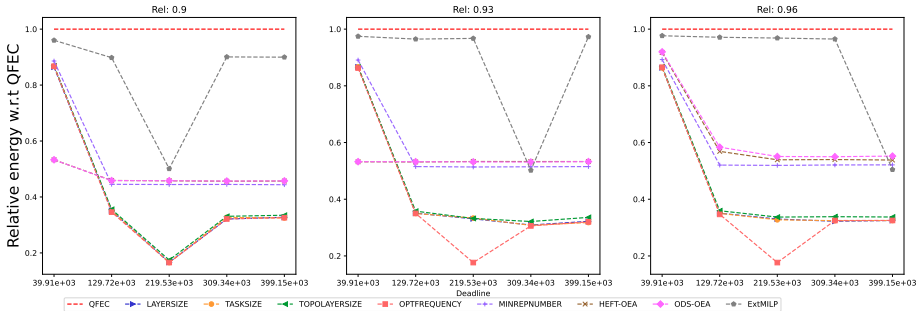

Figure 1019: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

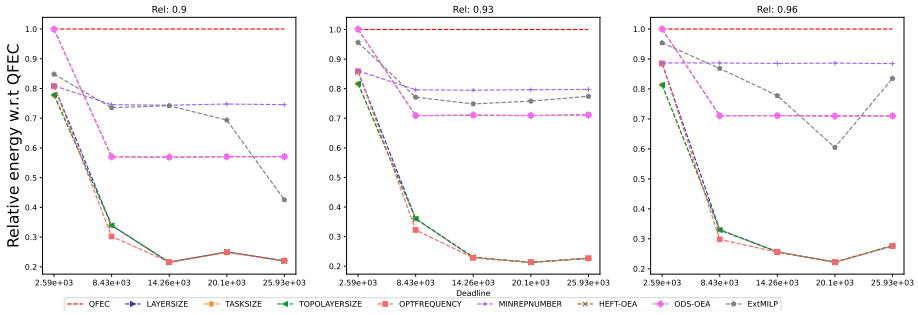

Figure 1020: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

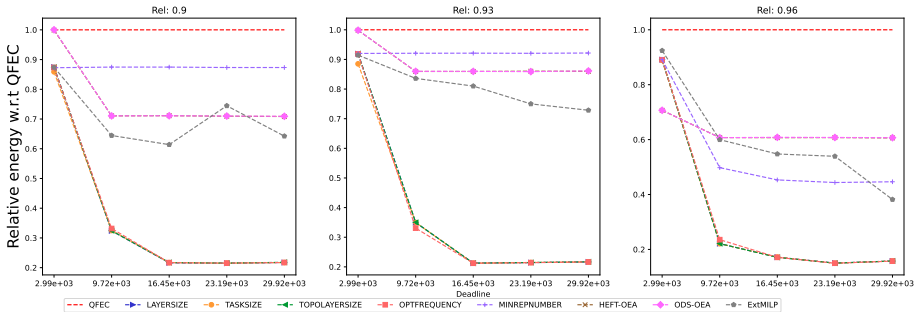

Figure 1021: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.3.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

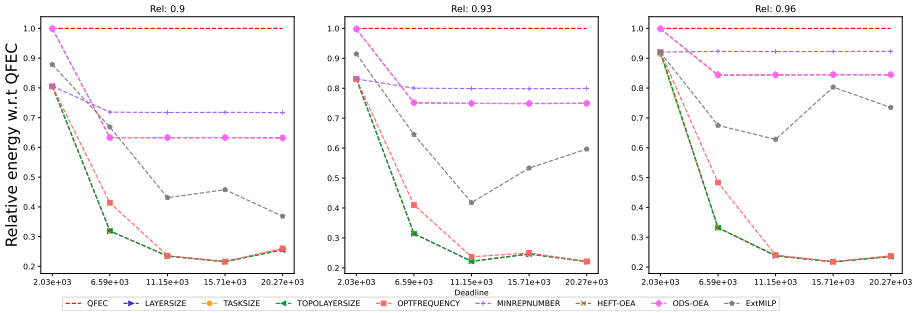

Figure 1022: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

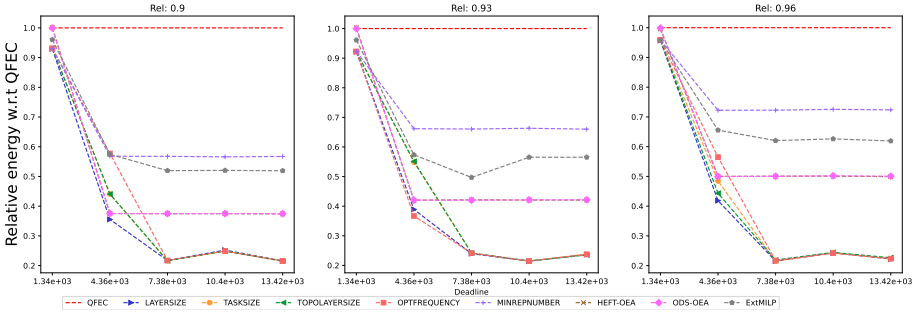

Figure 1023: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

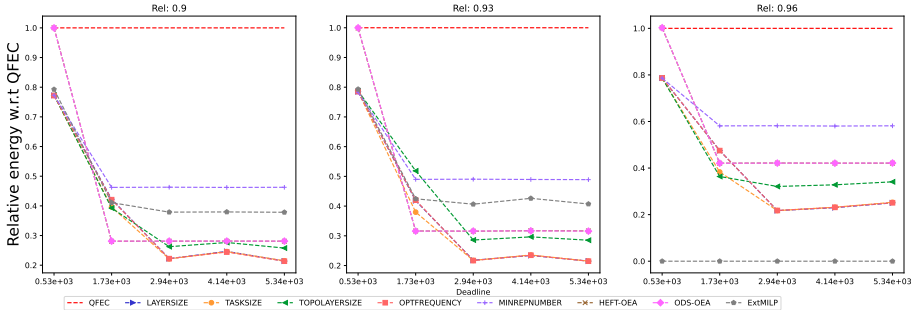

Figure 1024: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

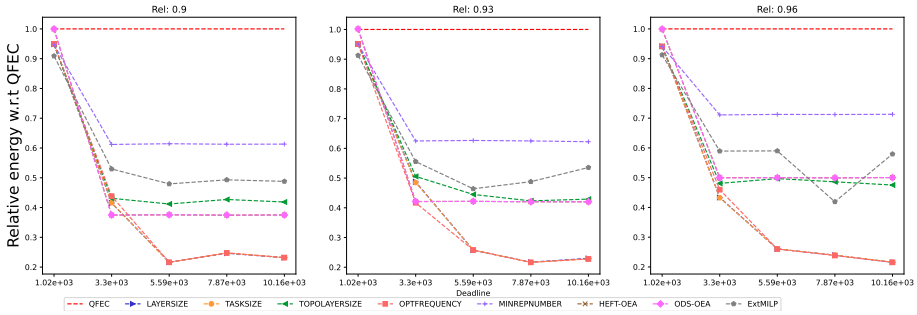

Figure 1025: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

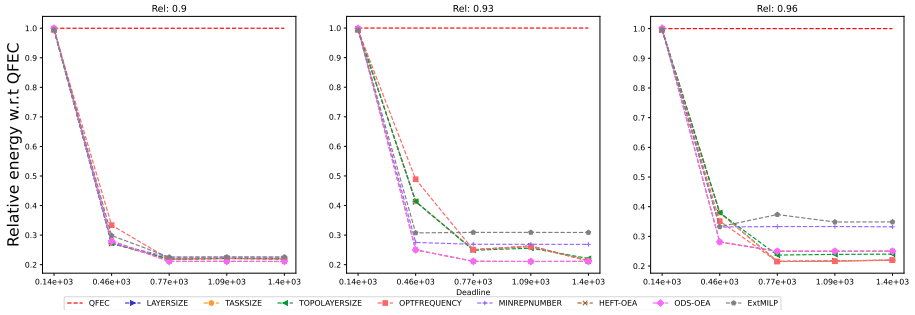

Figure 1026: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

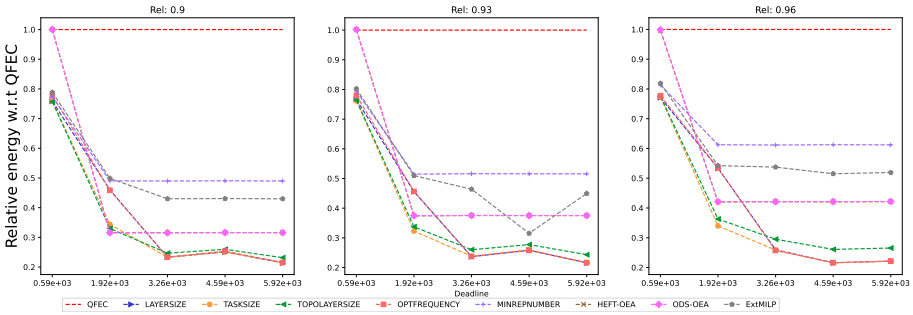

Figure 1027: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

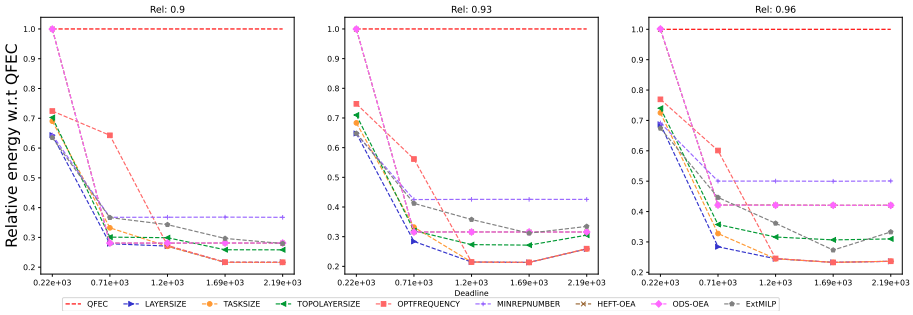

Figure 1028: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

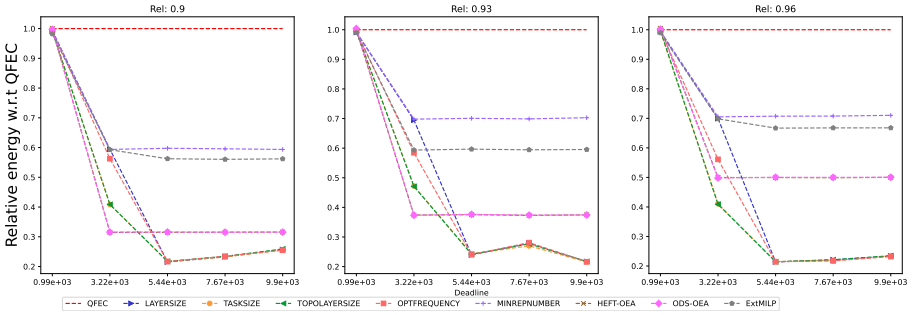

Figure 1029: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

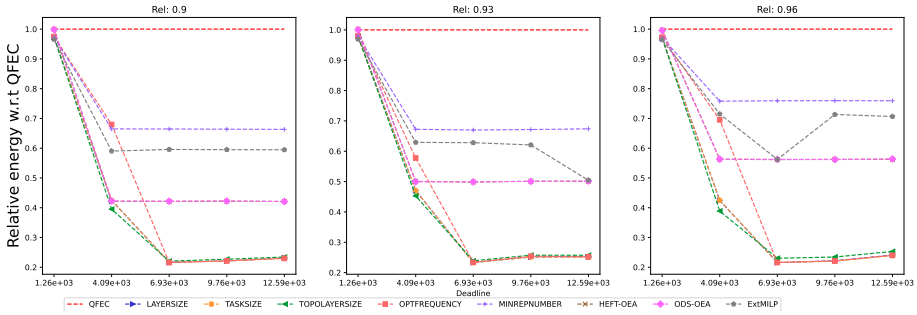

Figure 1030: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

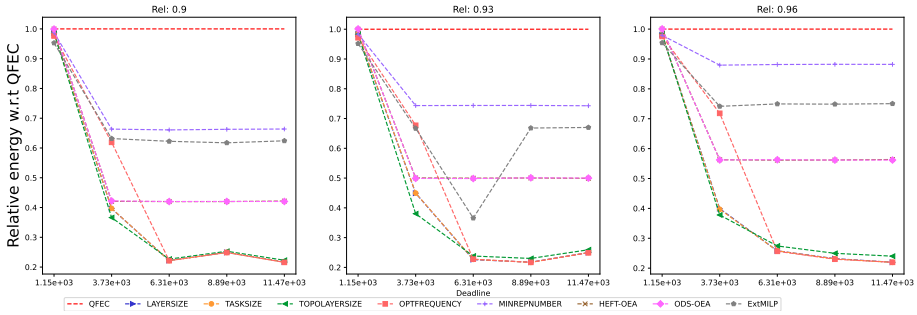

Figure 1031: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

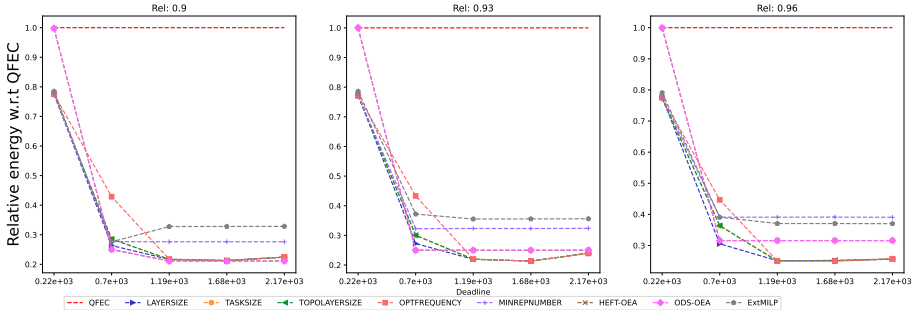

Figure 1032: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

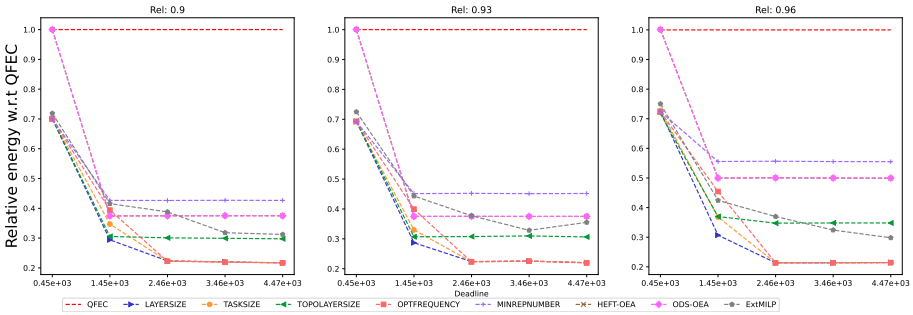

Figure 1033: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

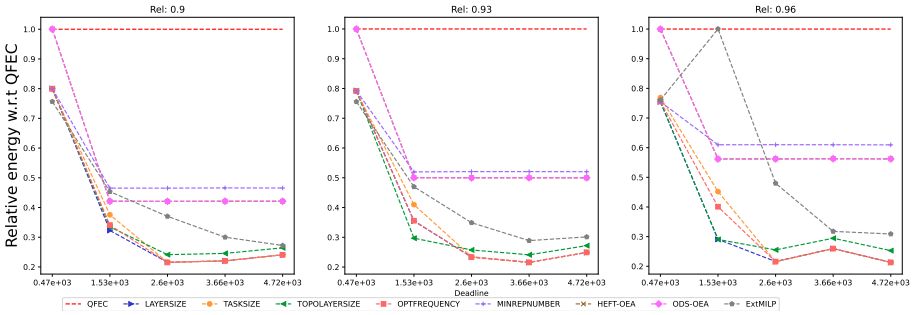

Figure 1034: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

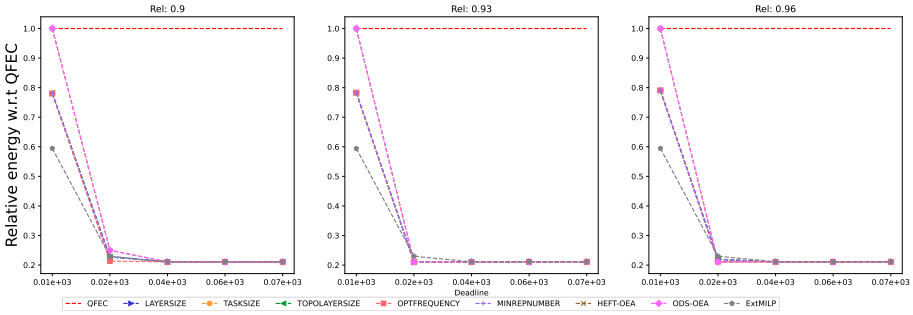

Figure 1035: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

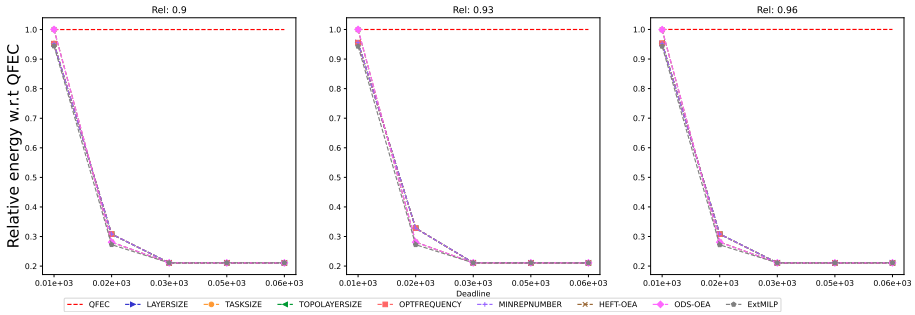

Figure 1036: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

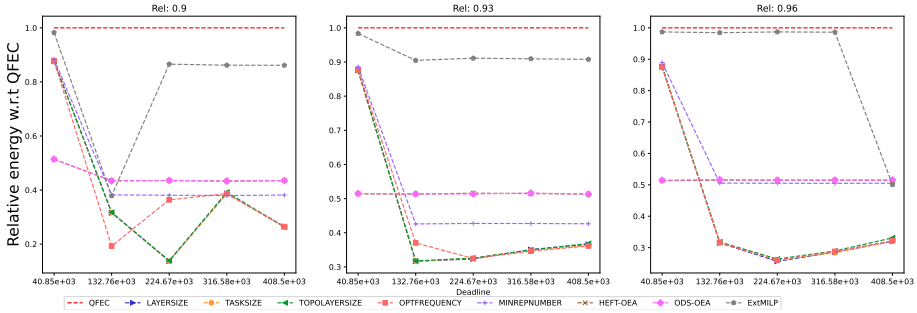

Figure 1037: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

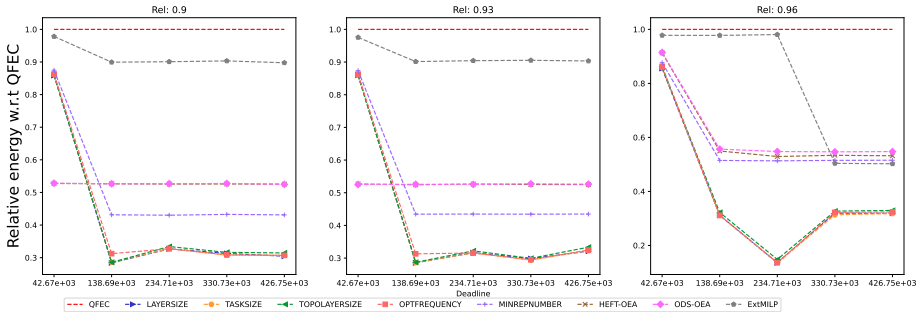

Figure 1038: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

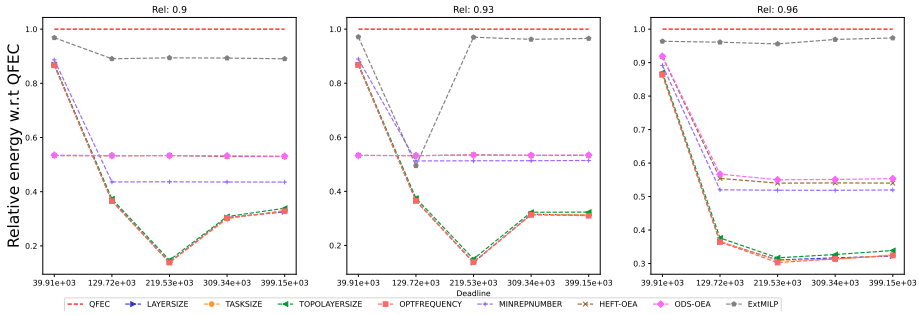

Figure 1039: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

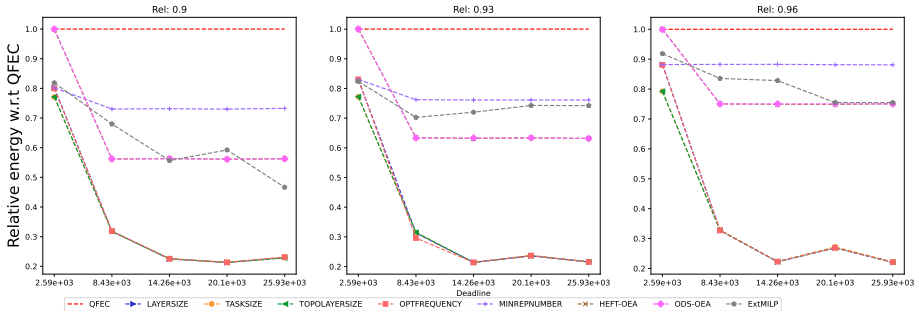

Figure 1040: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

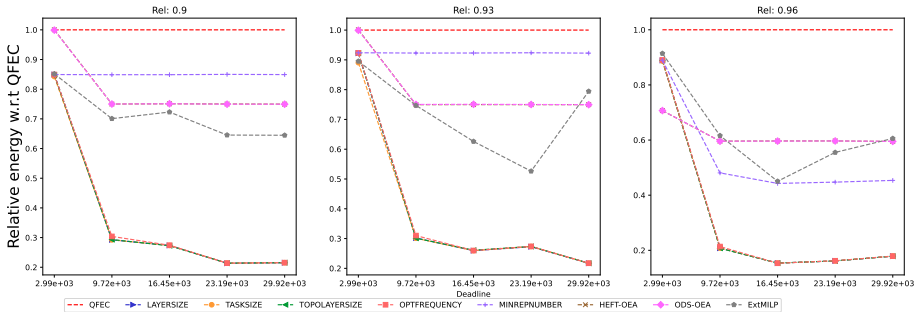

Figure 1041: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.4 $BC/WC = 0.4$

F.4.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

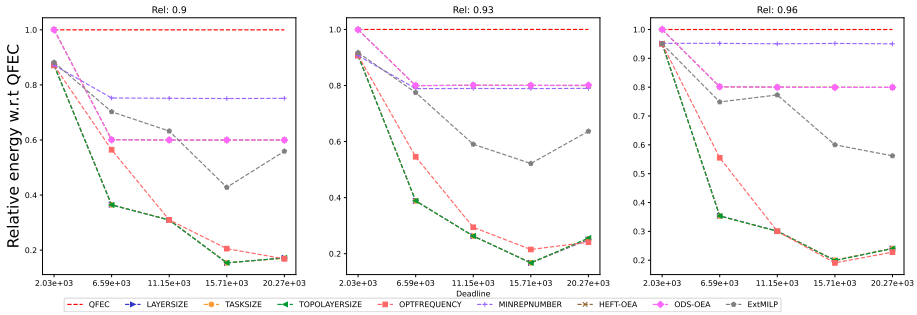

Figure 1042: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

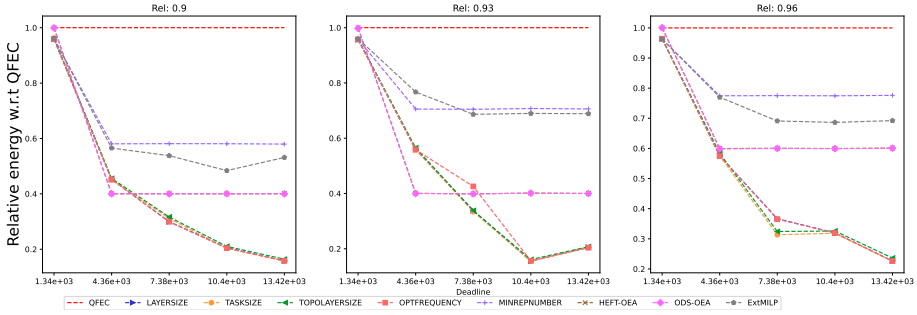

Figure 1043: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

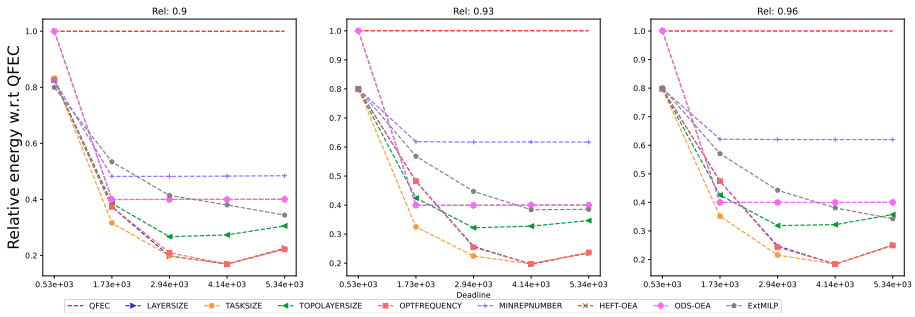

Figure 1044: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

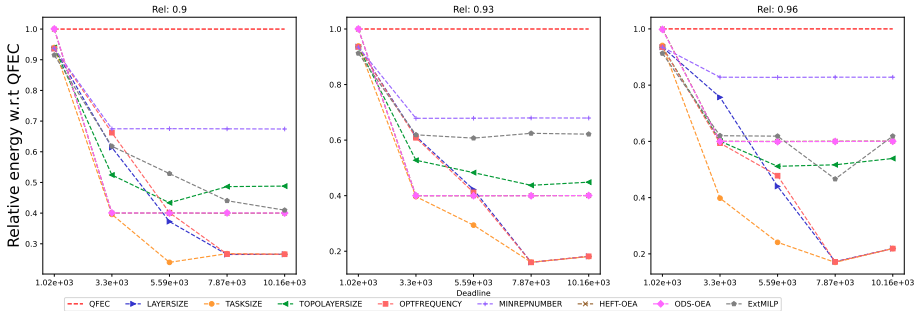

Figure 1045: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

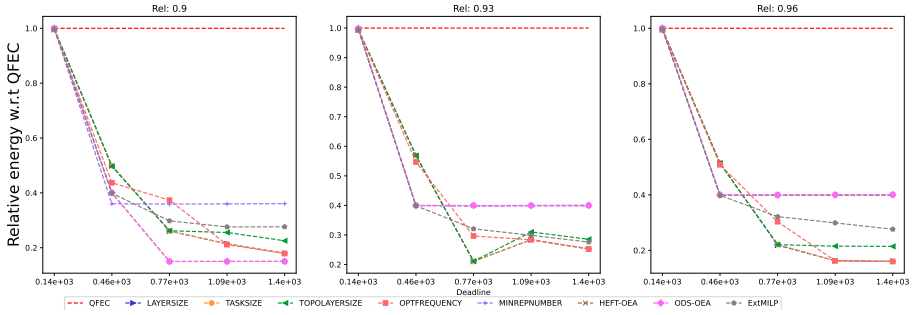

Figure 1046: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

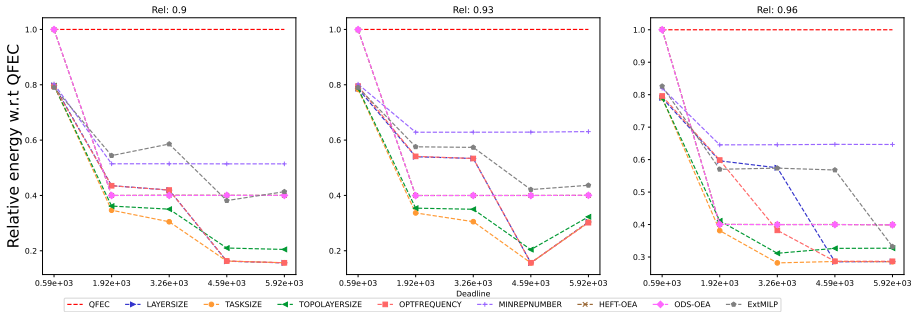

Figure 1047: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

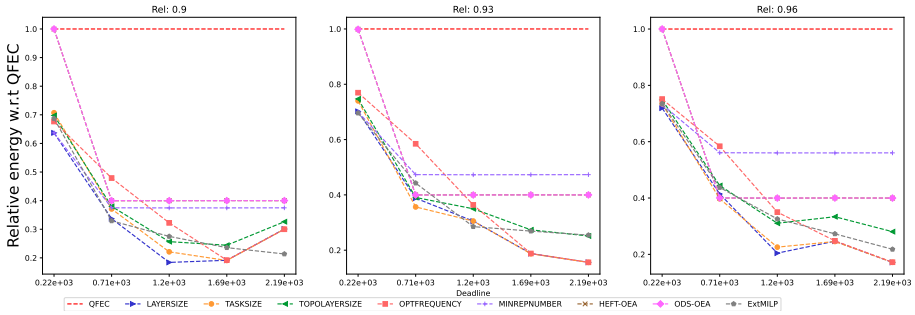

Figure 1048: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

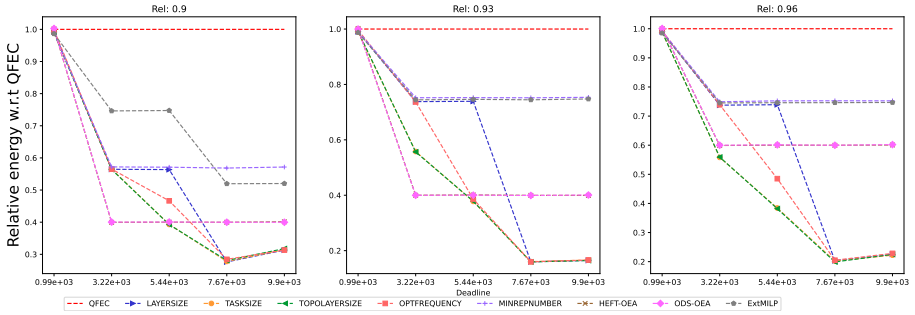

Figure 1049: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

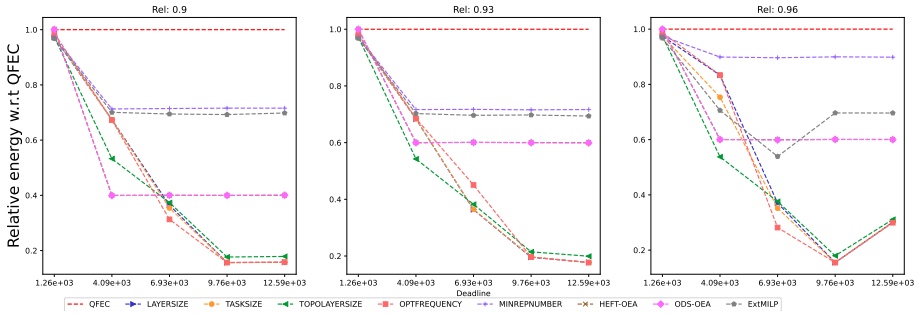

Figure 1050: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

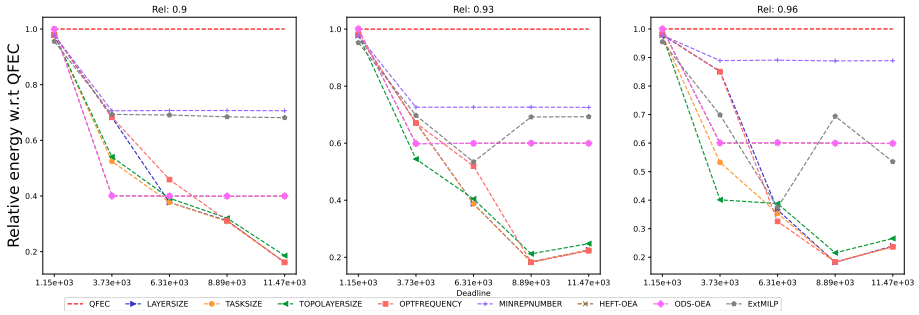

Figure 1051: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

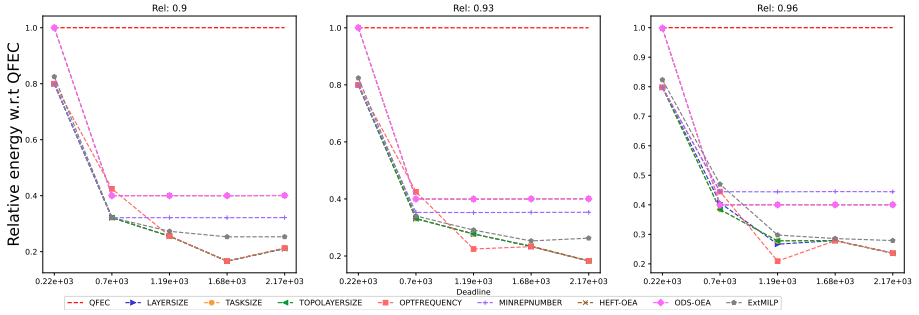

Figure 1052: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

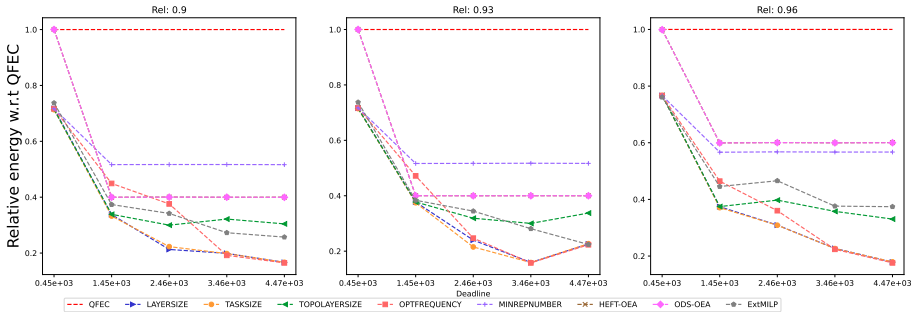

Figure 1053: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

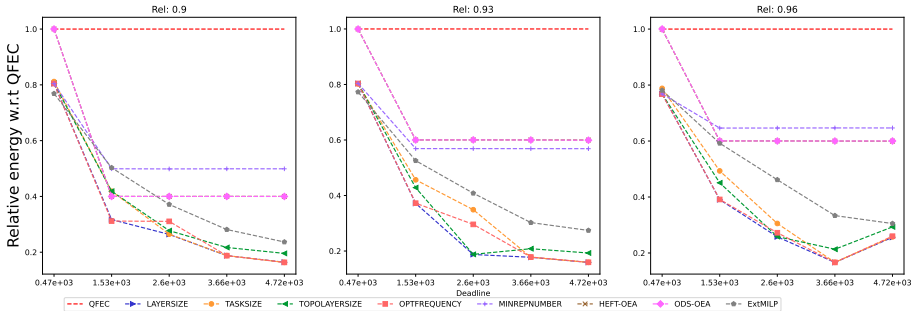

Figure 1054: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

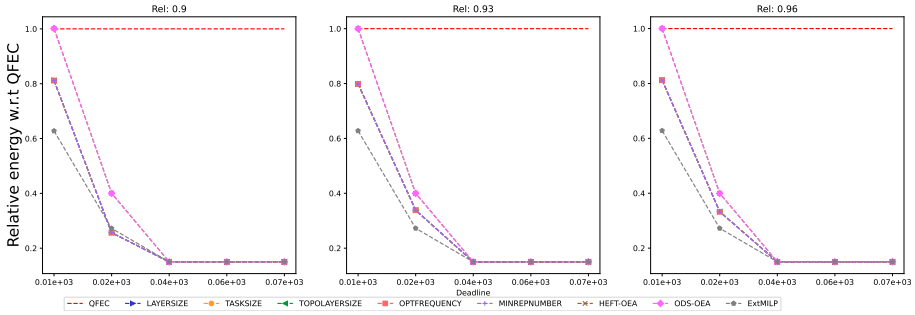

Figure 1055: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

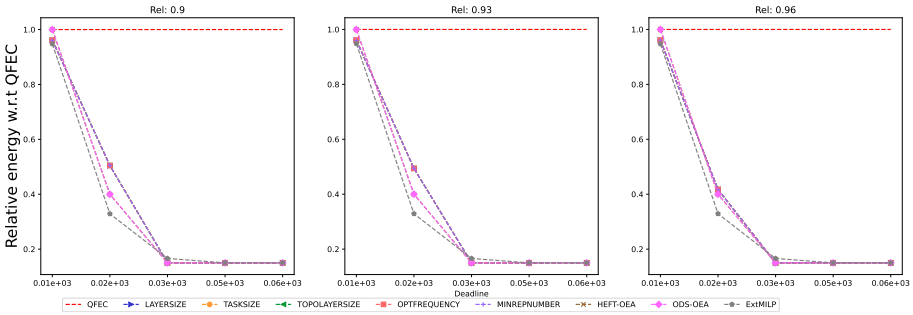

Figure 1056: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

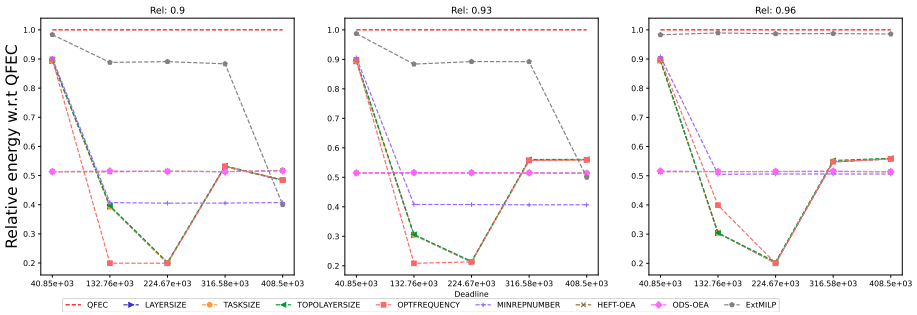

Figure 1057: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

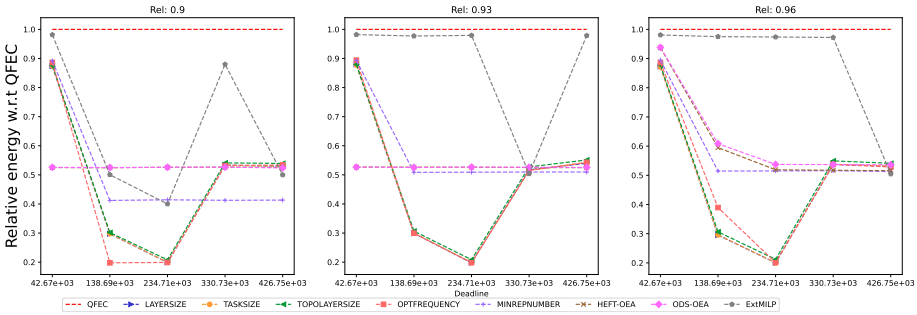

Figure 1058: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

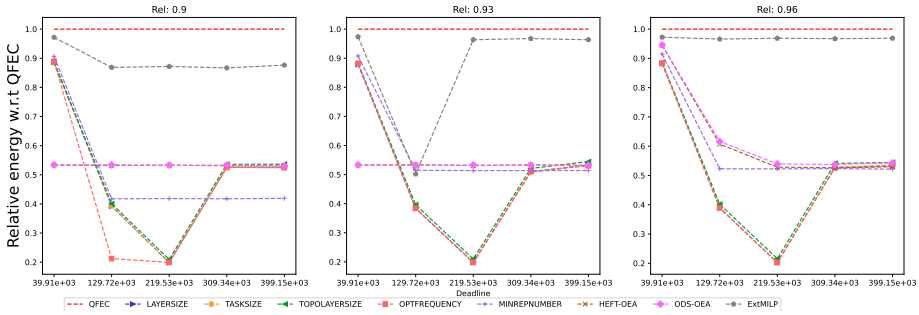

Figure 1059: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

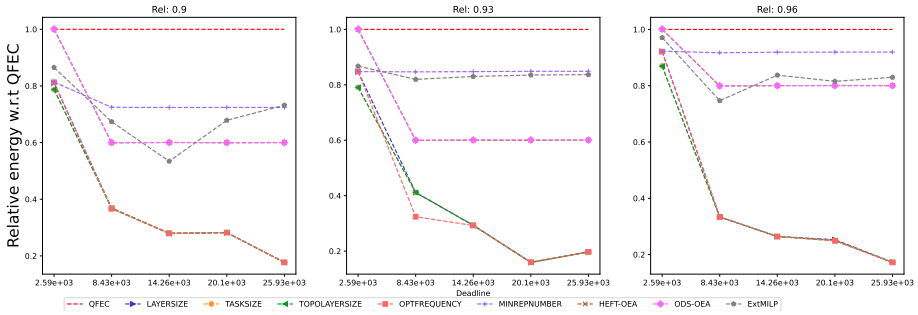

Figure 1060: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

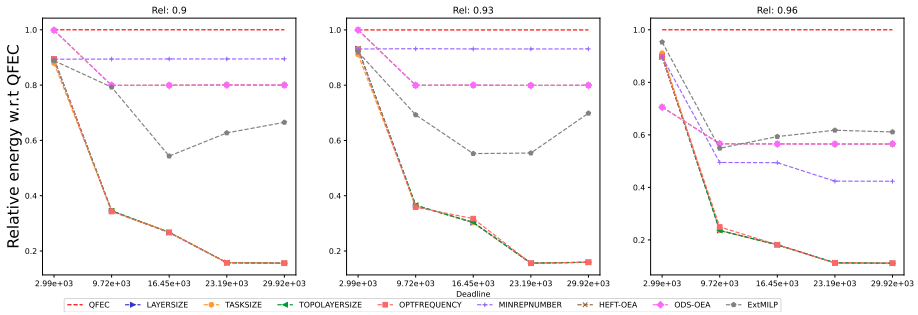

Figure 1061: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.4.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

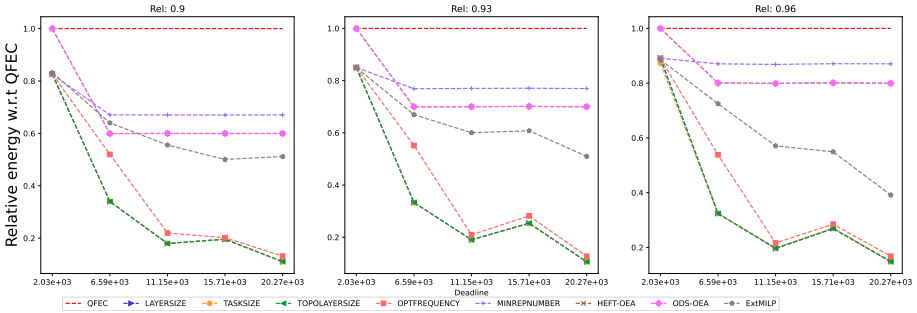

Figure 1062: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

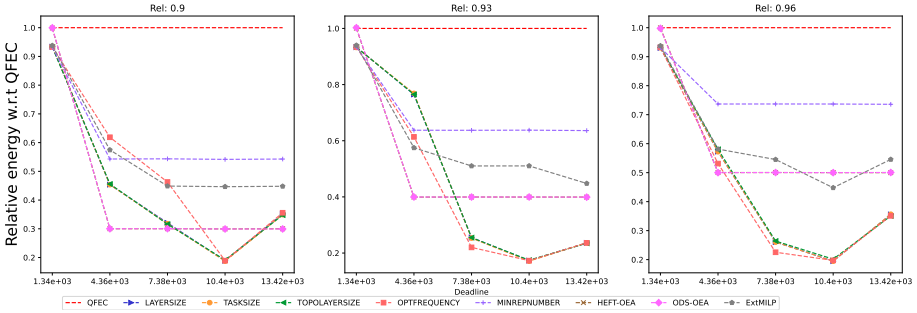

Figure 1063: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

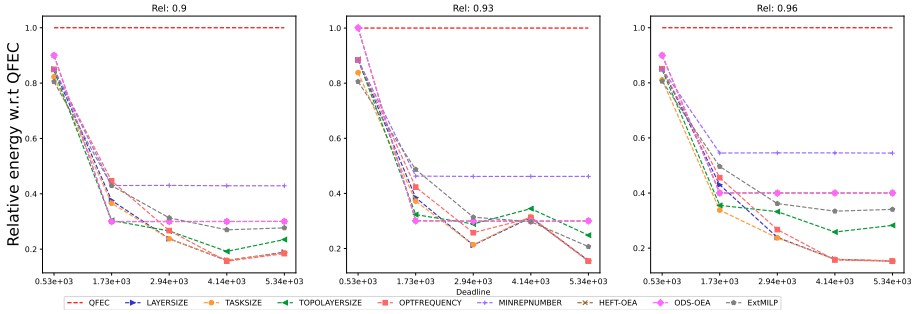

Figure 1064: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

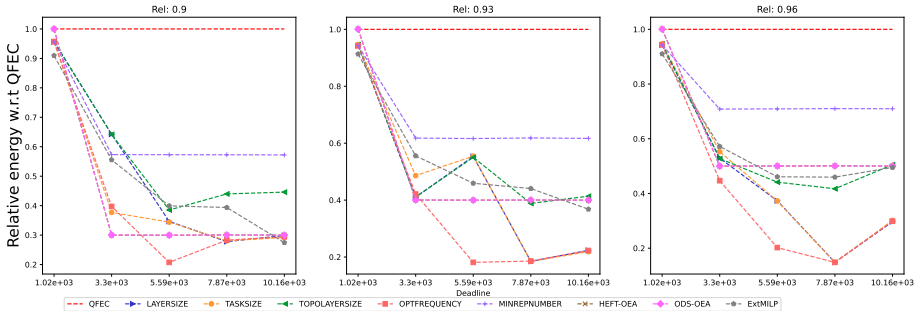

Figure 1065: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

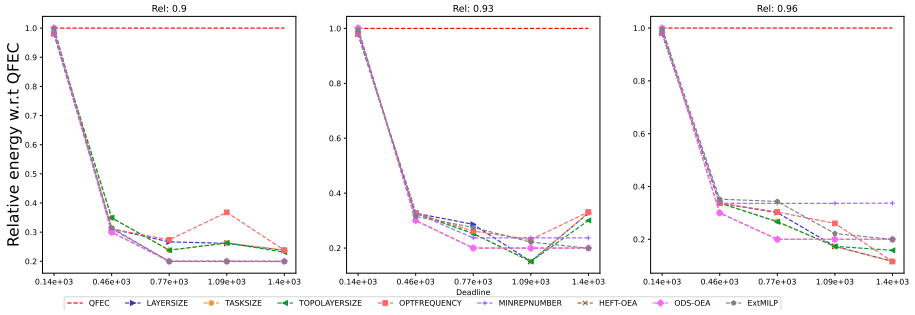

Figure 1066: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

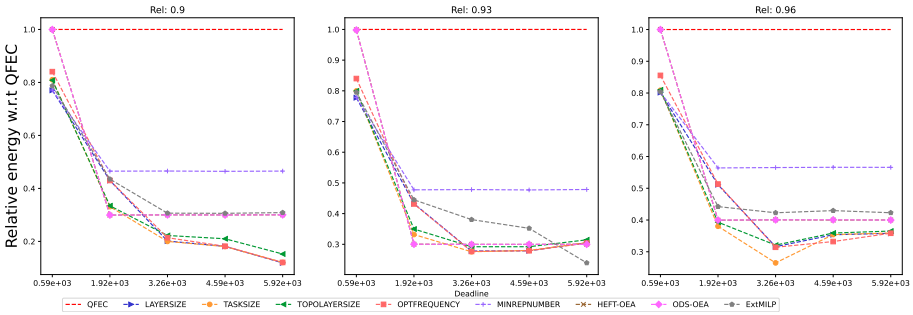

Figure 1067: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

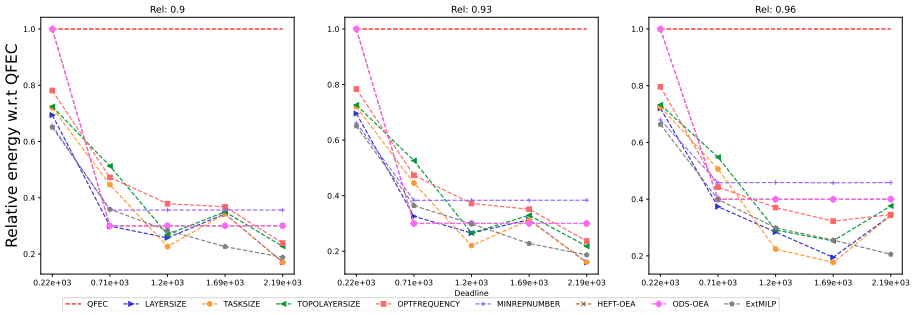

Figure 1068: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

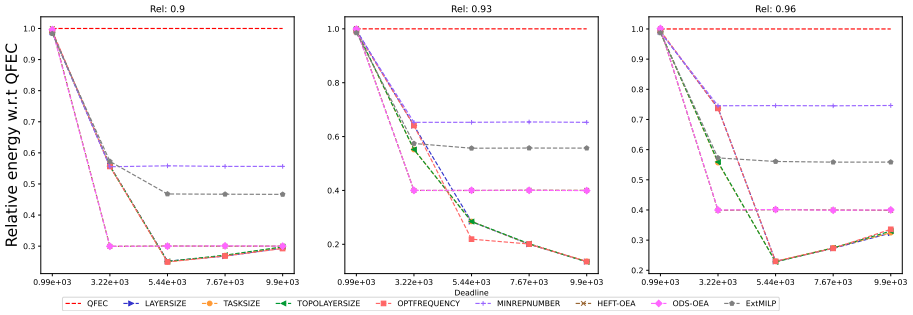

Figure 1069: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

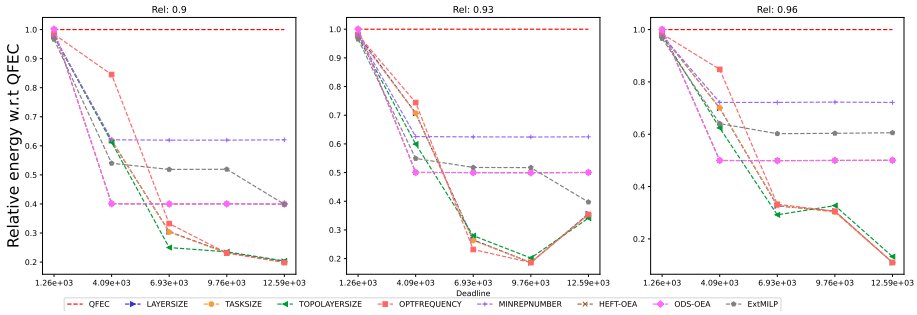

Figure 1070: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

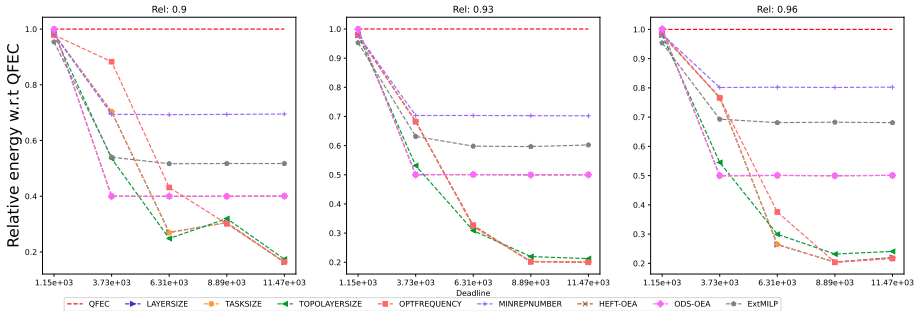

Figure 1071: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

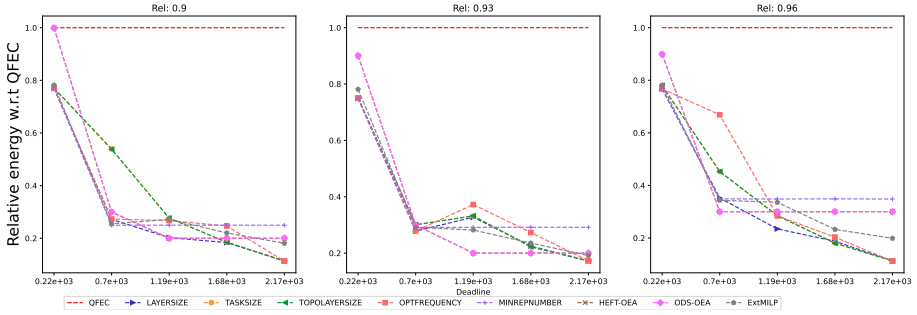

Figure 1072: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

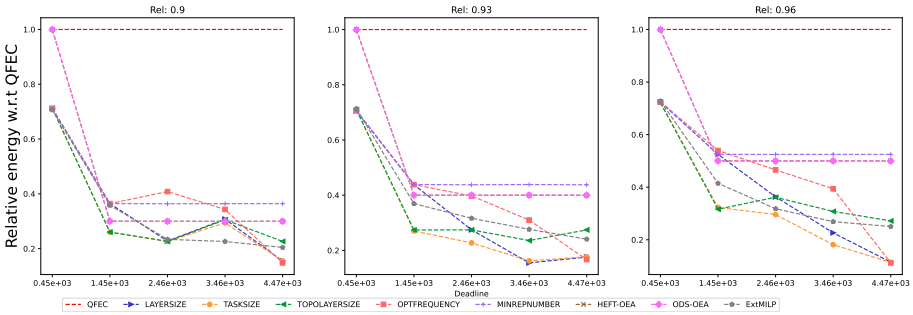

Figure 1073: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

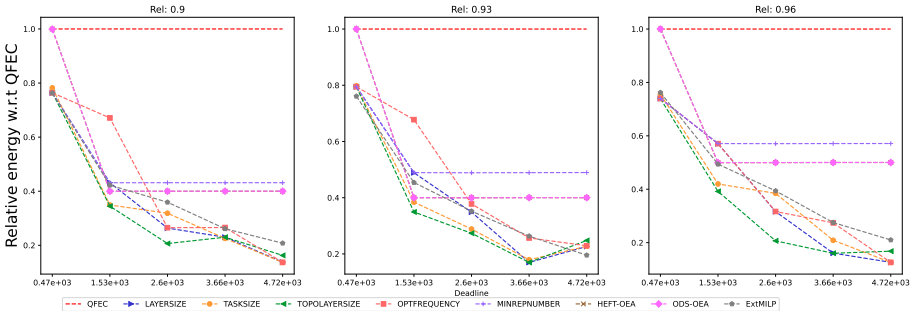

Figure 1074: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

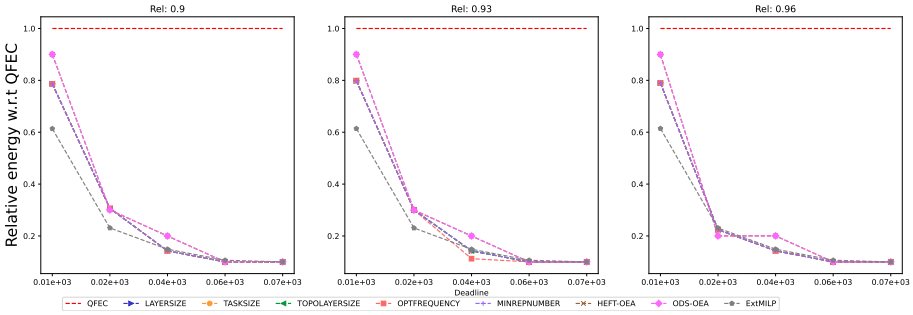

Figure 1075: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

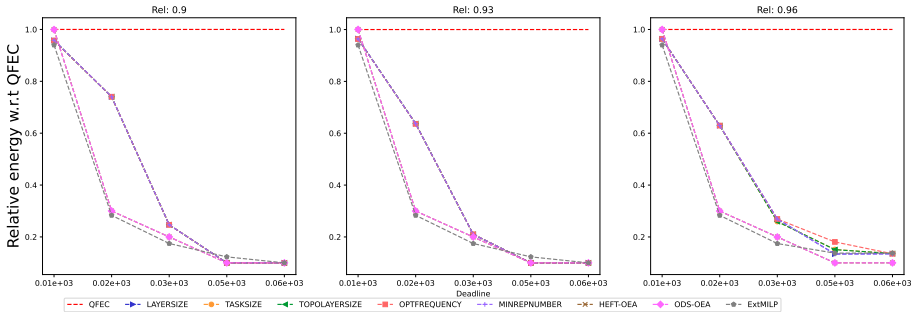

Figure 1076: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

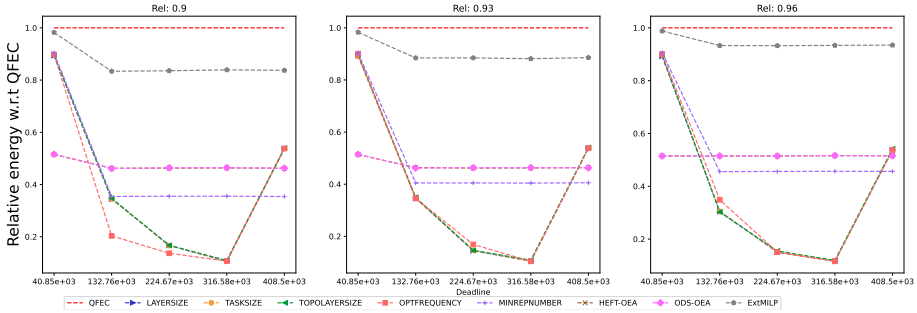

Figure 1077: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

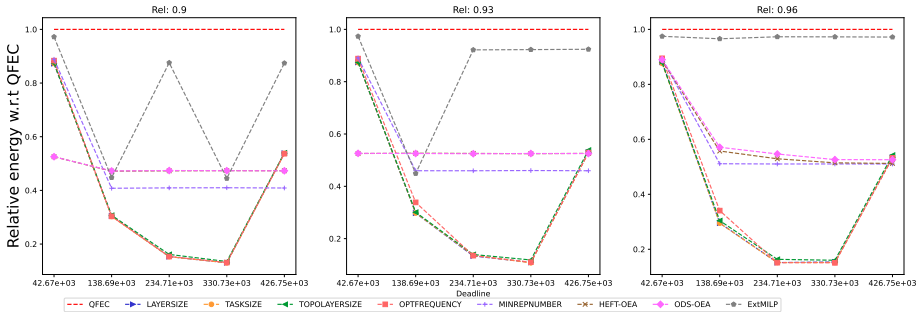

Figure 1078: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

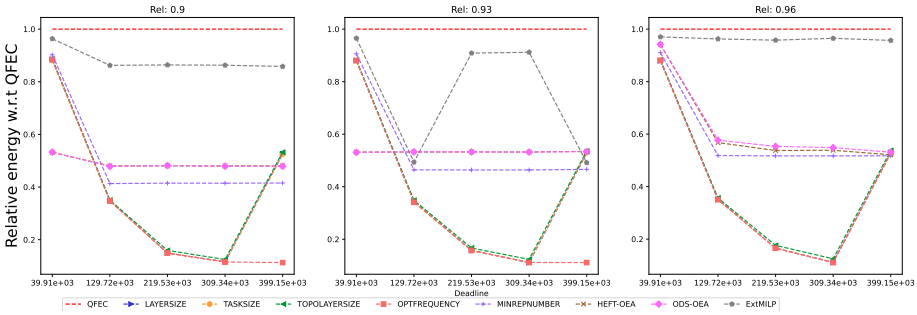

Figure 1079: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

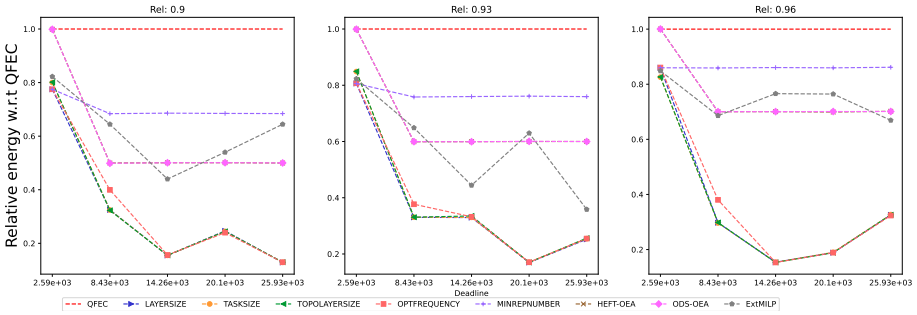

Figure 1080: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

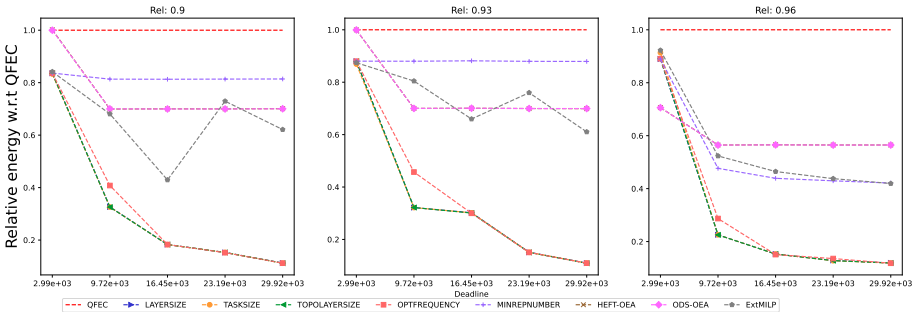

Figure 1081: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.4.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

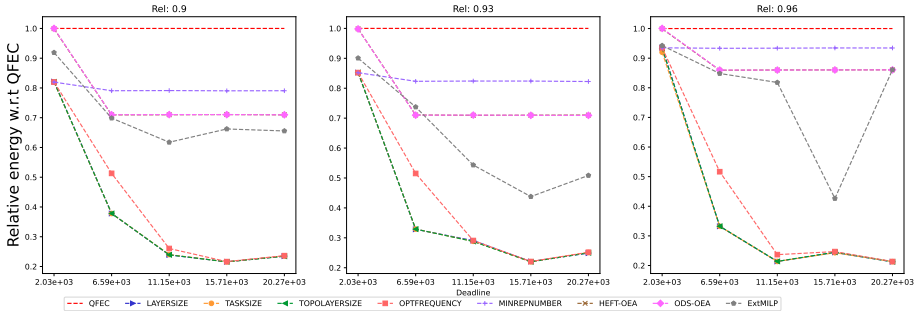

Figure 1082: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

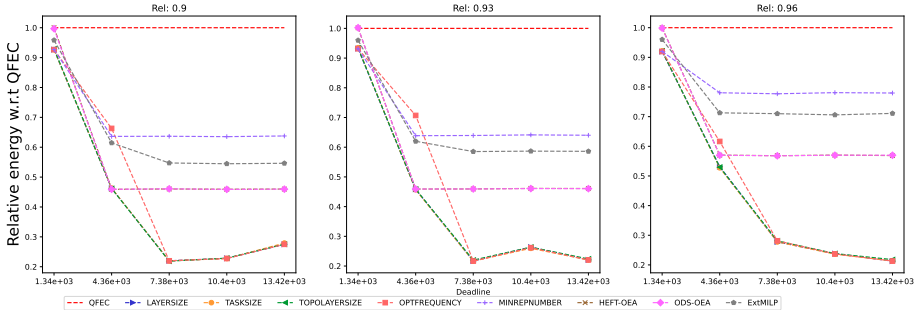

Figure 1083: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

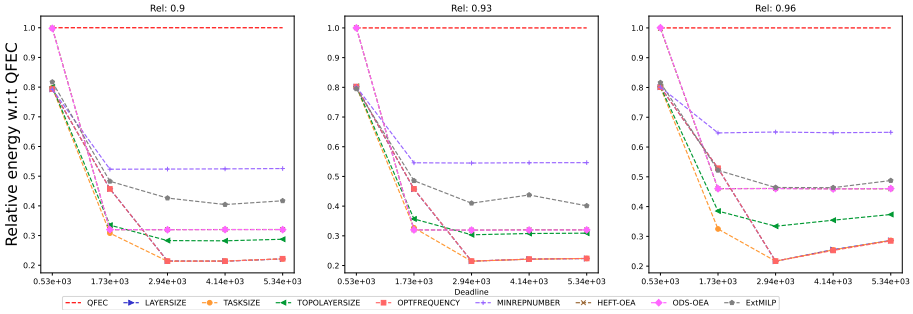

Figure 1084: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

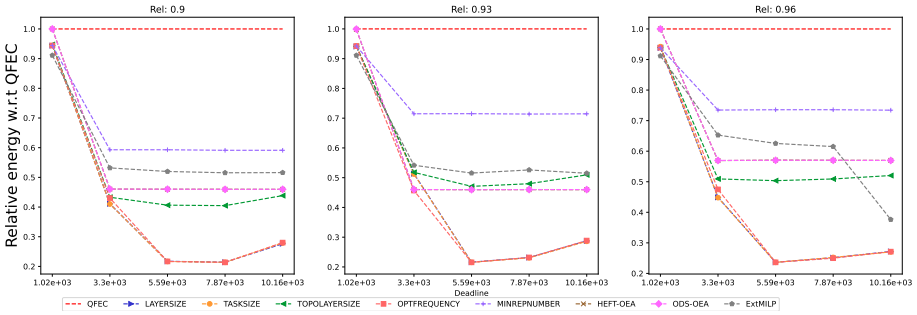

Figure 1085: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

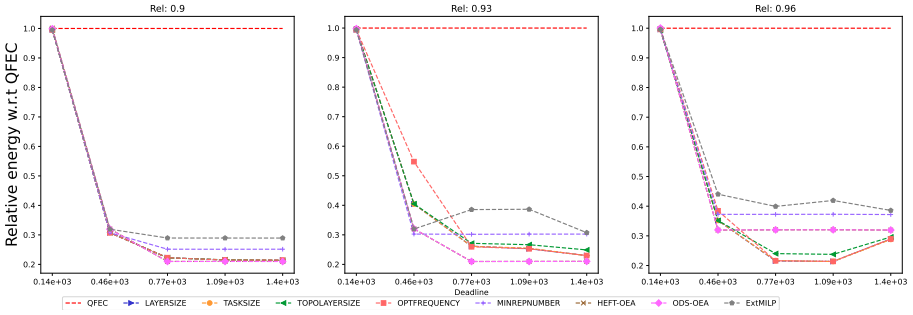

Figure 1086: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

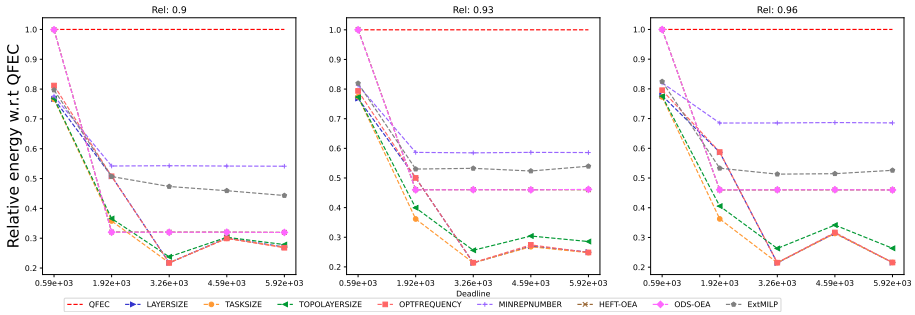

Figure 1087: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

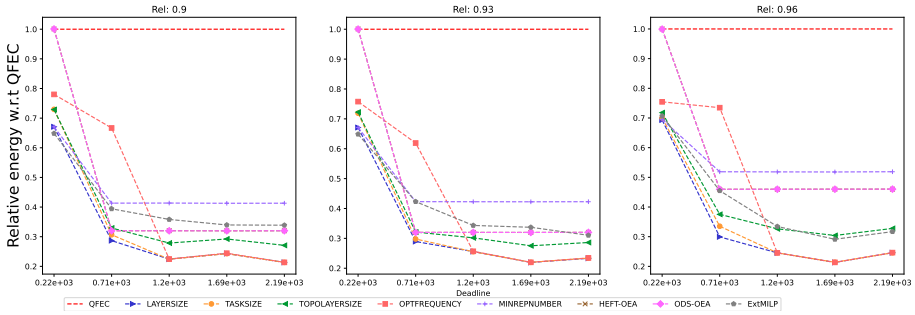

Figure 1088: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

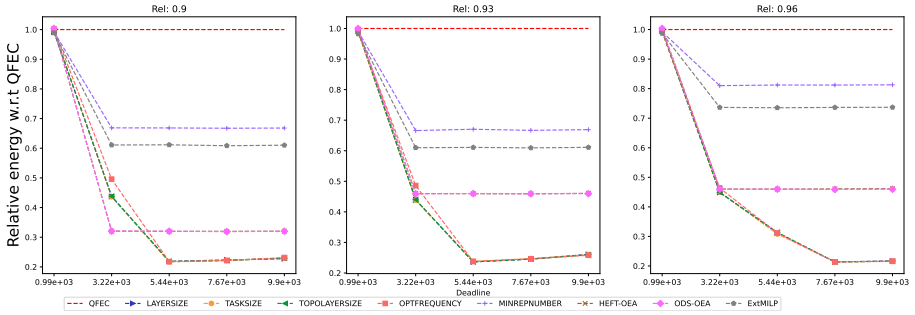

Figure 1089: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

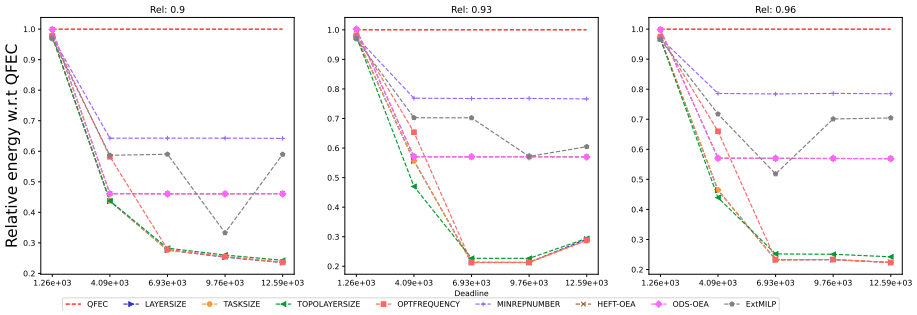

Figure 1090: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

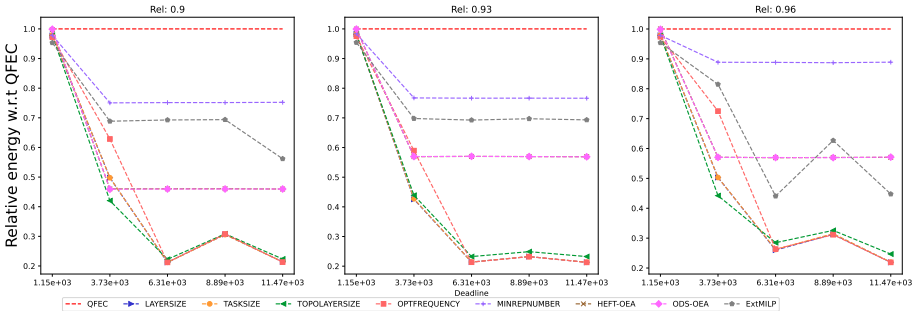

Figure 1091: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

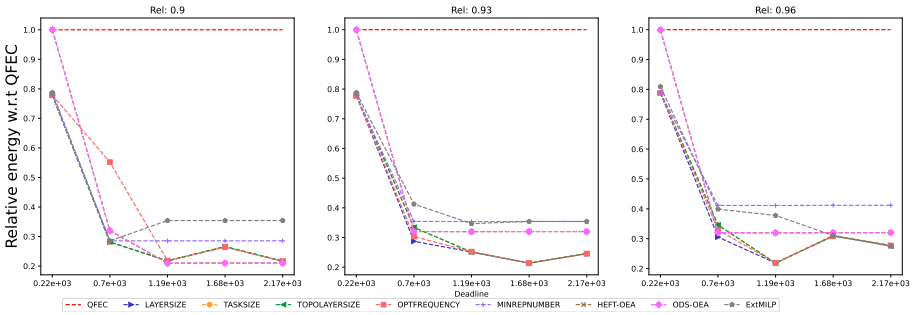

Figure 1092: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

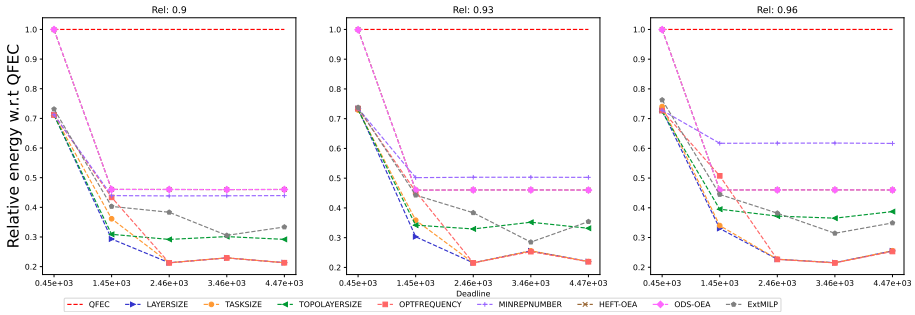

Figure 1093: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

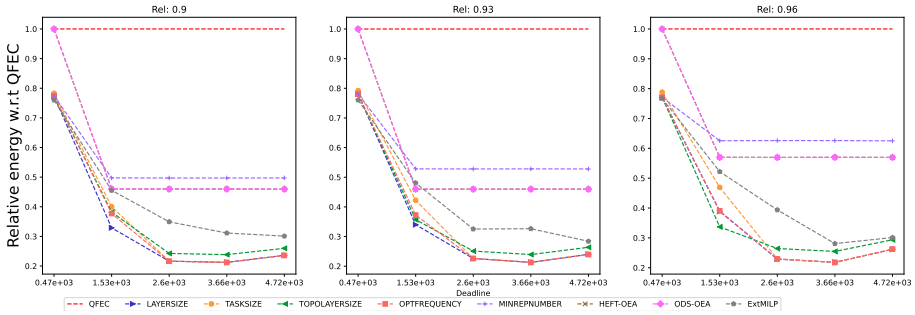

Figure 1094: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

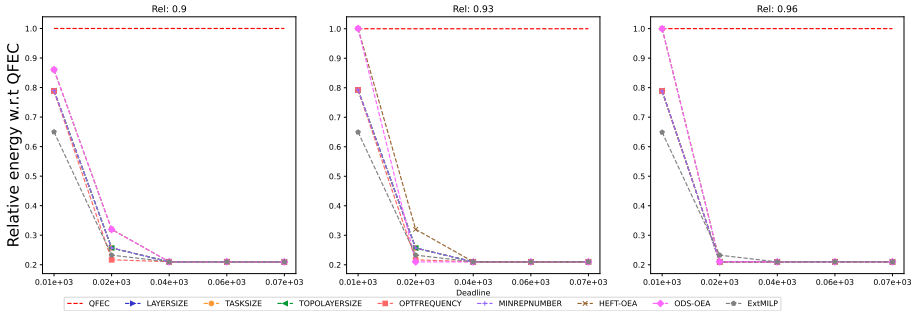

Figure 1095: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

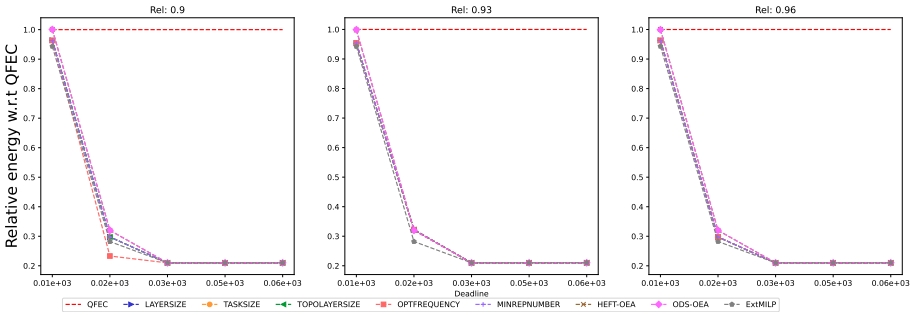

Figure 1096: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

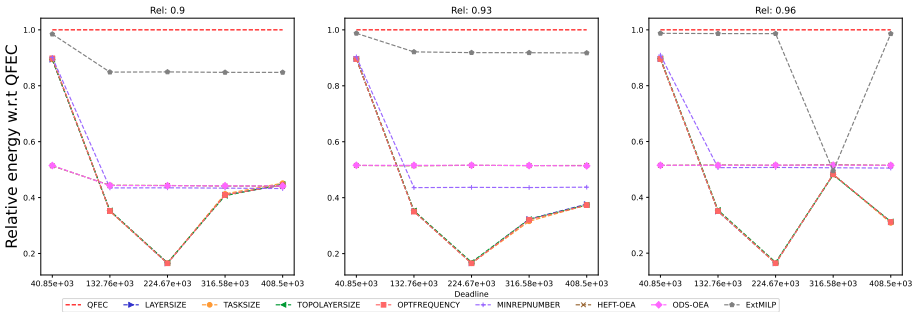

Figure 1097: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

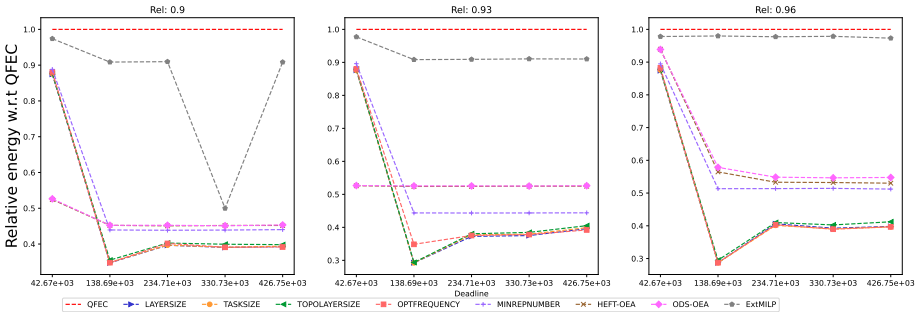

Figure 1098: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

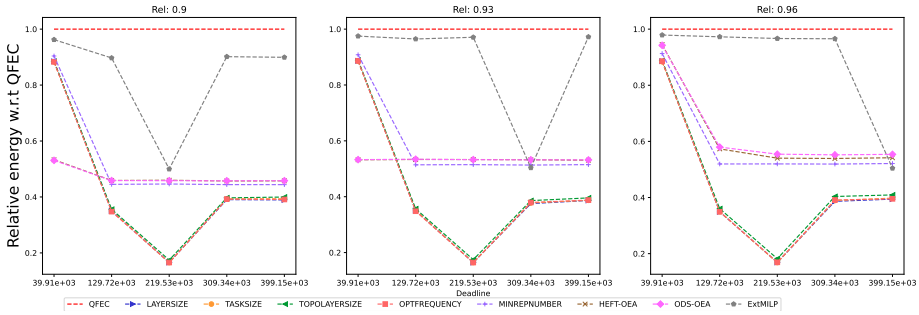

Figure 1099: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

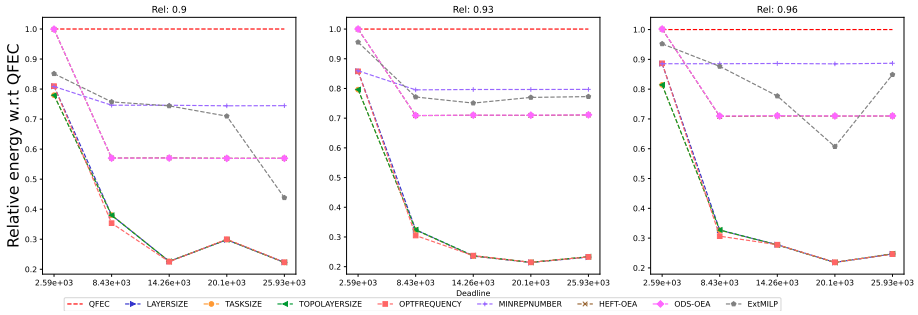

Figure 1100: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

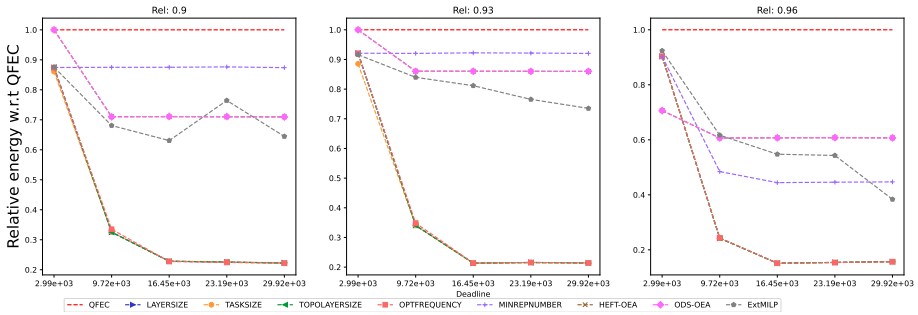

Figure 1101: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.4.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

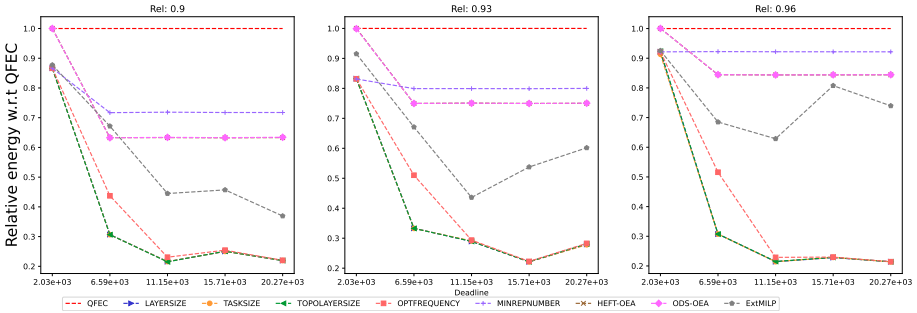

Figure 1102: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

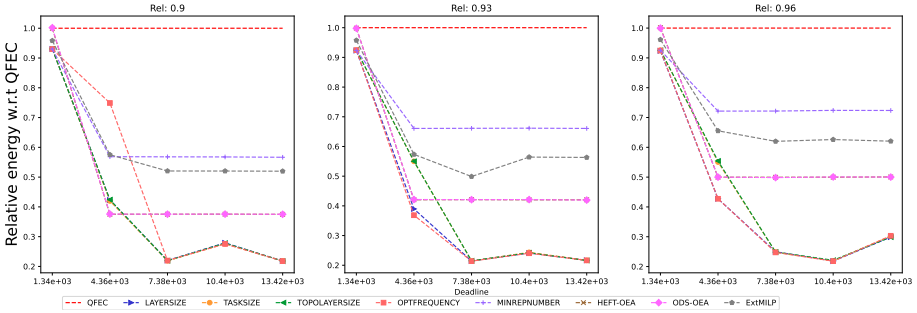

Figure 1103: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

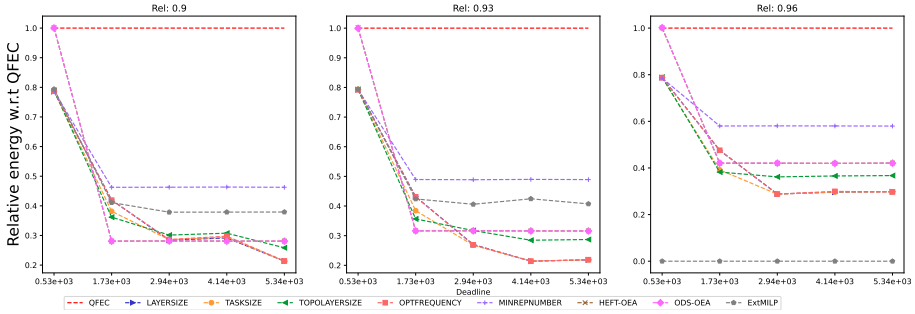

Figure 1104: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

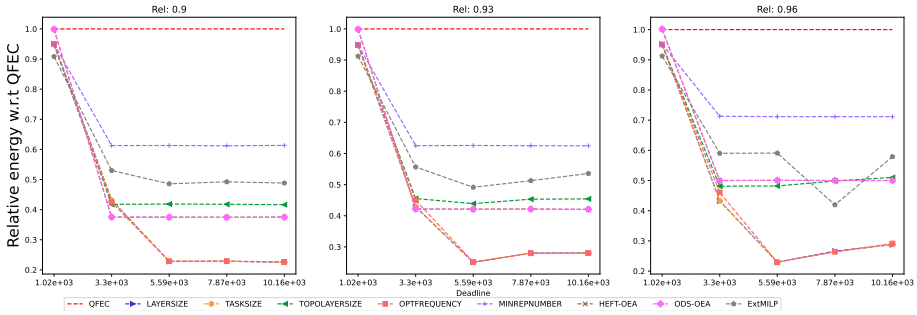

Figure 1105: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

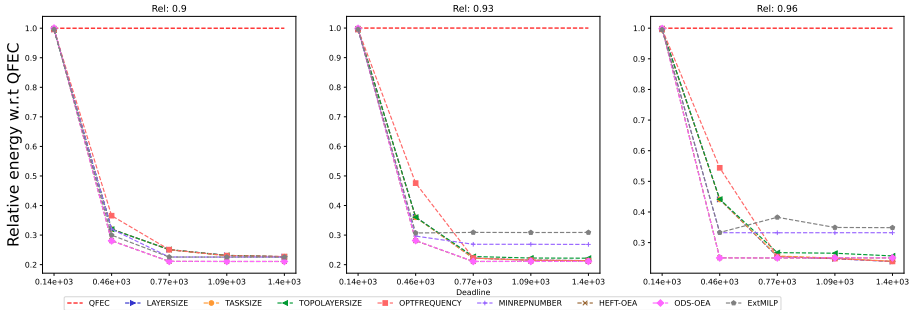

Figure 1106: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

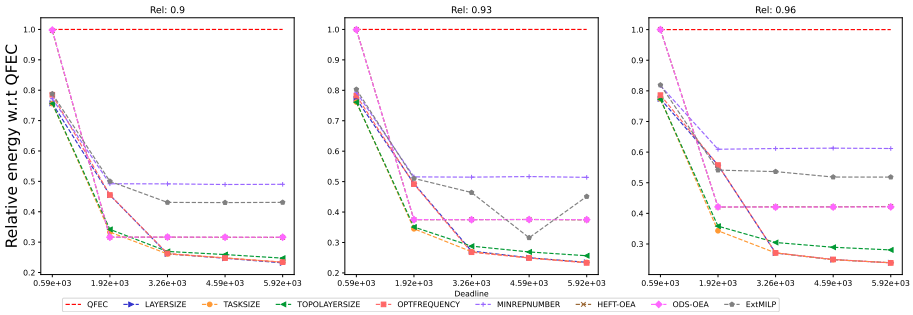

Figure 1107: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

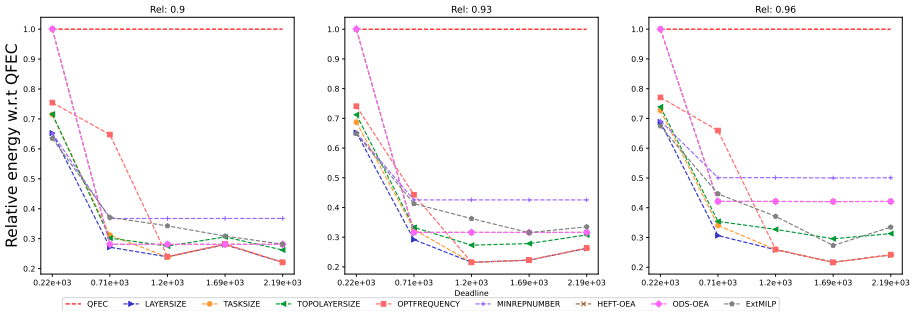

Figure 1108: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

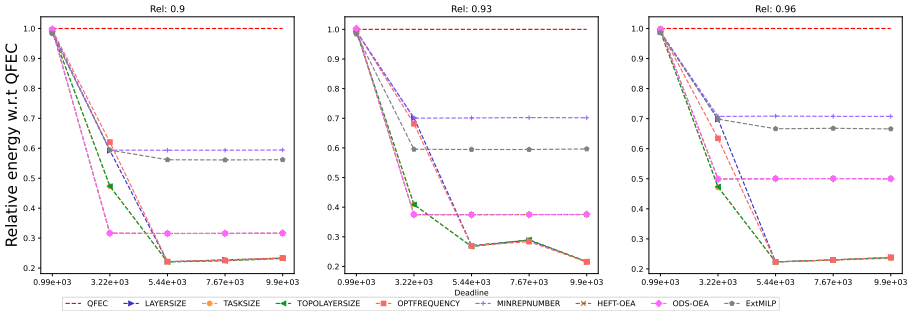

Figure 1109: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

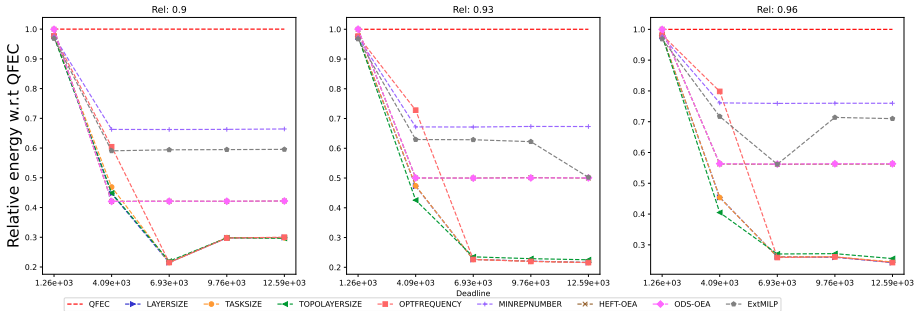

Figure 1110: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

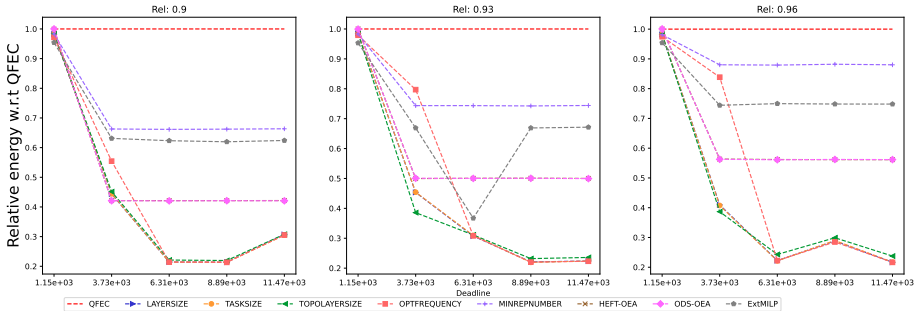

Figure 1111: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

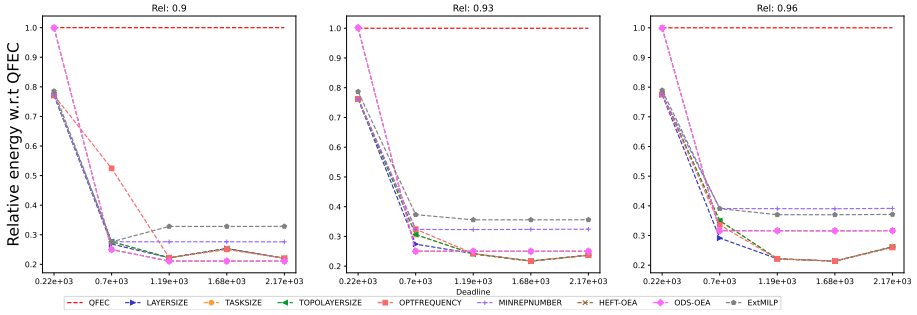

Figure 1112: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

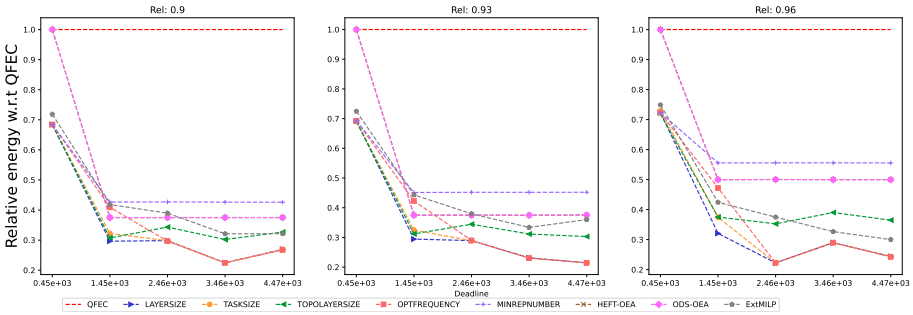

Figure 1113: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

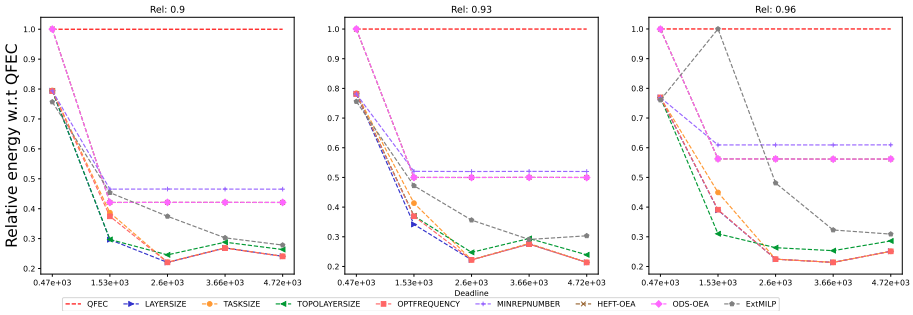

Figure 1114: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

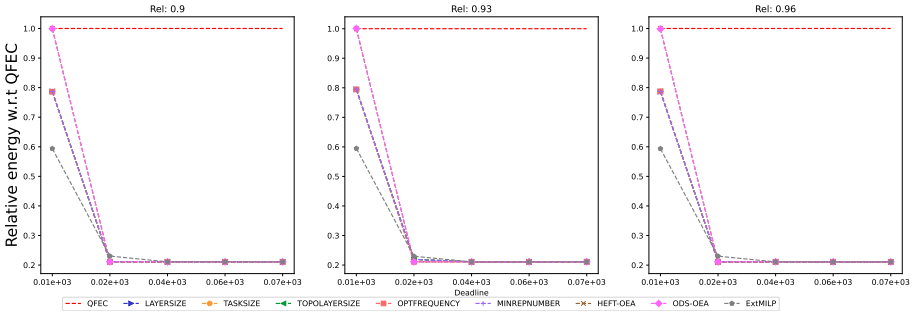

Figure 1115: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

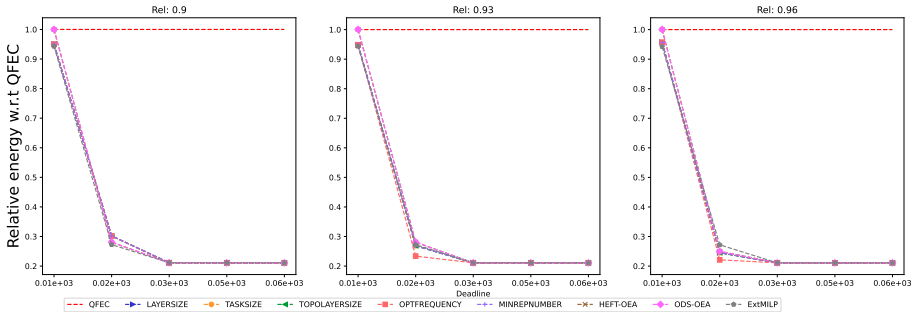

Figure 1116: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

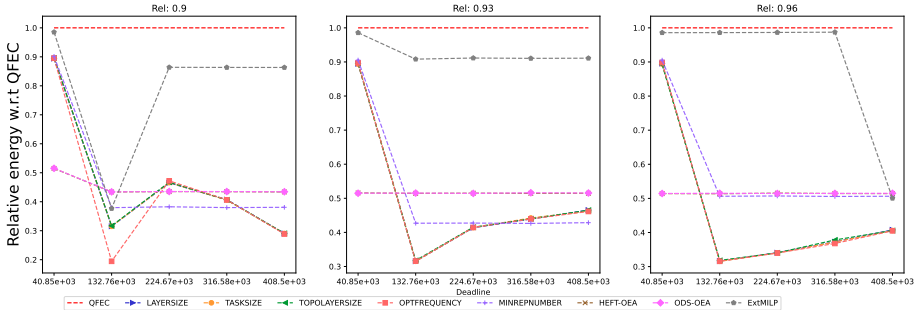

Figure 1117: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

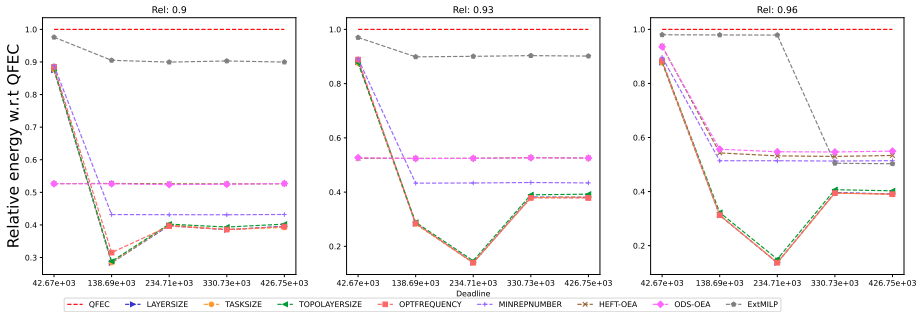

Figure 1118: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

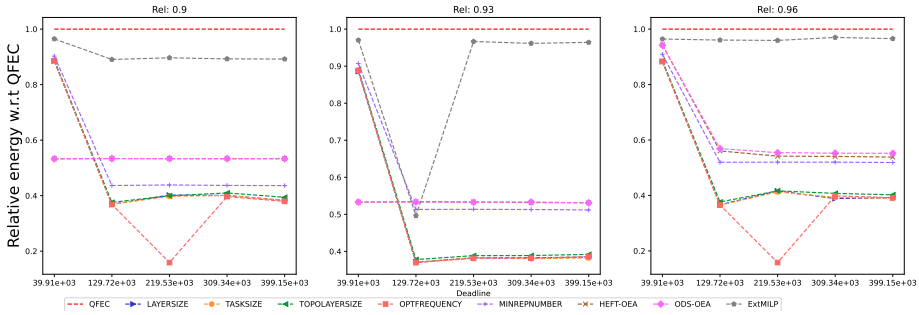

Figure 1119: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

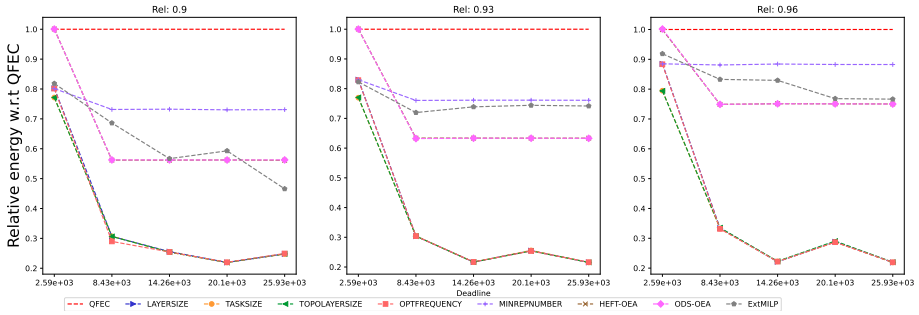

Figure 1120: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

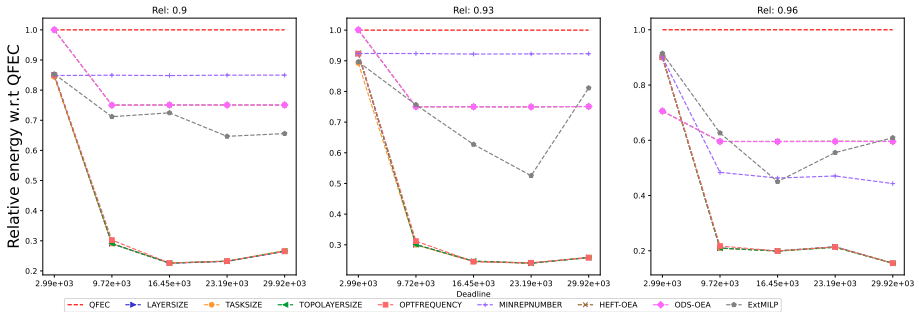

Figure 1121: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.5 $BC/WC = 0.5$

F.5.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

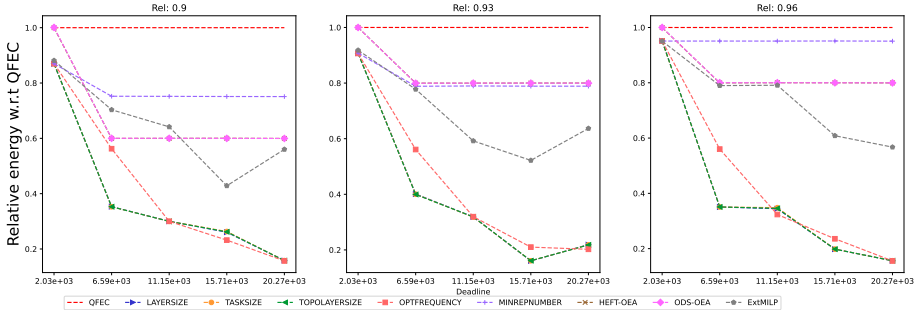

Figure 1122: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

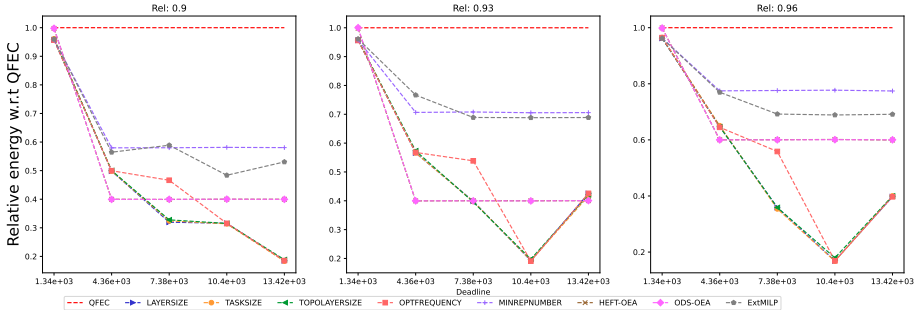

Figure 1123: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

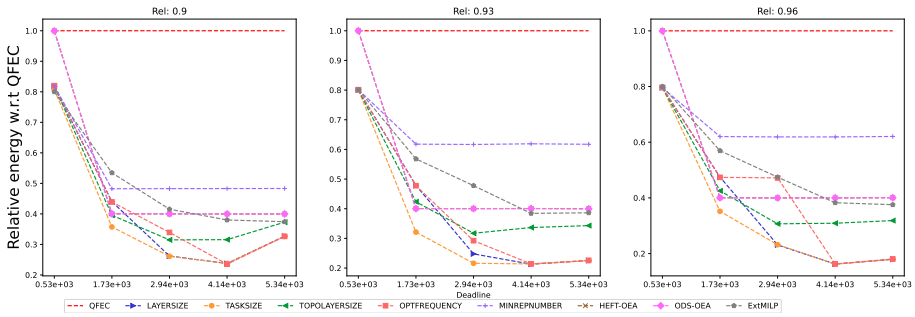

Figure 1124: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

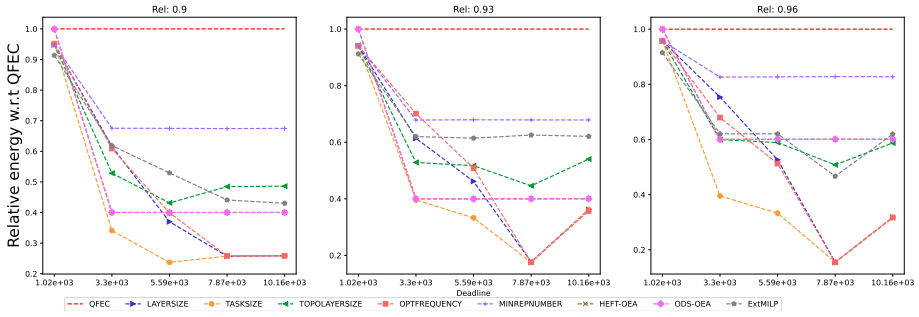

Figure 1125: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

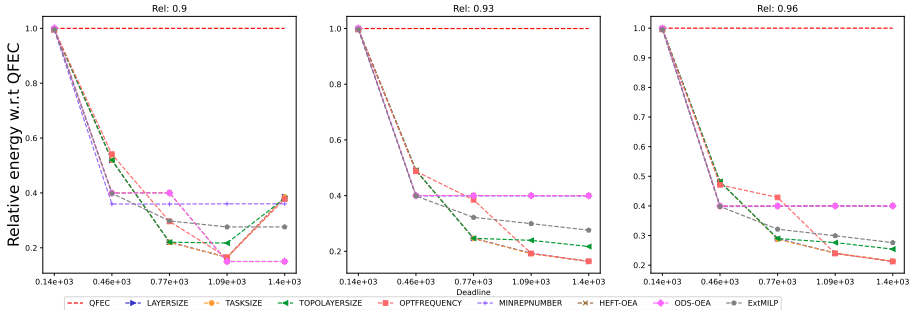

Figure 1126: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

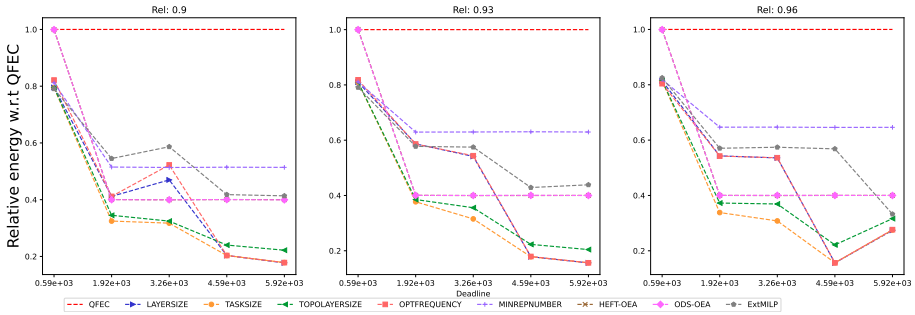

Figure 1127: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

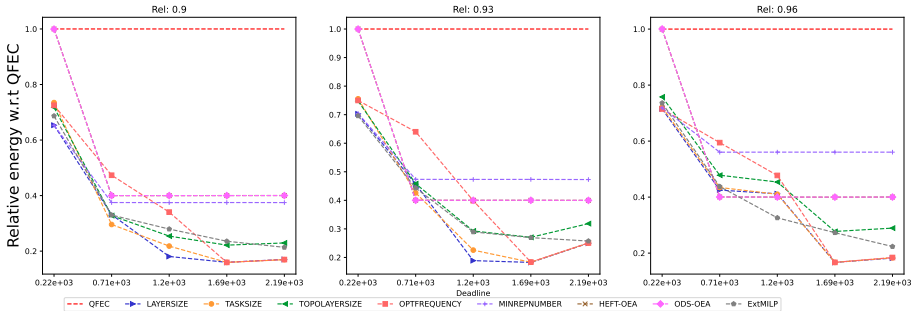

Figure 1128: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

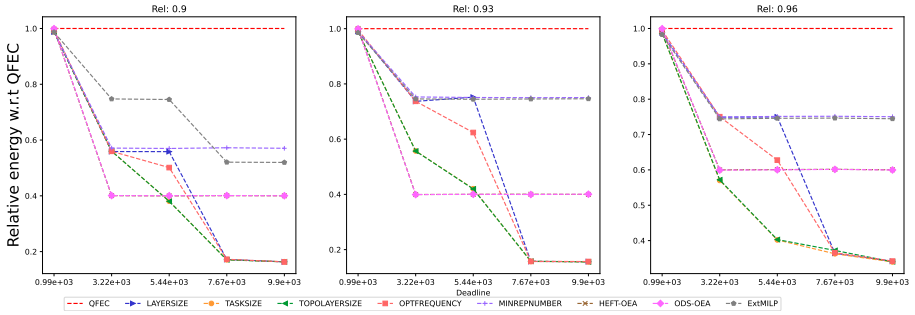

Figure 1129: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

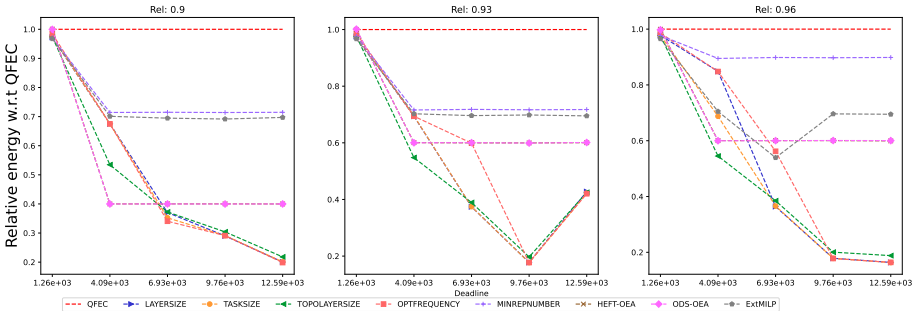

Figure 1130: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

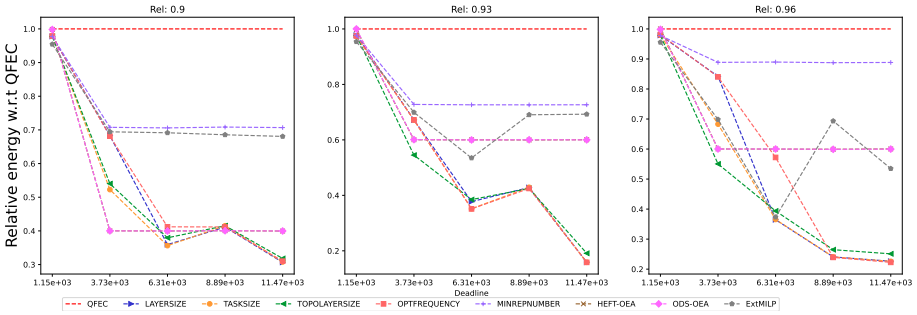

Figure 1131: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

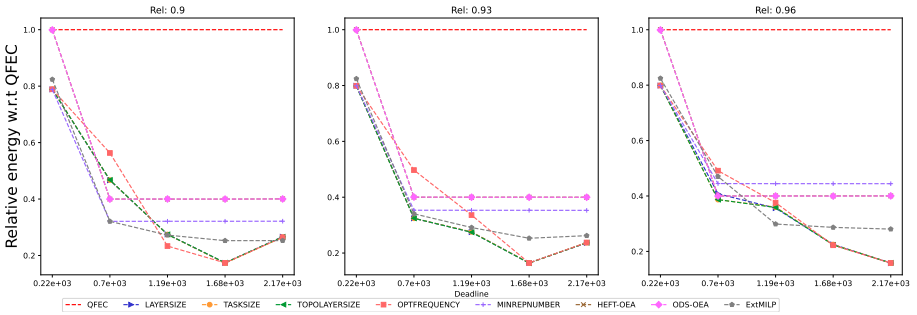

Figure 1132: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

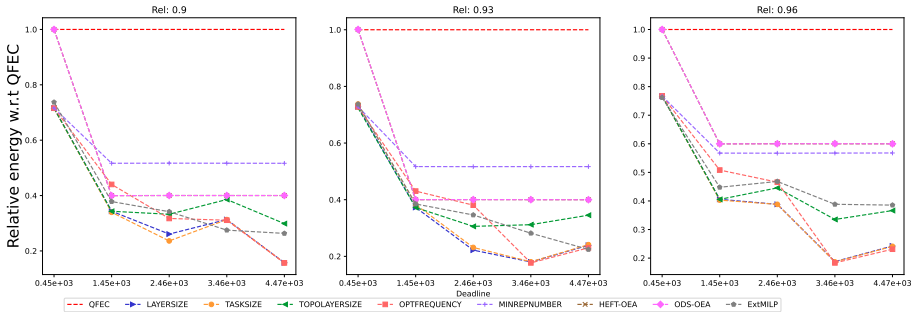

Figure 1133: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

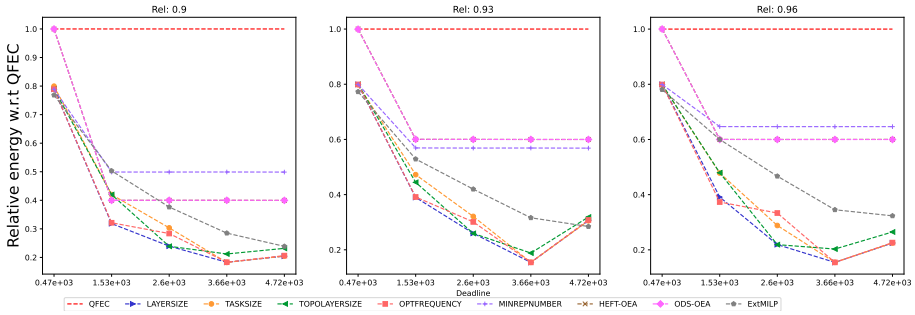

Figure 1134: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

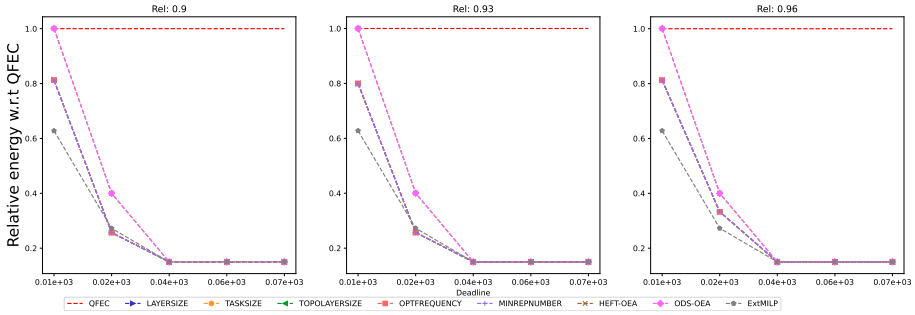

Figure 1135: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

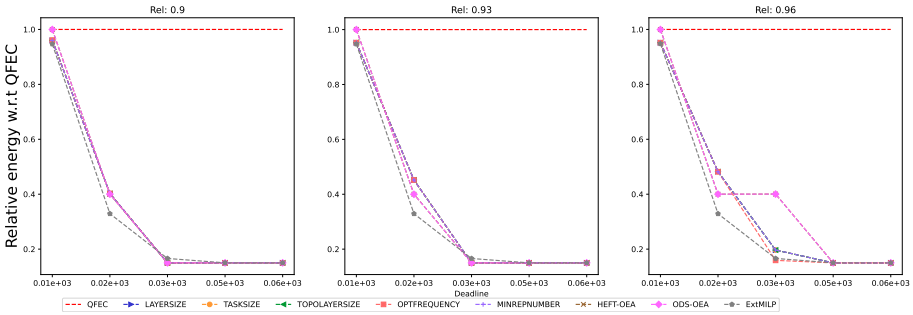

Figure 1136: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

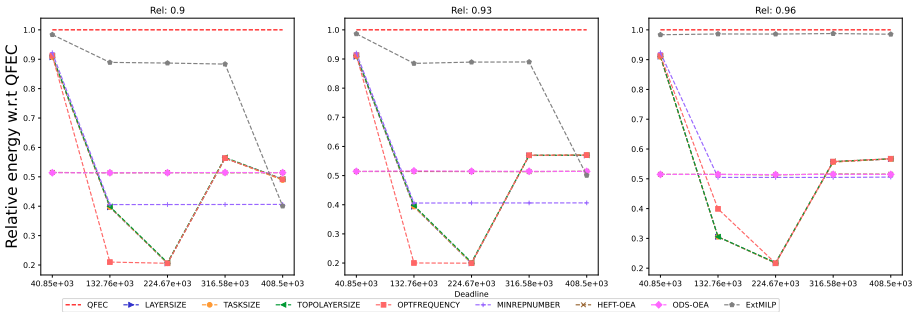

Figure 1137: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

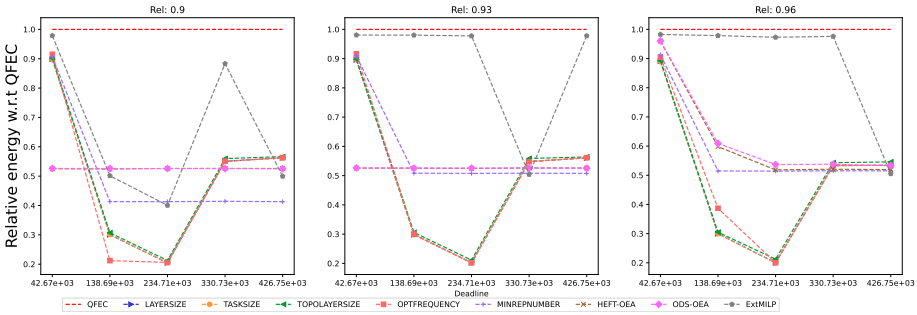

Figure 1138: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

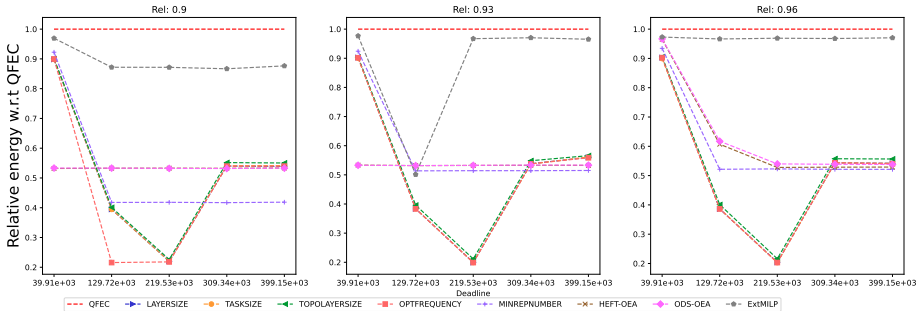

Figure 1139: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

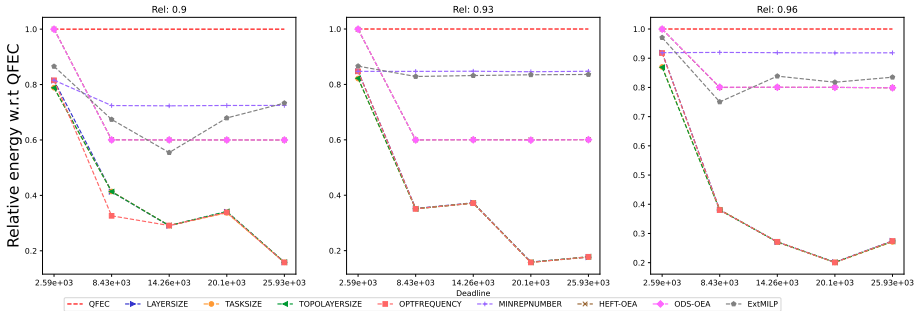

Figure 1140: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

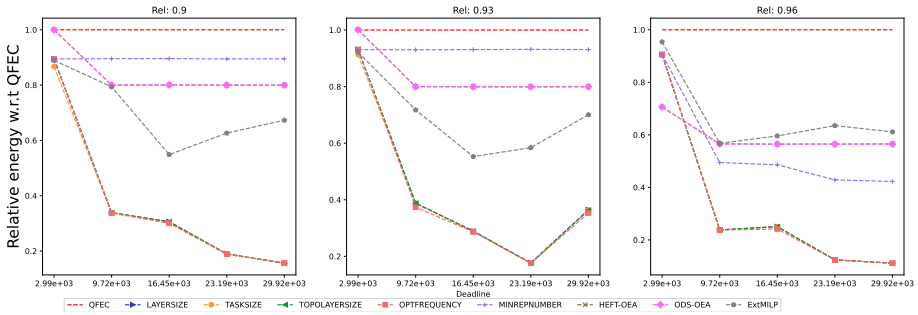

Figure 1141: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.5.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

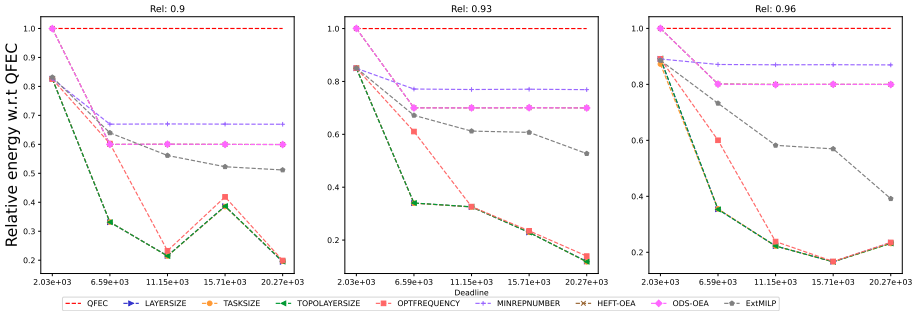

Figure 1142: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

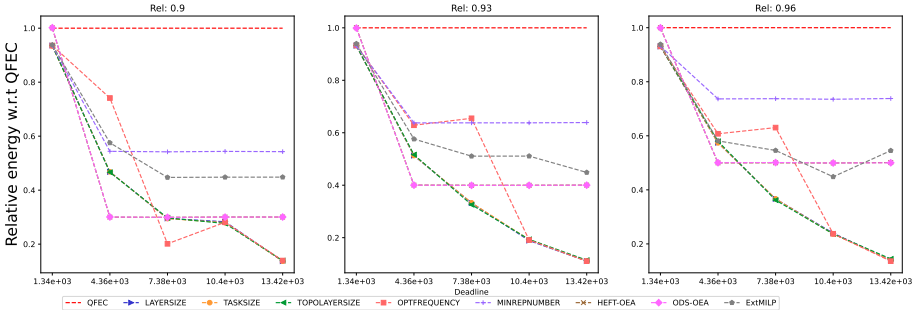

Figure 1143: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

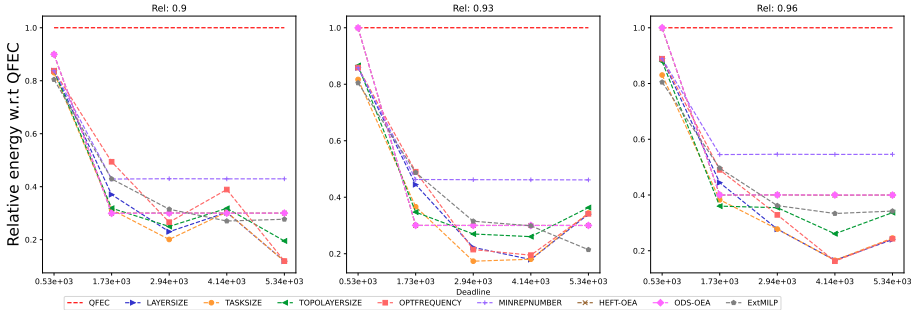

Figure 1144: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

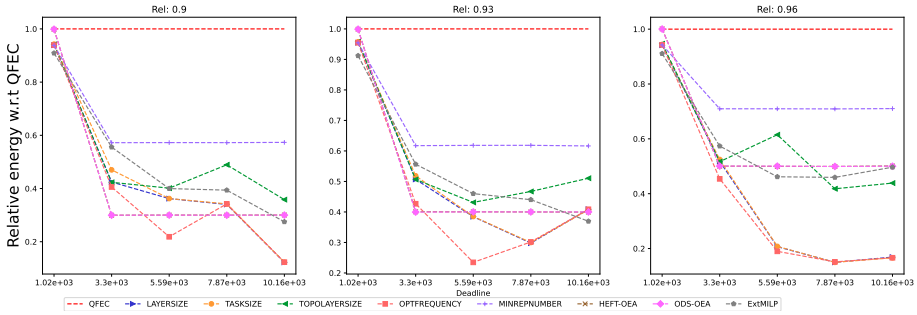

Figure 1145: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

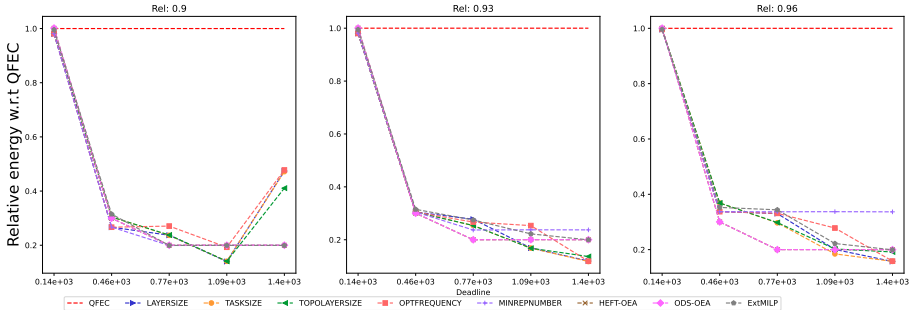

Figure 1146: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

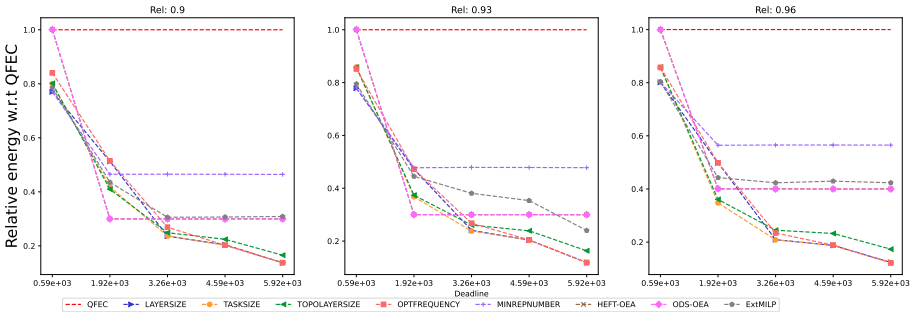

Figure 1147: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

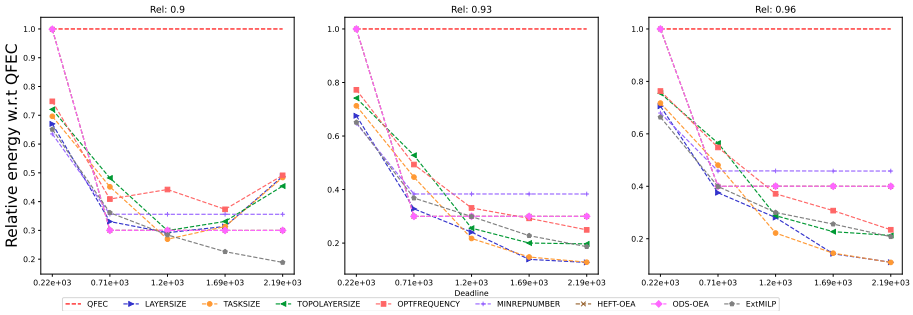

Figure 1148: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

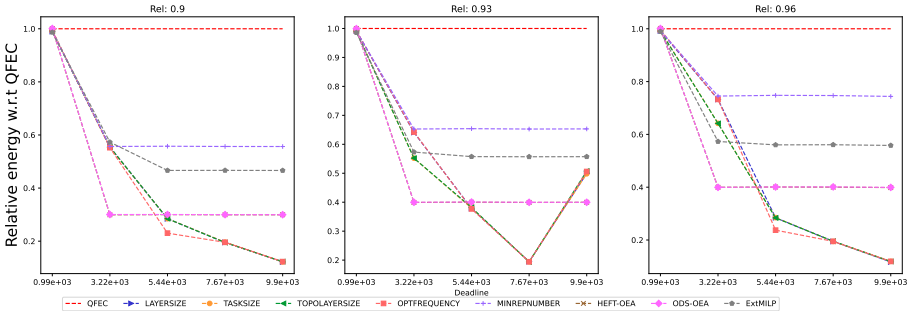

Figure 1149: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

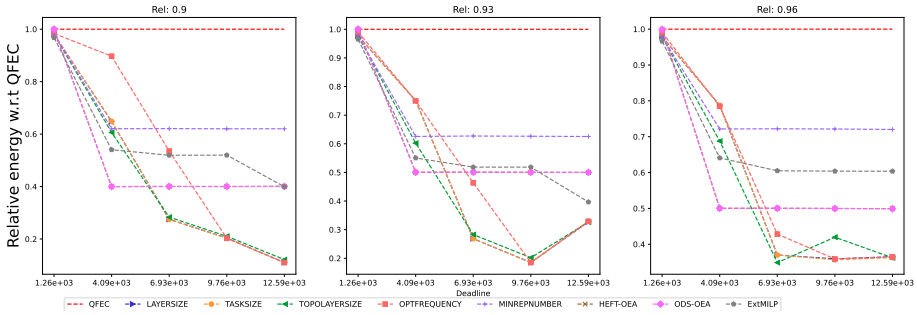

Figure 1150: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

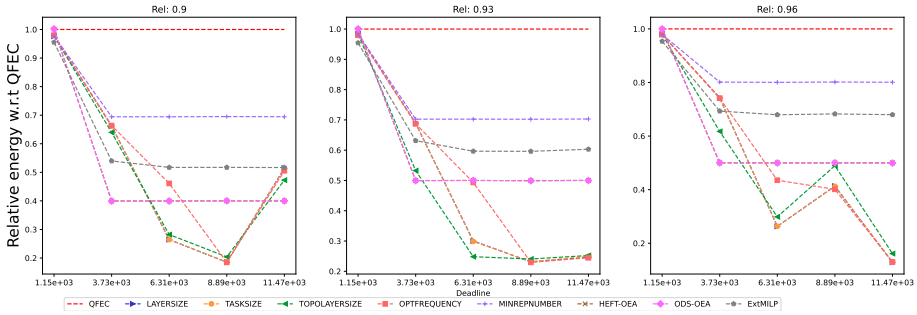

Figure 1151: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

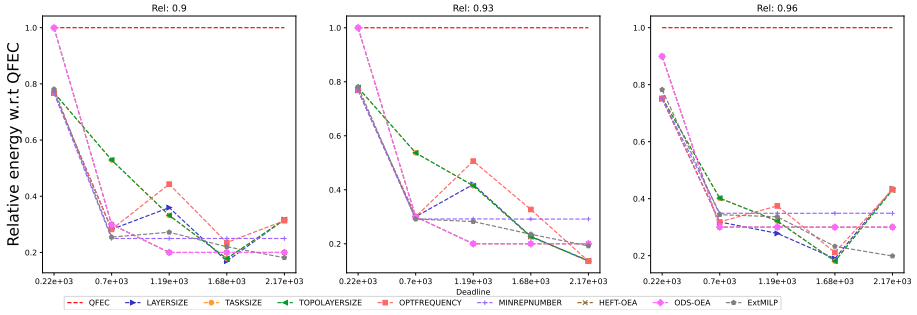

Figure 1152: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

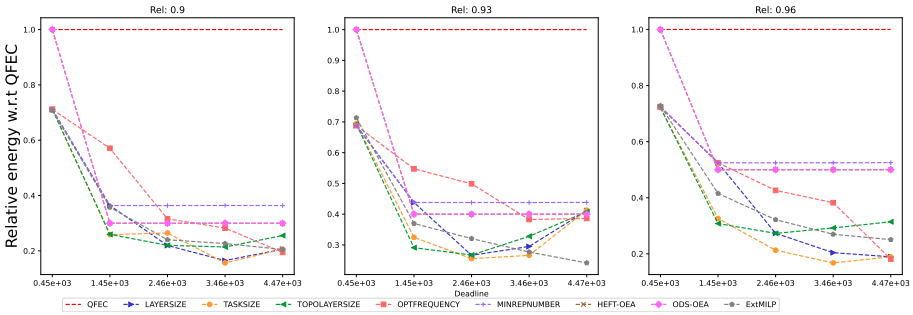

Figure 1153: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

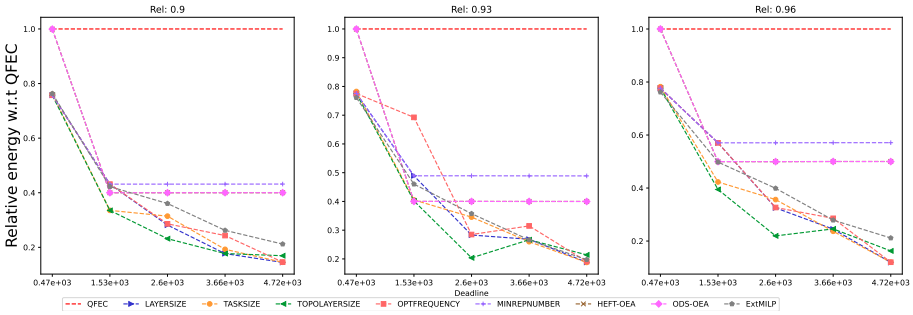

Figure 1154: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

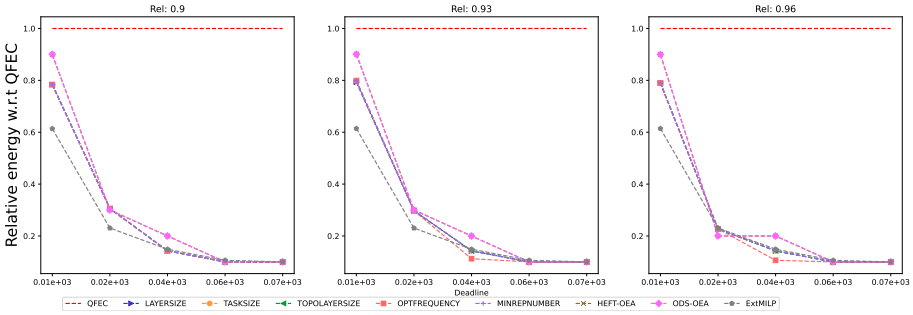

Figure 1155: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

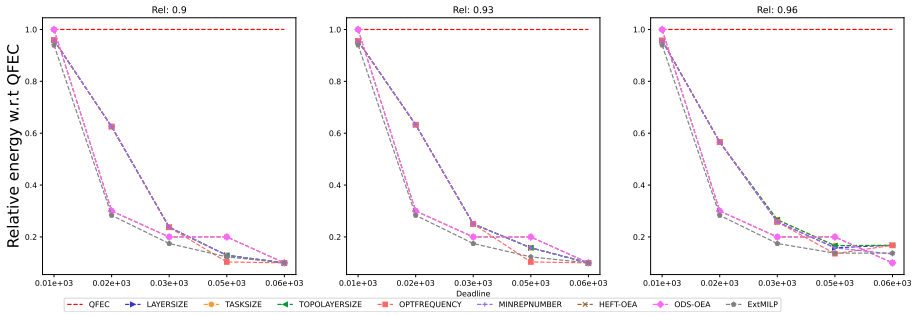

Figure 1156: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

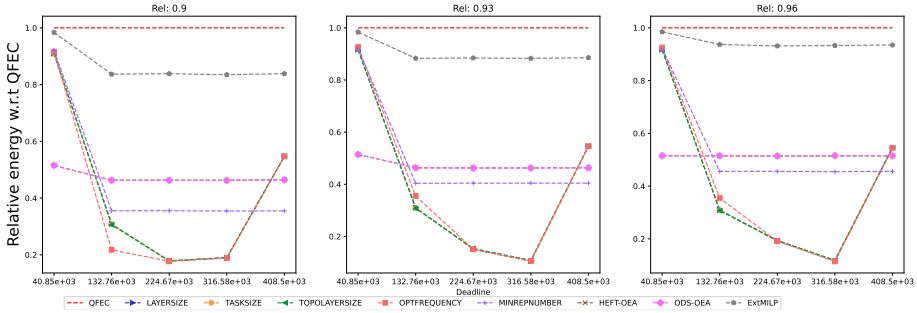

Figure 1157: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

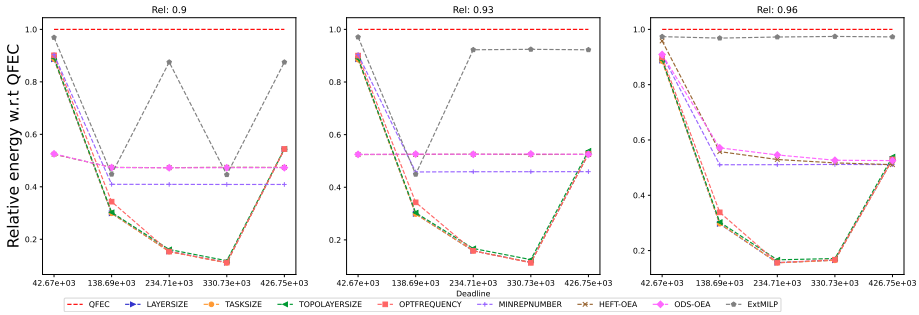

Figure 1158: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

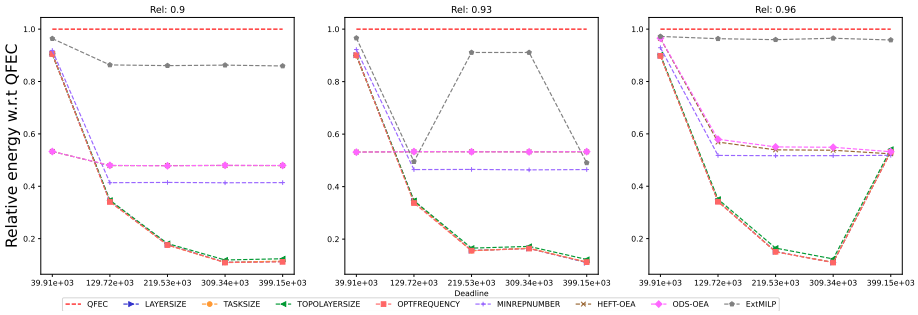

Figure 1159: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

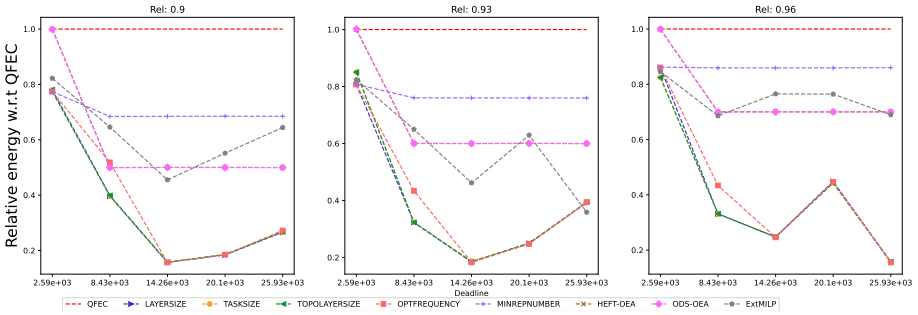

Figure 1160: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

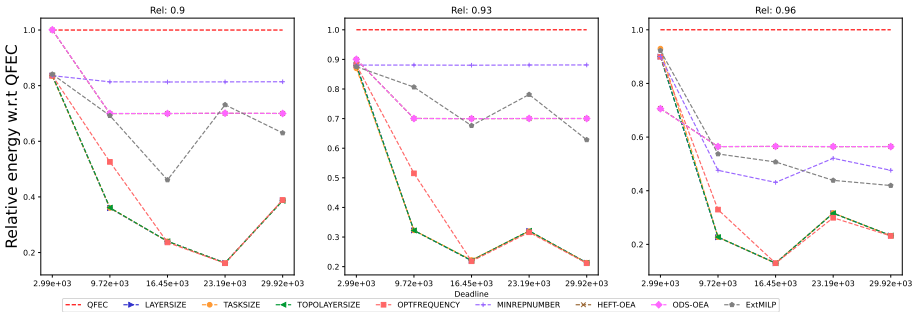

Figure 1161: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.5.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

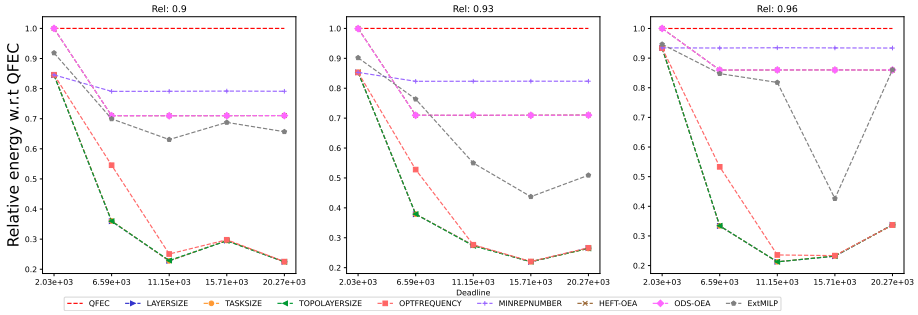

Figure 1162: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

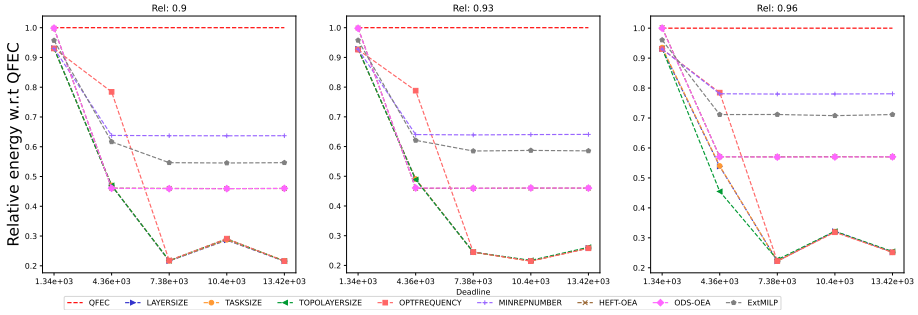

Figure 1163: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

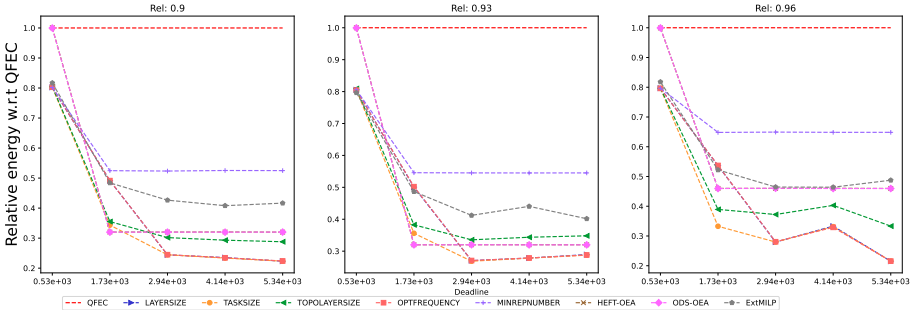

Figure 1164: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

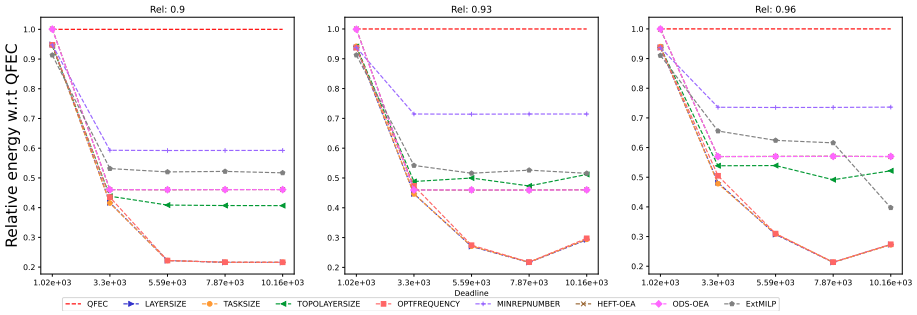

Figure 1165: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

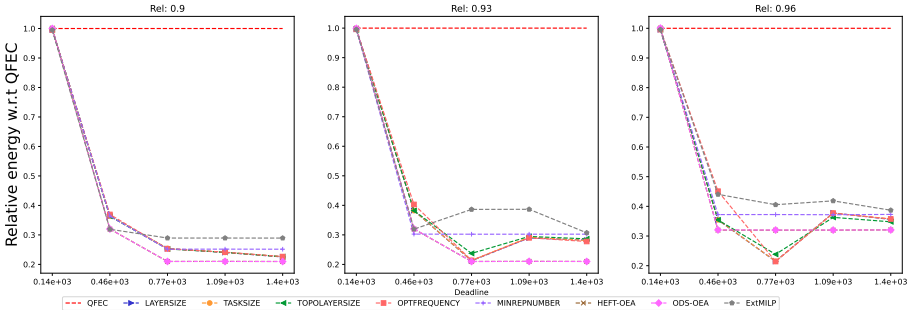

Figure 1166: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

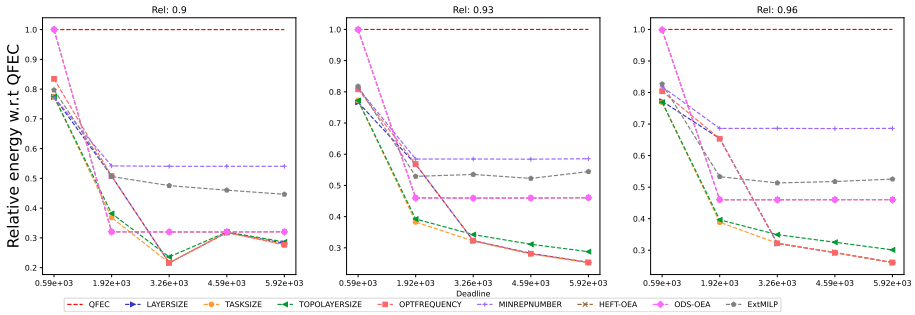

Figure 1167: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

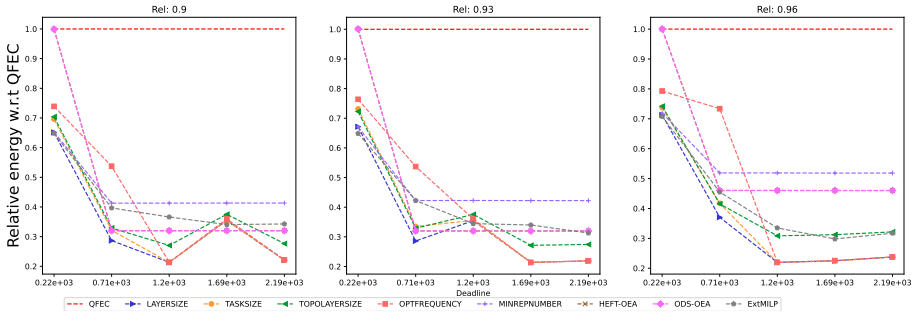

Figure 1168: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

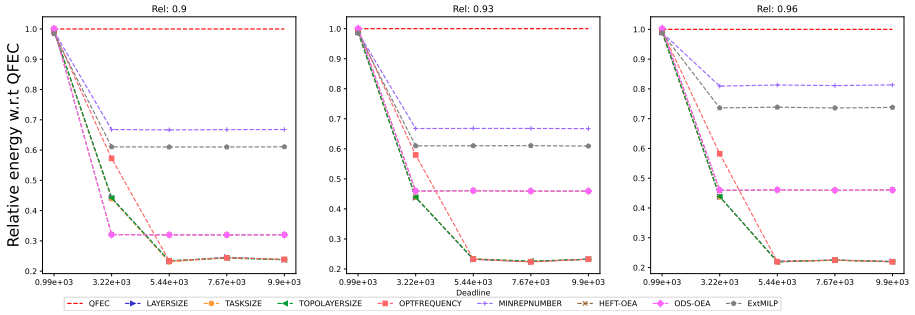

Figure 1169: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

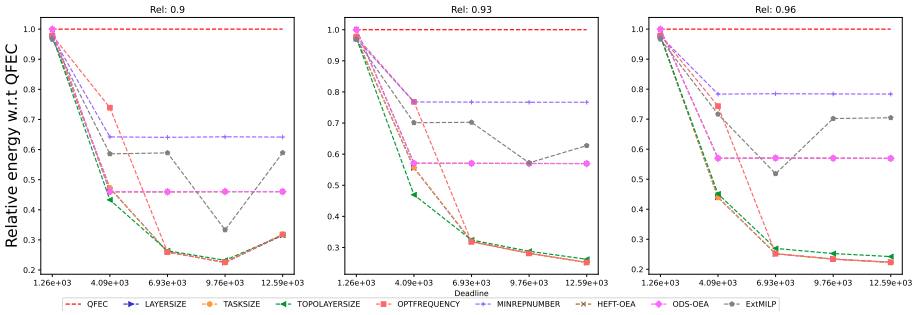

Figure 1170: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

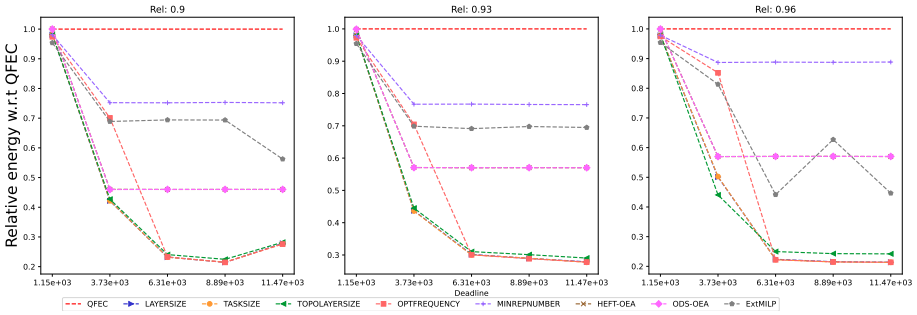

Figure 1171: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

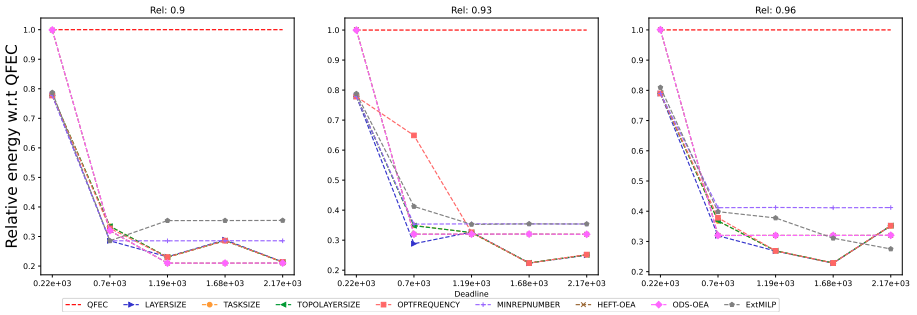

Figure 1172: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

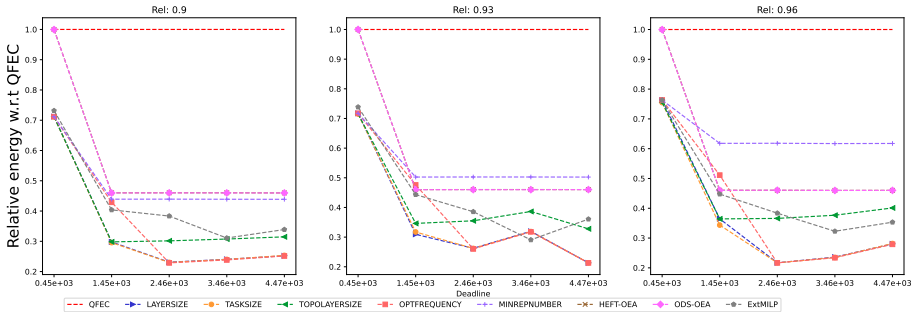

Figure 1173: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

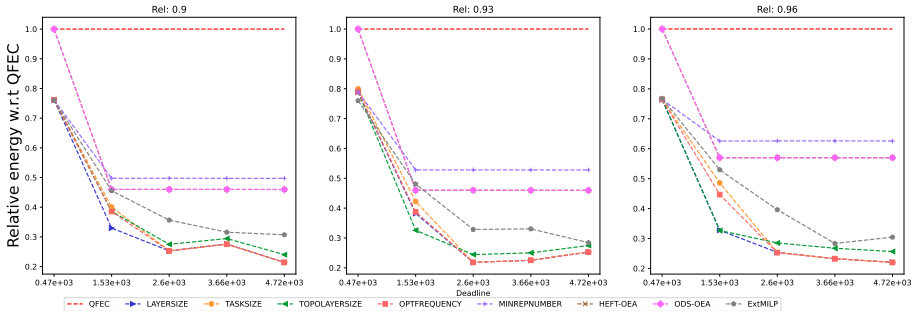

Figure 1174: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

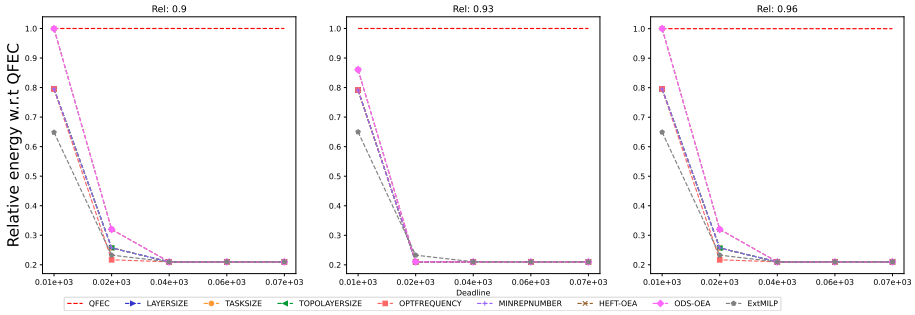

Figure 1175: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

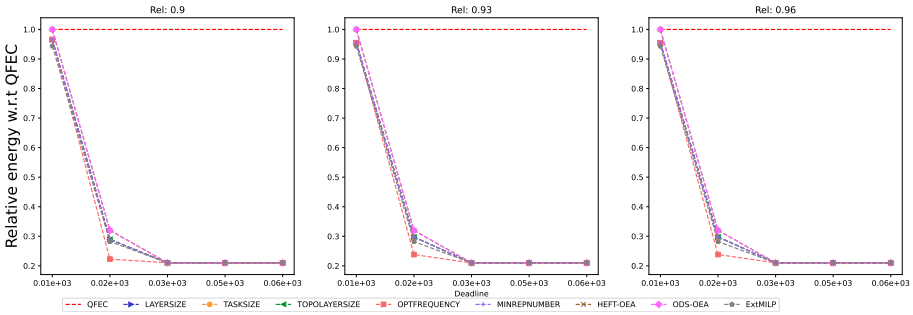

Figure 1176: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

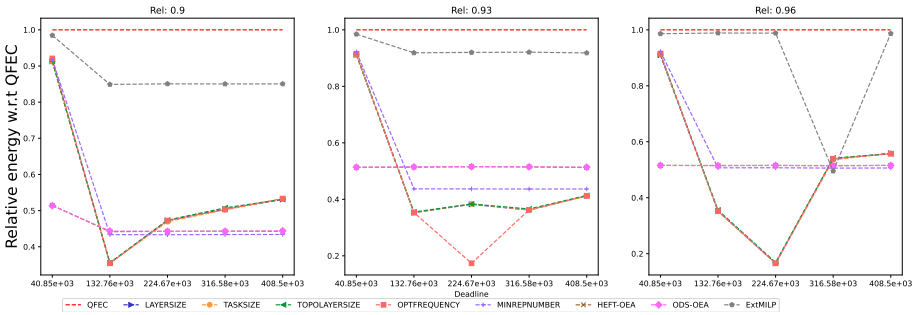

Figure 1177: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

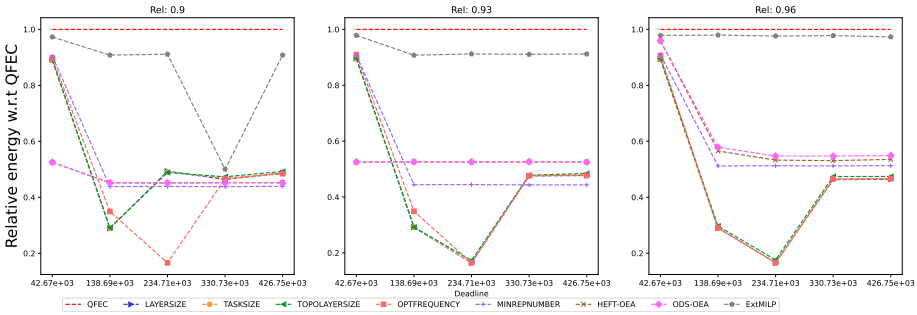

Figure 1178: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

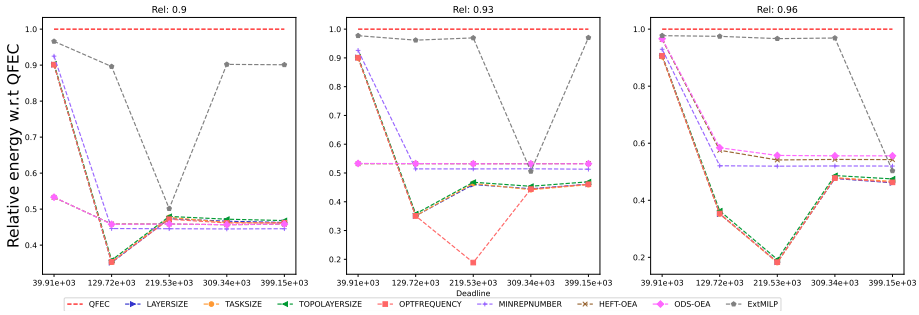

Figure 1179: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

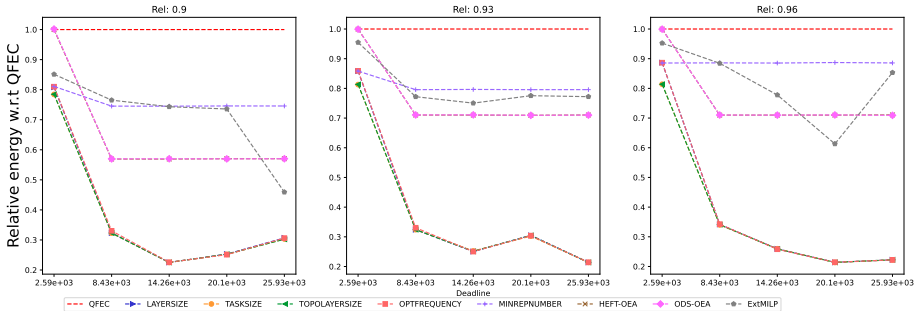

Figure 1180: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

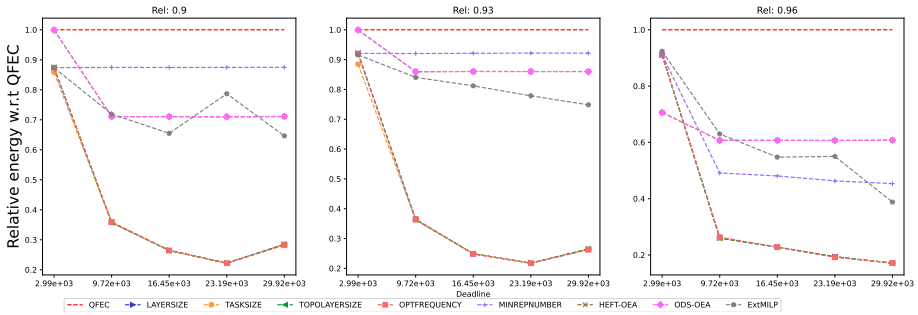

Figure 1181: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.5.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

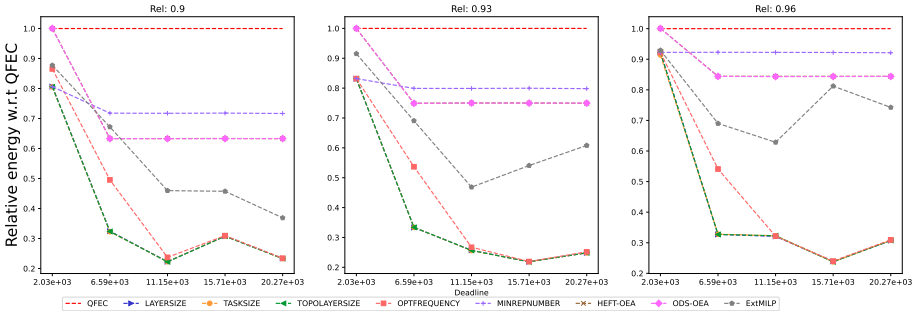

Figure 1182: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

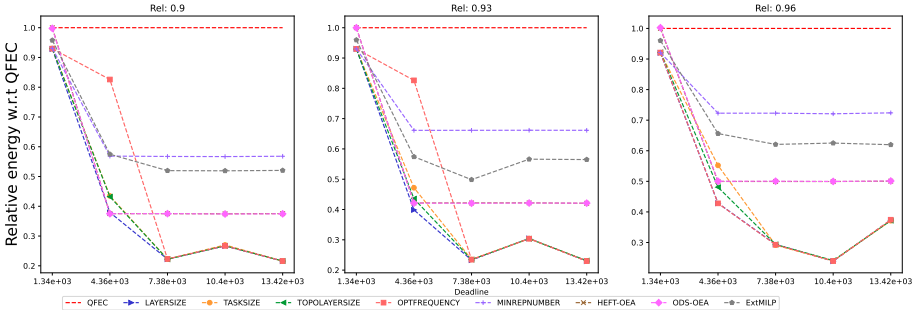

Figure 1183: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

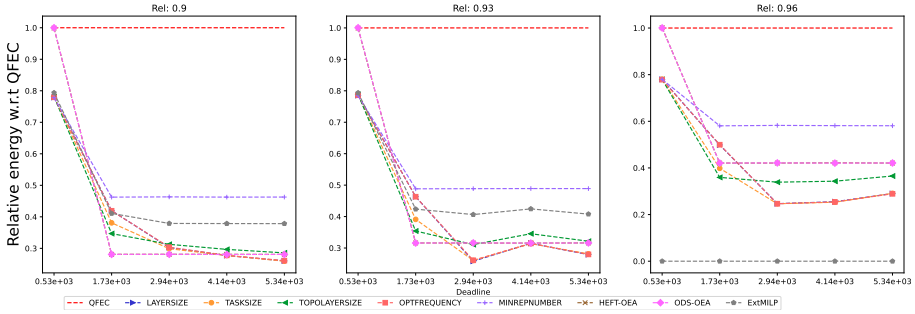

Figure 1184: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

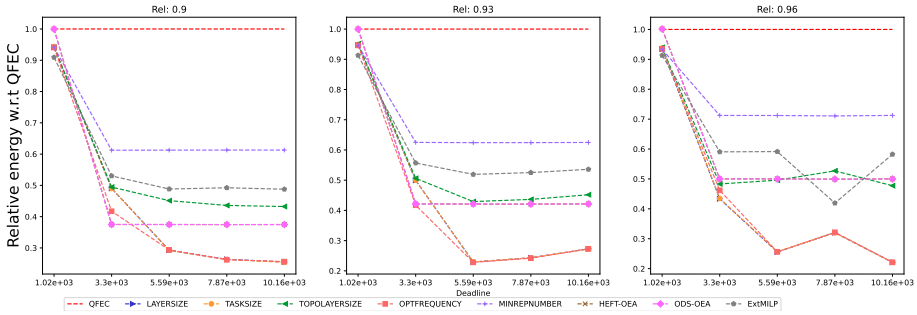

Figure 1185: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

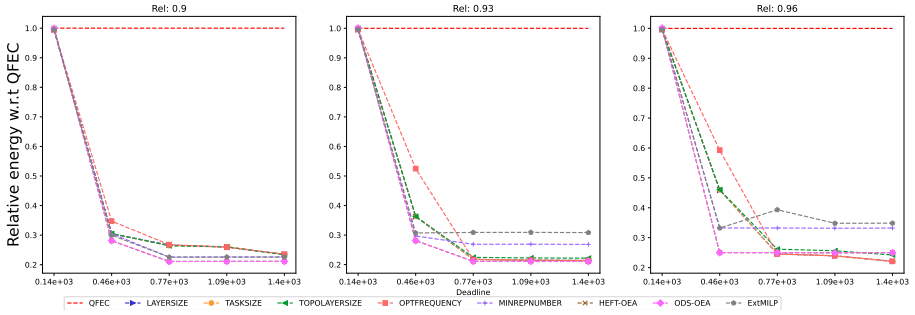

Figure 1186: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

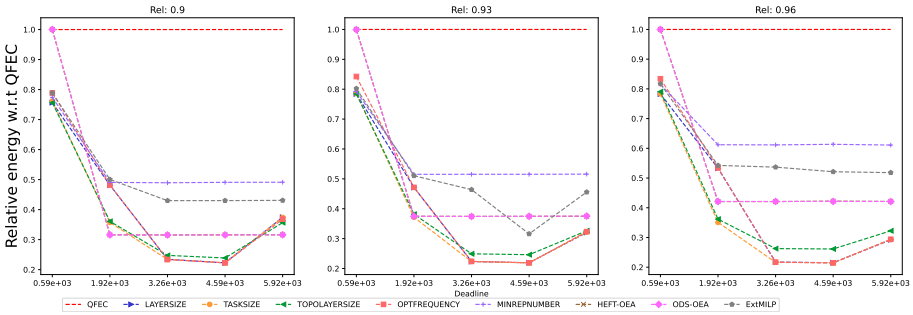

Figure 1187: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

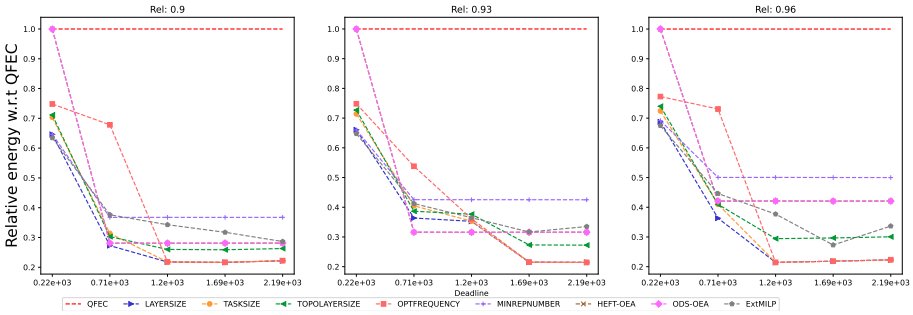

Figure 1188: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

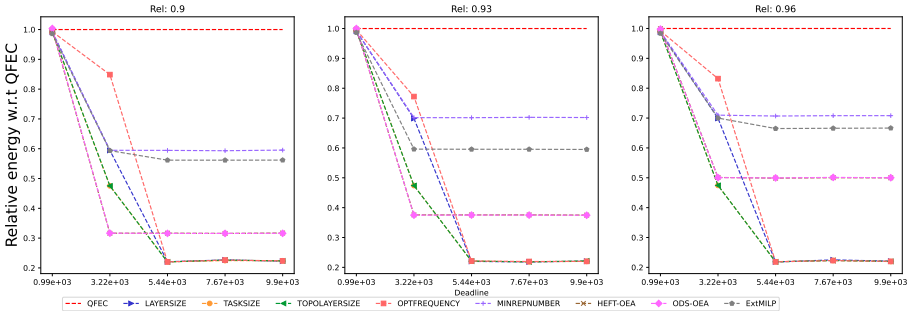

Figure 1189: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

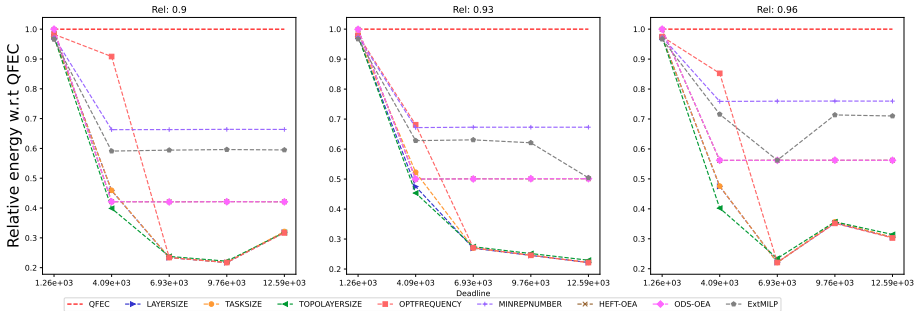

Figure 1190: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

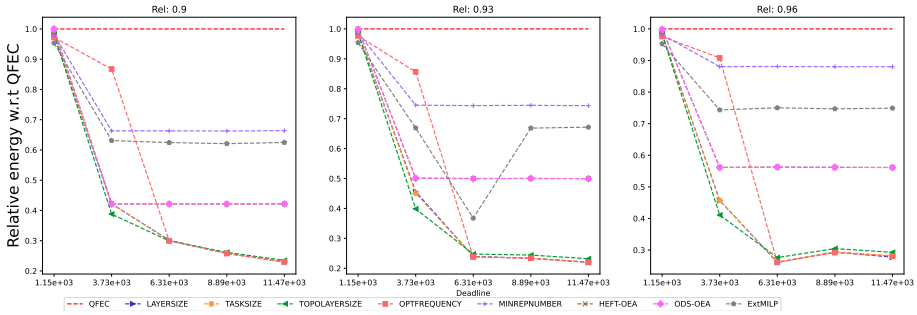

Figure 1191: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

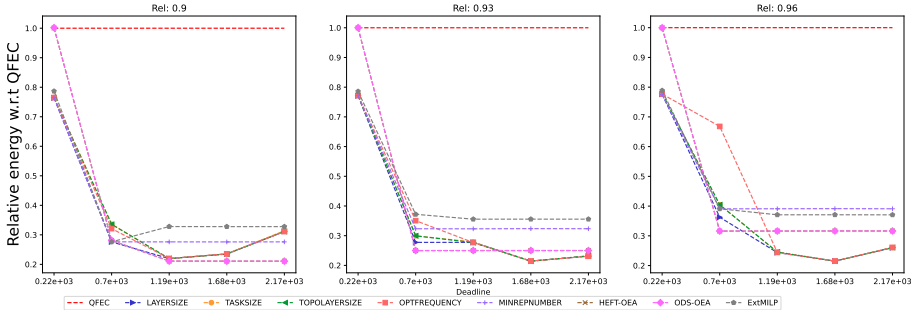

Figure 1192: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

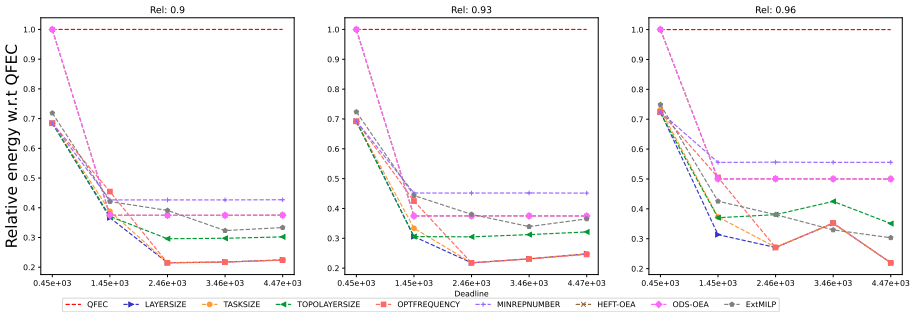

Figure 1193: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

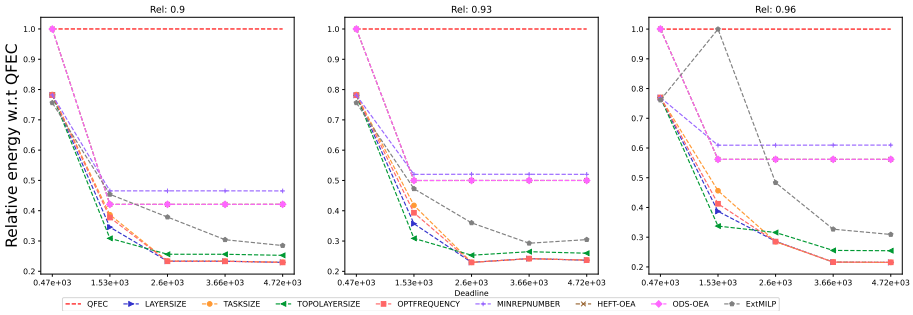

Figure 1194: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

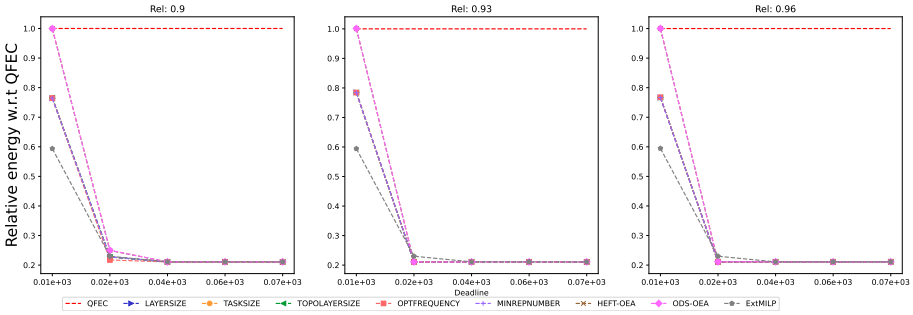

Figure 1195: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

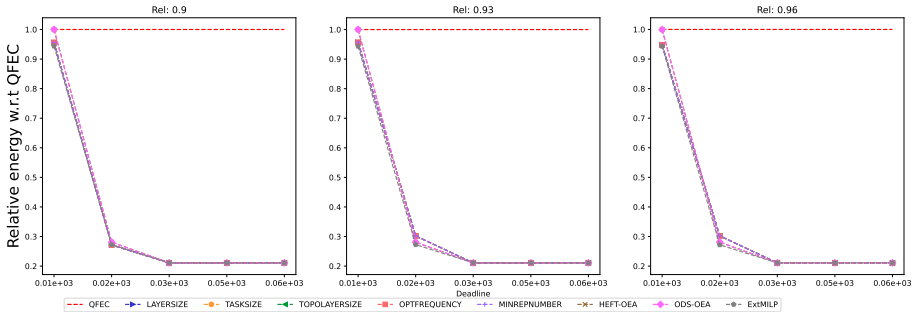

Figure 1196: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

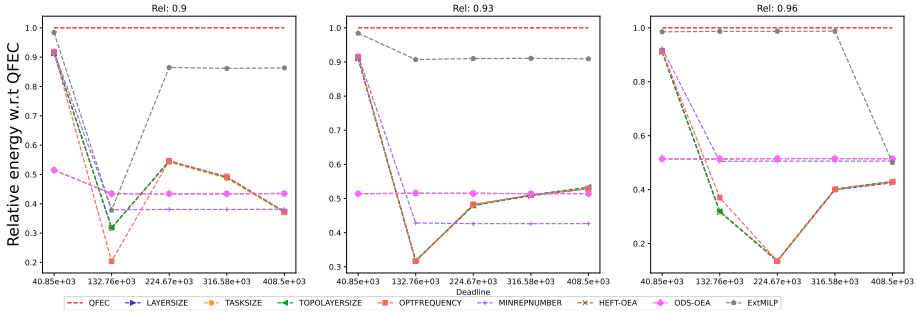

Figure 1197: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

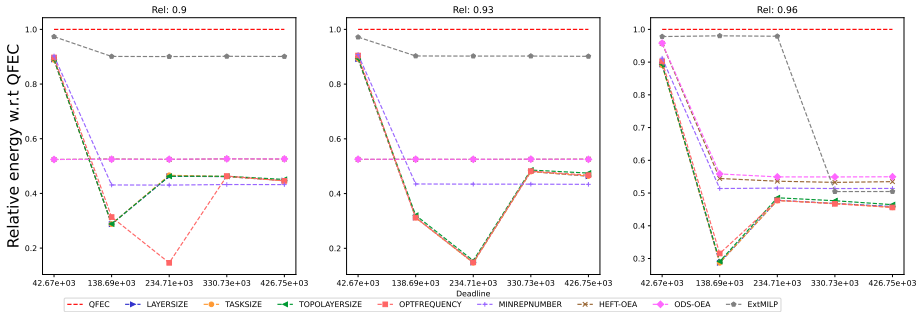

Figure 1198: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

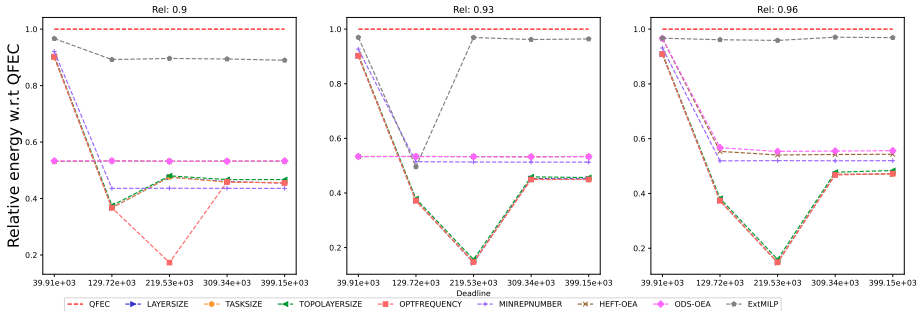

Figure 1199: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

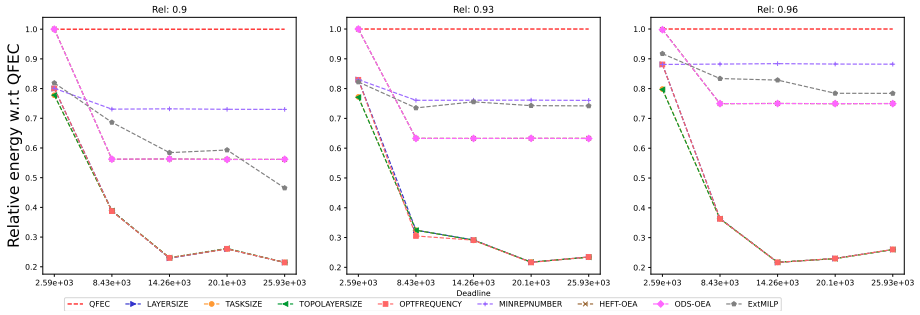

Figure 1200: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

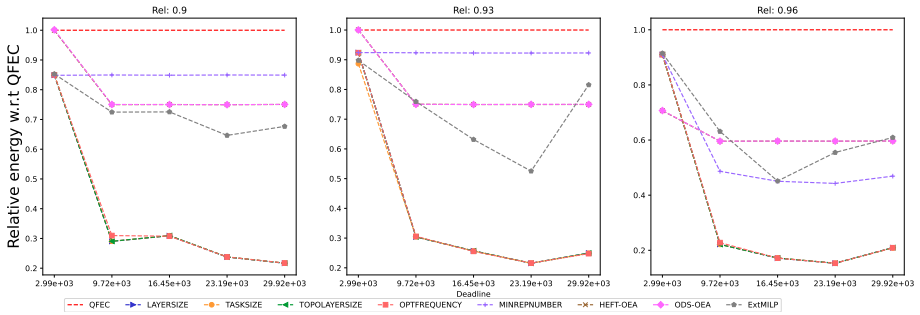

Figure 1201: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.6 $BC/WC = 0.6$

F.6.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

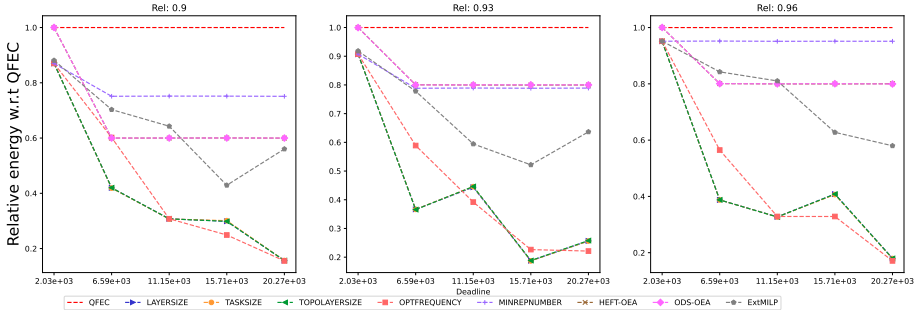

Figure 1202: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

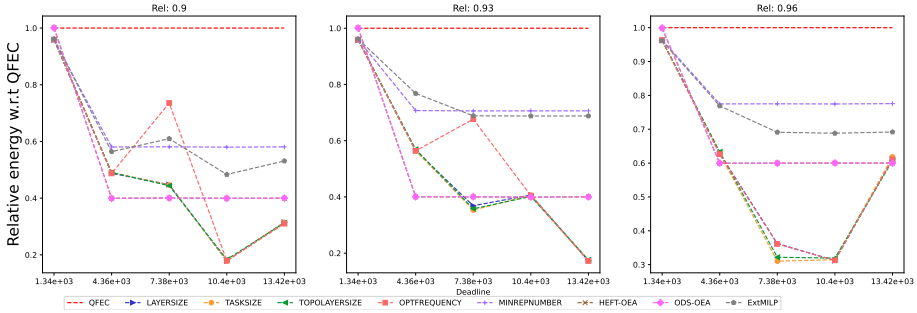

Figure 1203: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

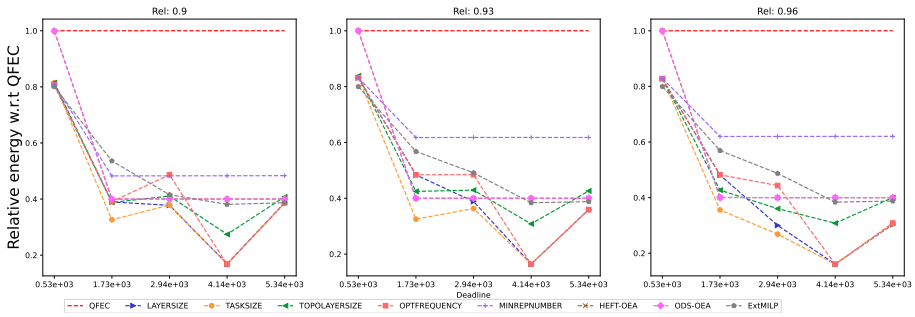

Figure 1204: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

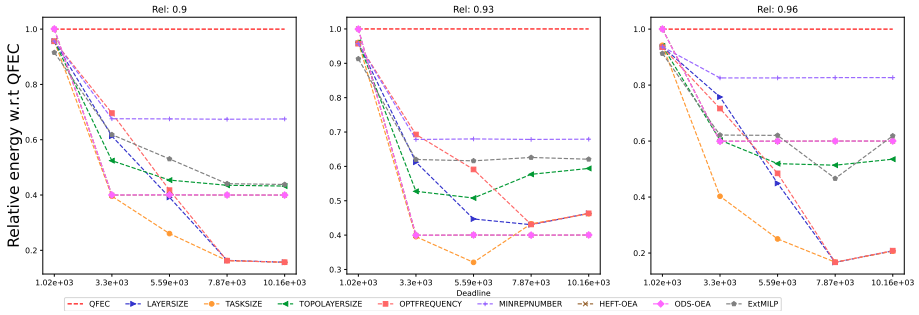

Figure 1205: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

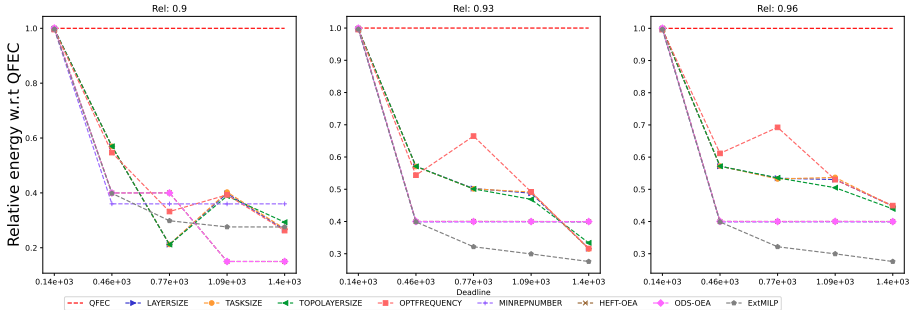

Figure 1206: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

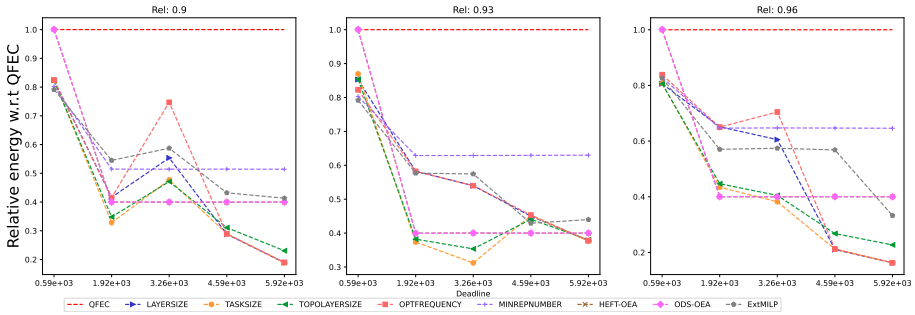

Figure 1207: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

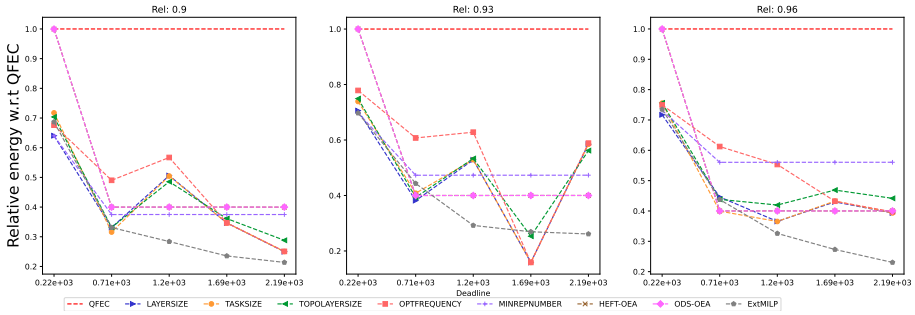

Figure 1208: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

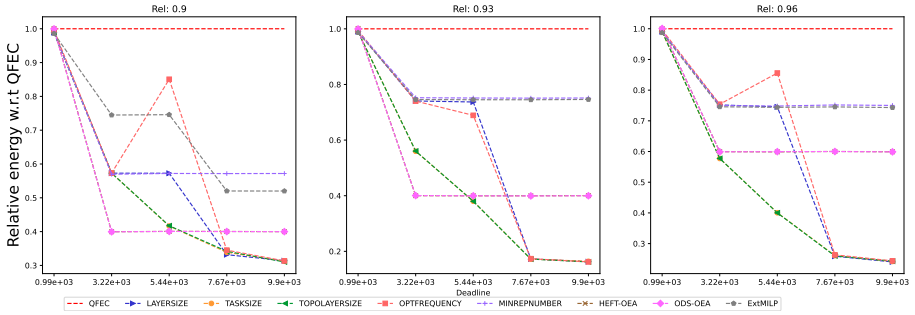

Figure 1209: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

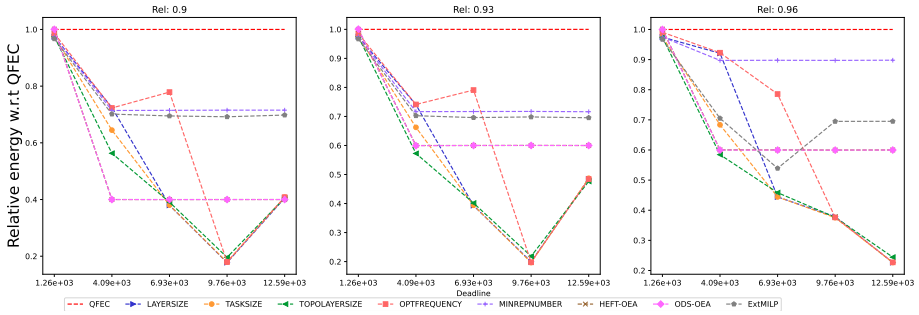

Figure 1210: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

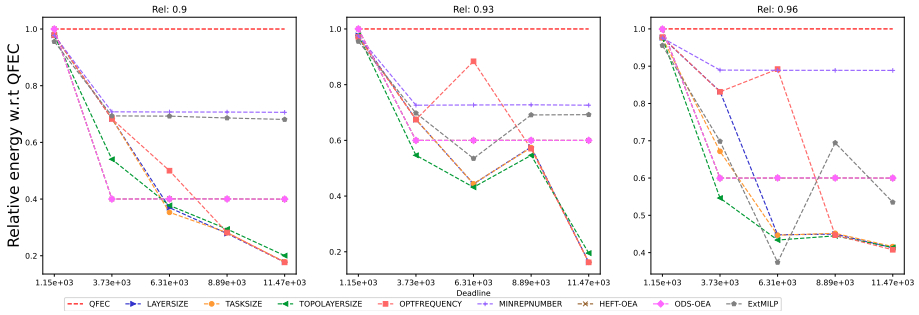

Figure 1211: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

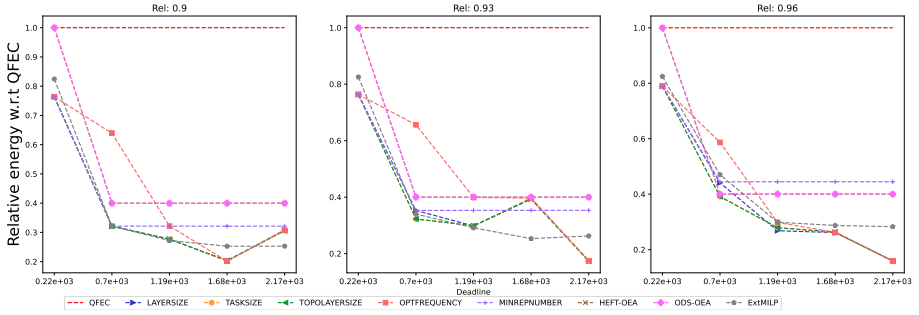

Figure 1212: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

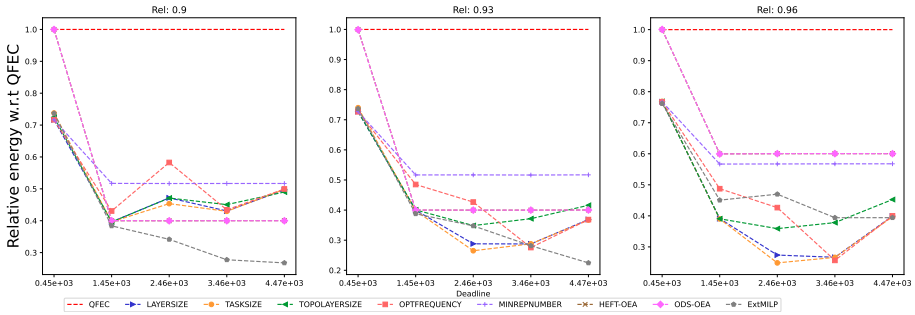

Figure 1213: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

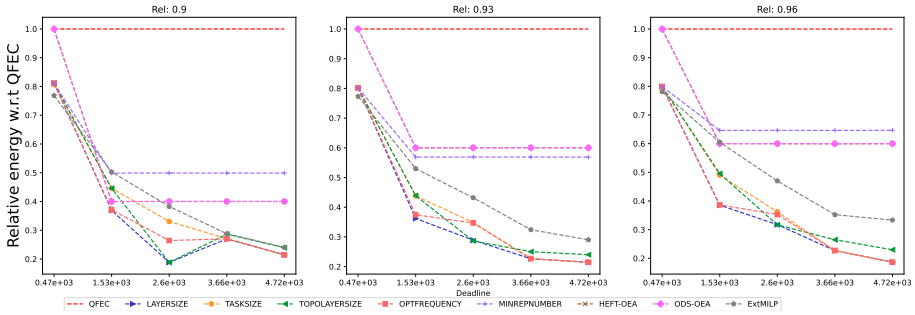

Figure 1214: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

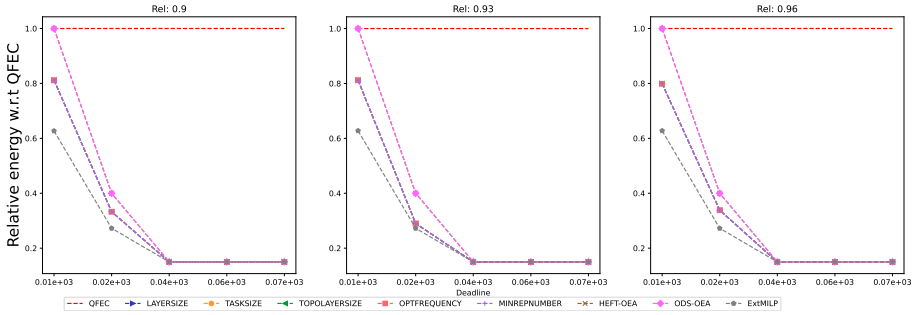

Figure 1215: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

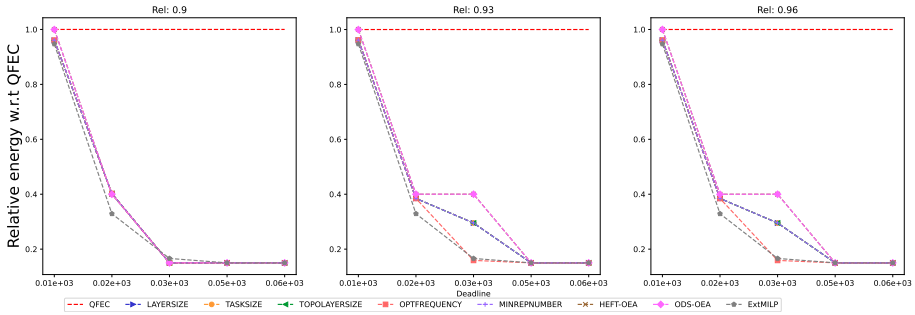

Figure 1216: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

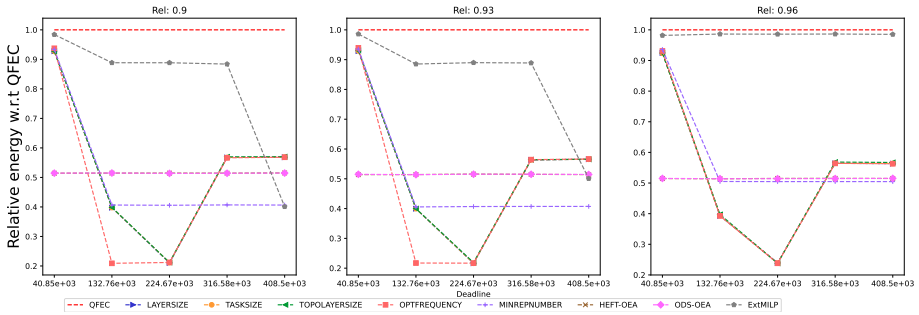

Figure 1217: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

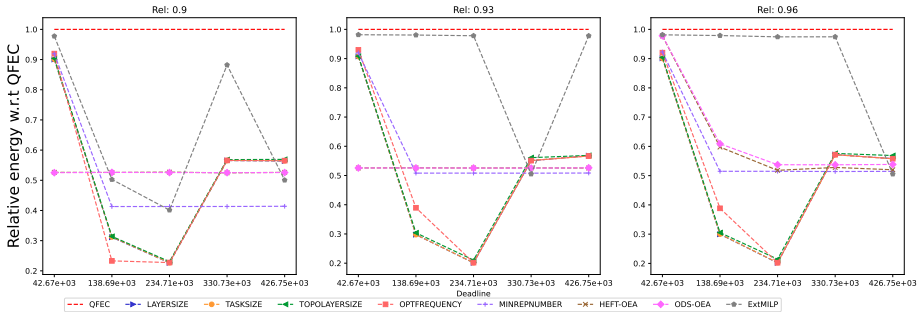

Figure 1218: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

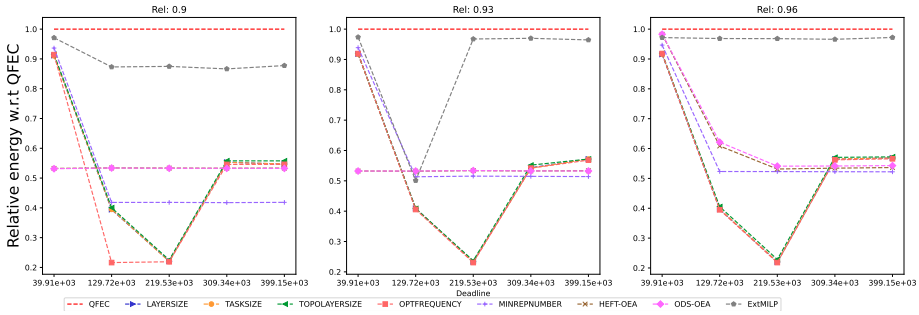

Figure 1219: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

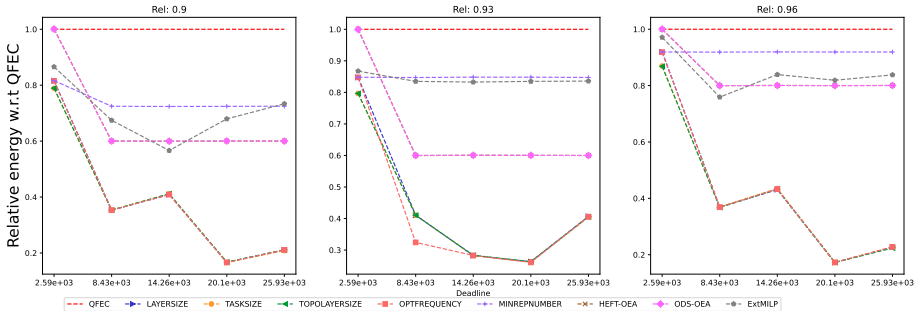

Figure 1220: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

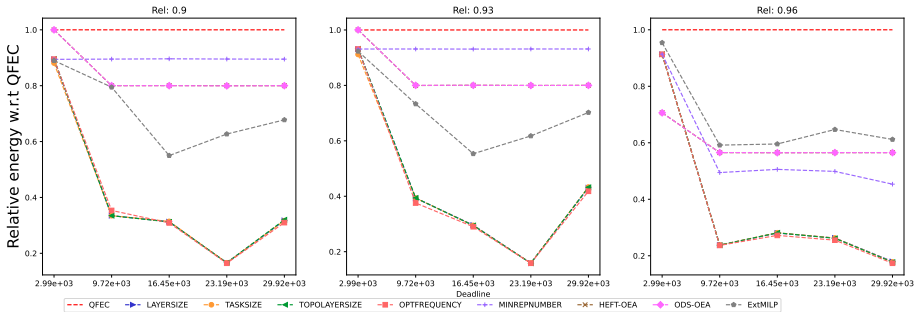

Figure 1221: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.6.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

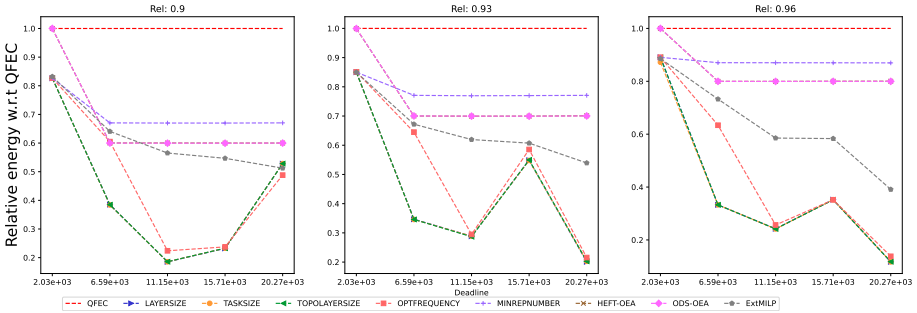

Figure 1222: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

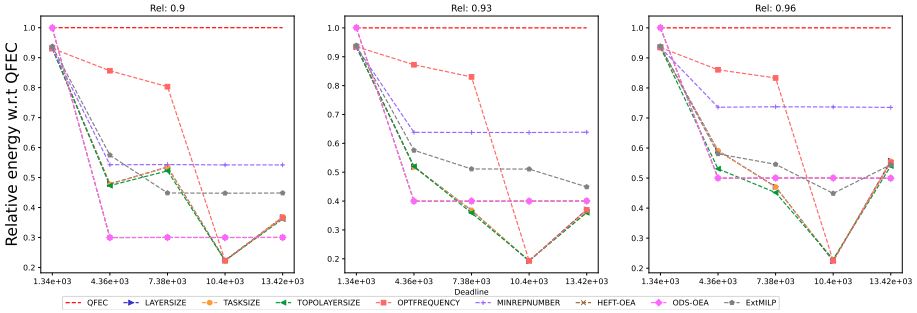

Figure 1223: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

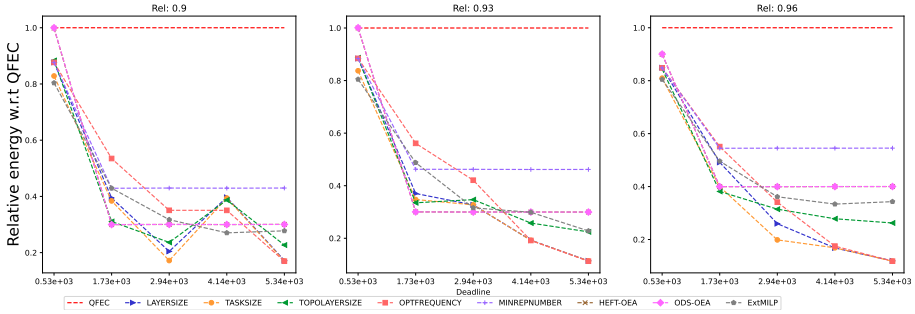

Figure 1224: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

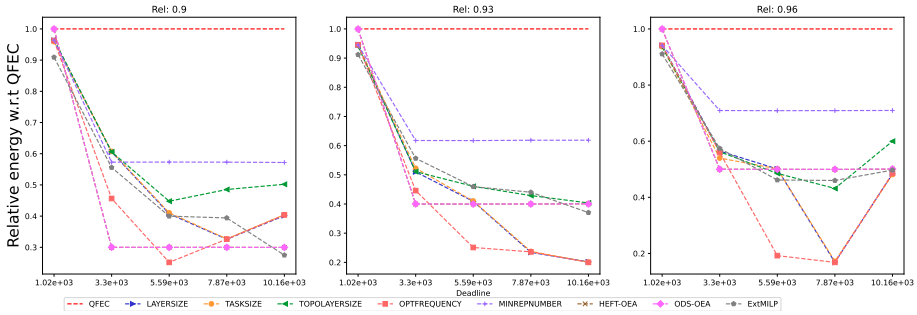

Figure 1225: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

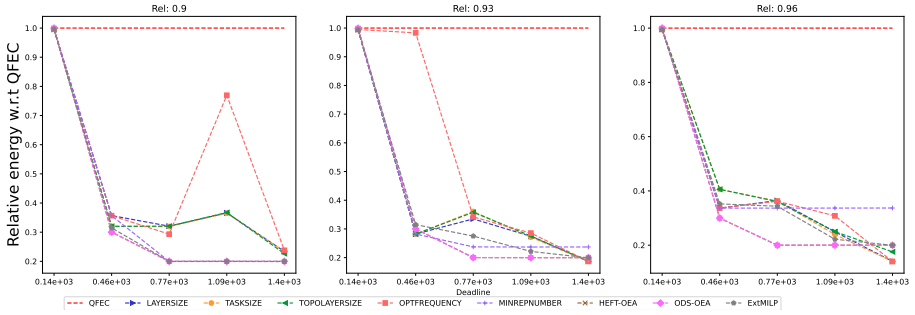

Figure 1226: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

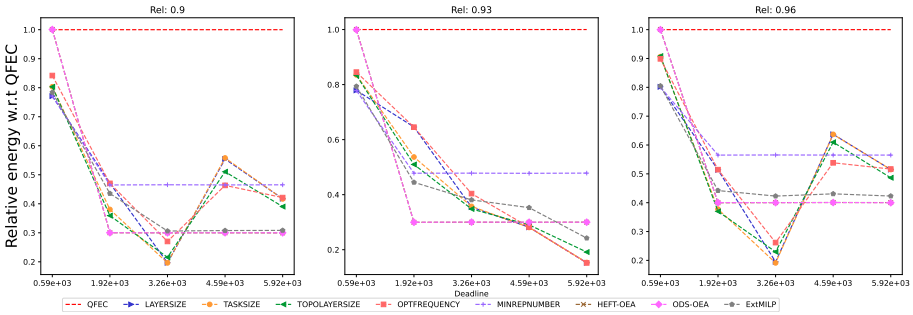

Figure 1227: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

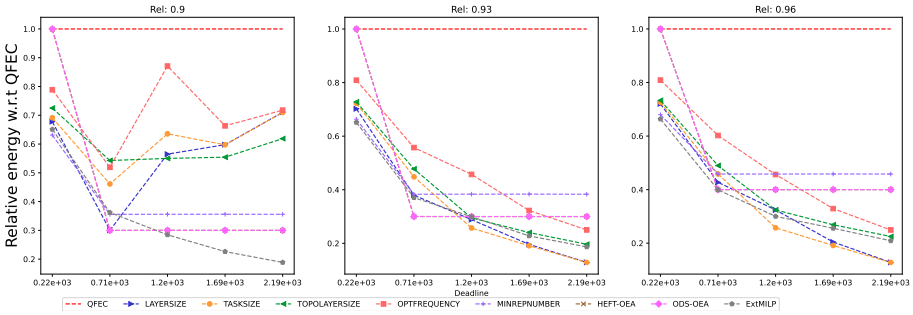

Figure 1228: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

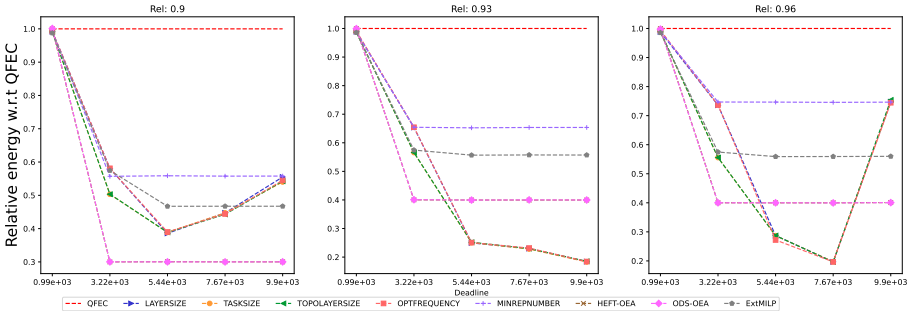

Figure 1229: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

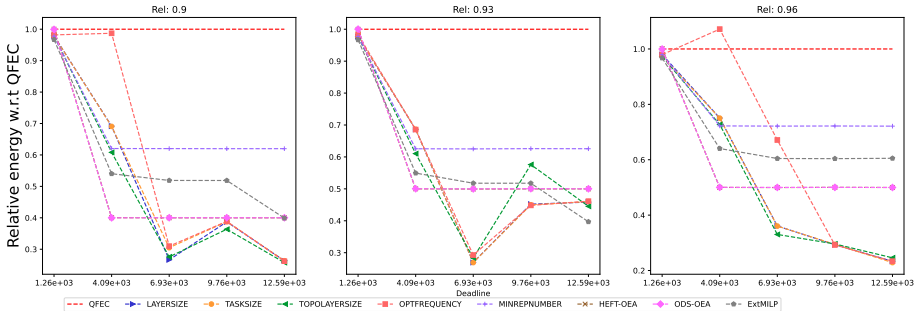

Figure 1230: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

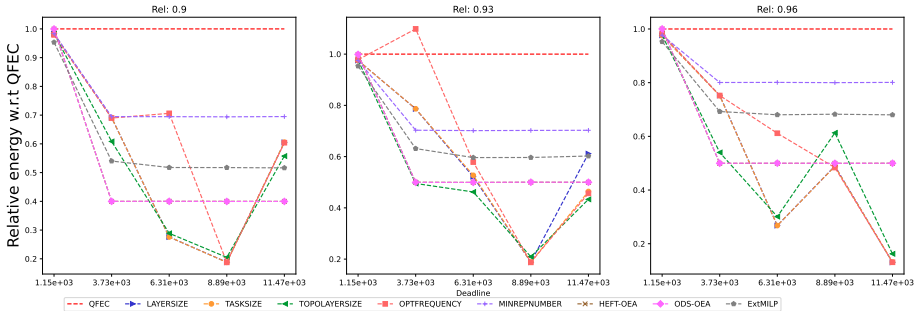

Figure 1231: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

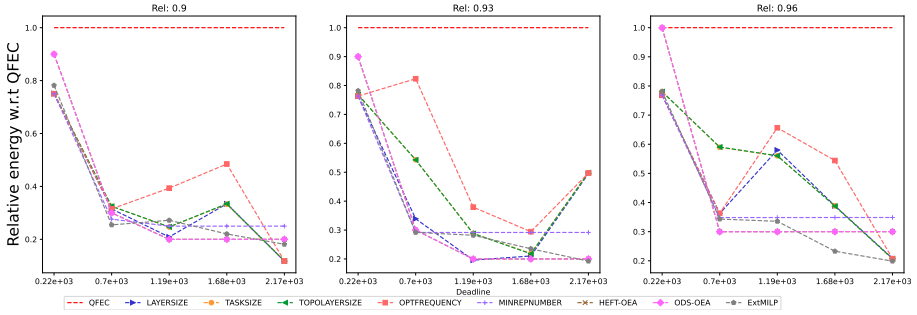

Figure 1232: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

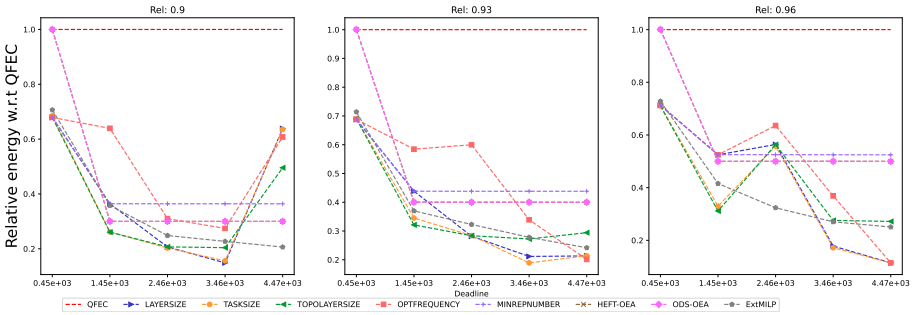

Figure 1233: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

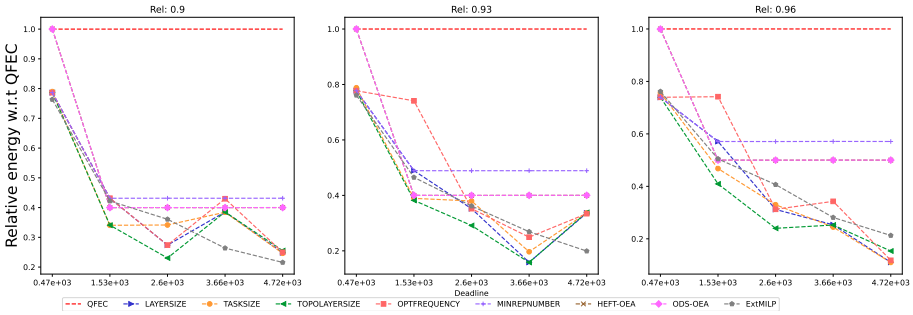

Figure 1234: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

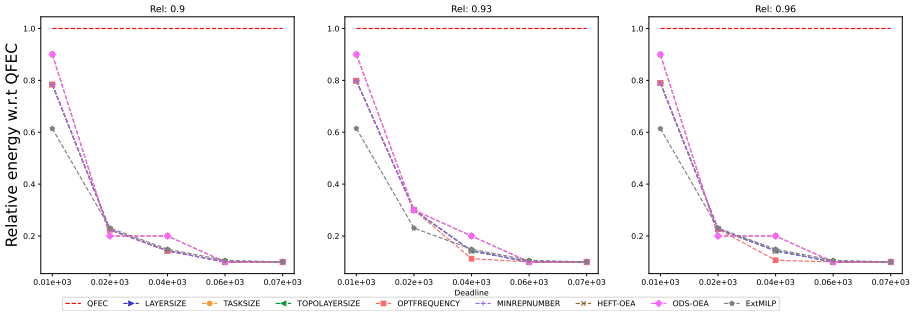

Figure 1235: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

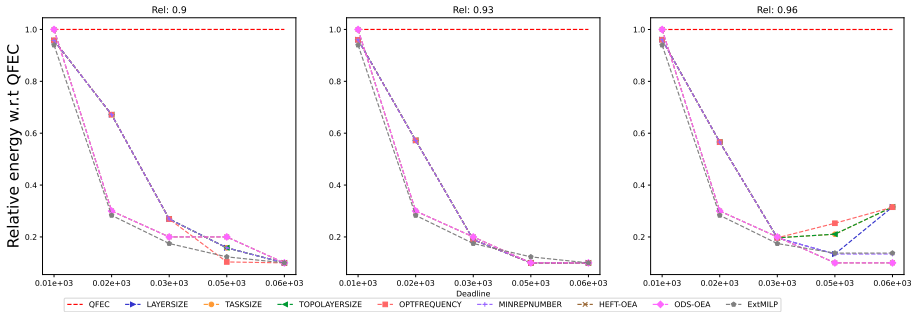

Figure 1236: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

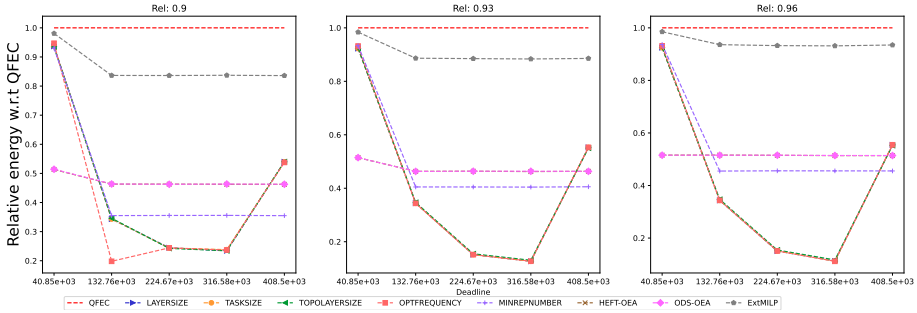

Figure 1237: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

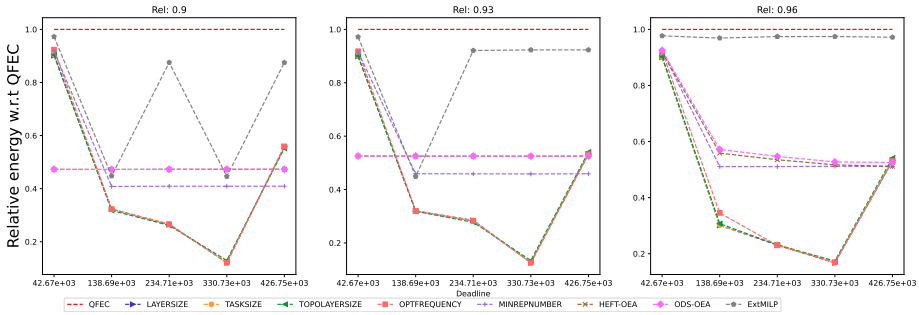

Figure 1238: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

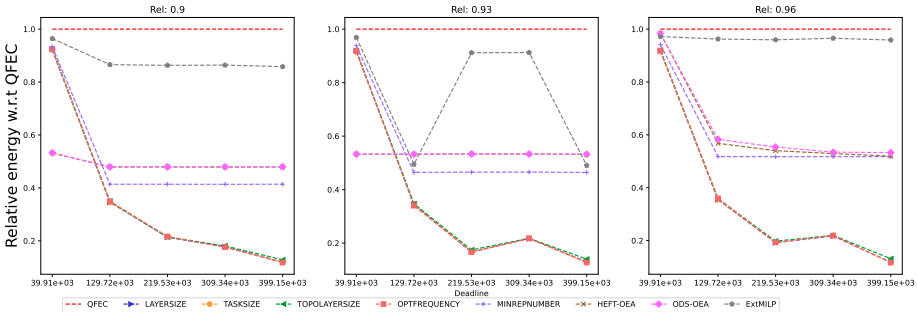

Figure 1239: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

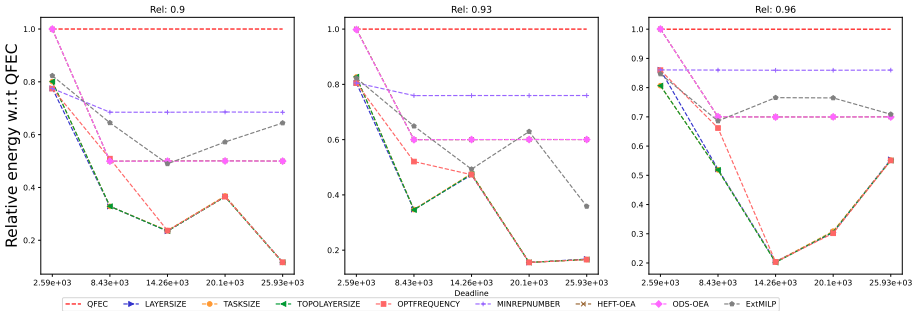

Figure 1240: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

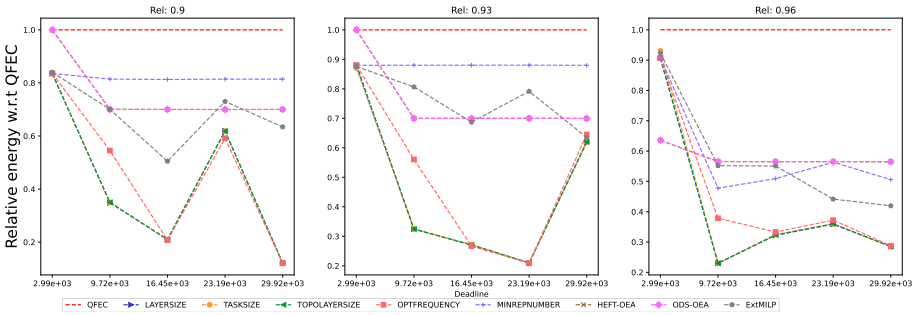

Figure 1241: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.6.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

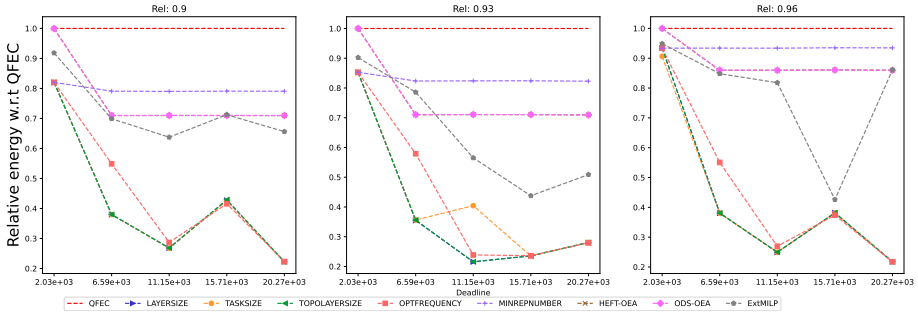

Figure 1242: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

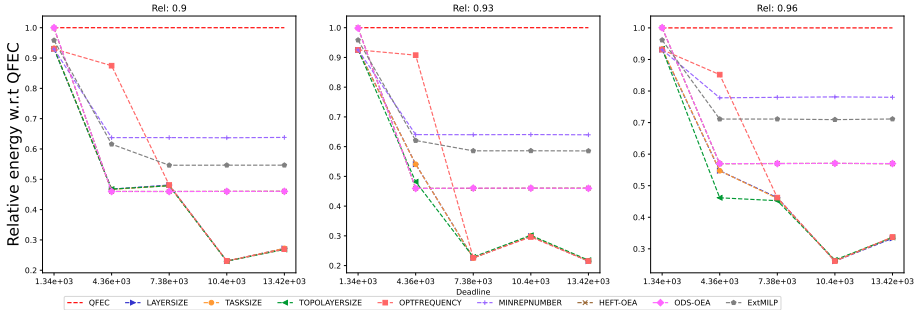

Figure 1243: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

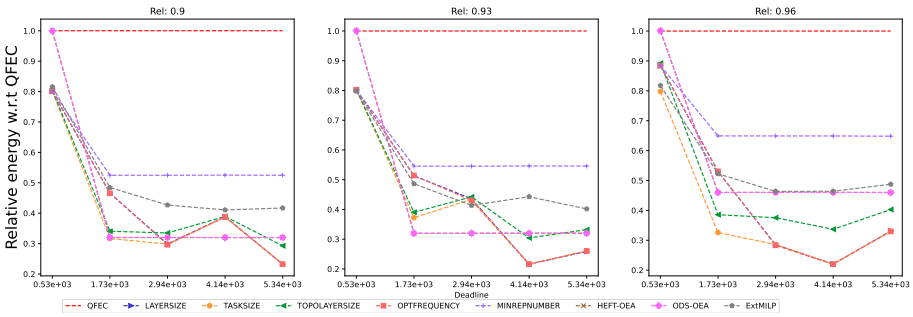

Figure 1244: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

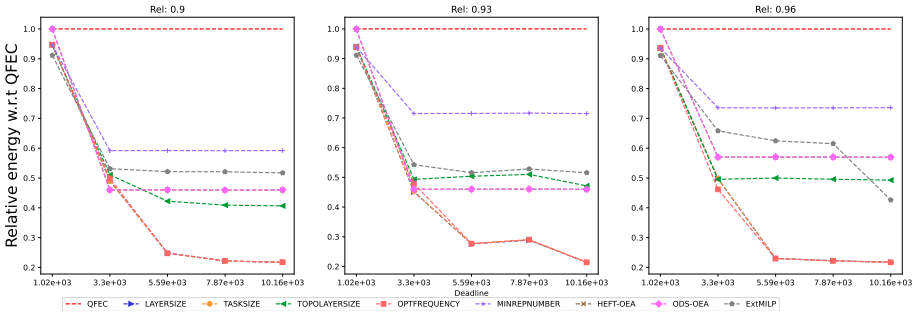

Figure 1245: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

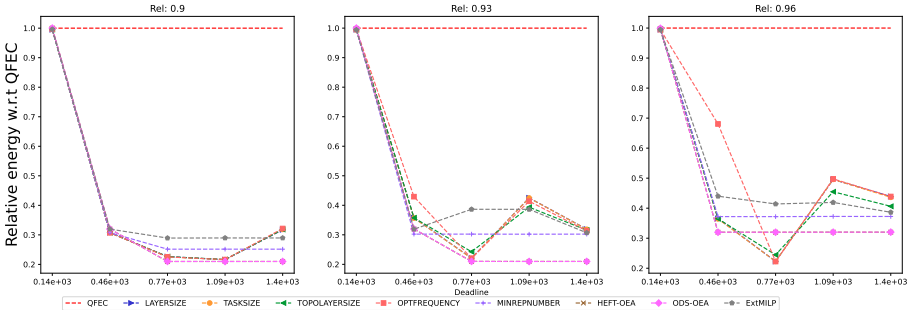

Figure 1246: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

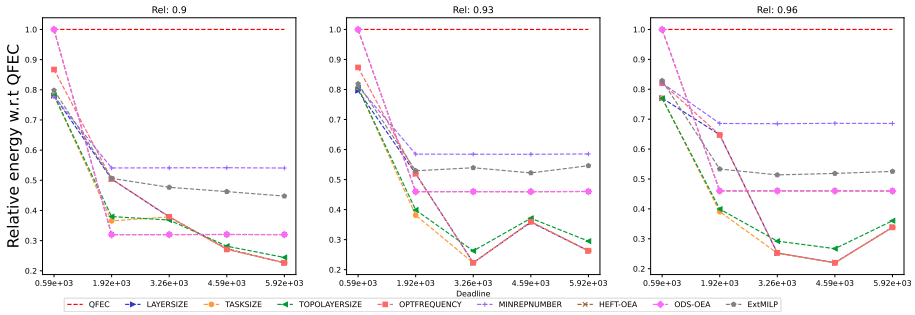

Figure 1247: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

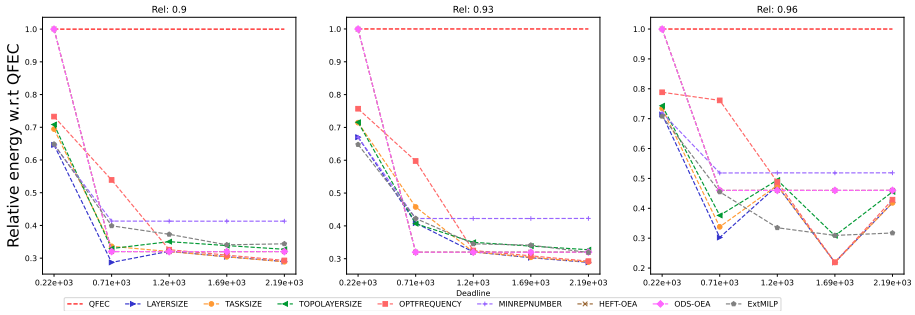

Figure 1248: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

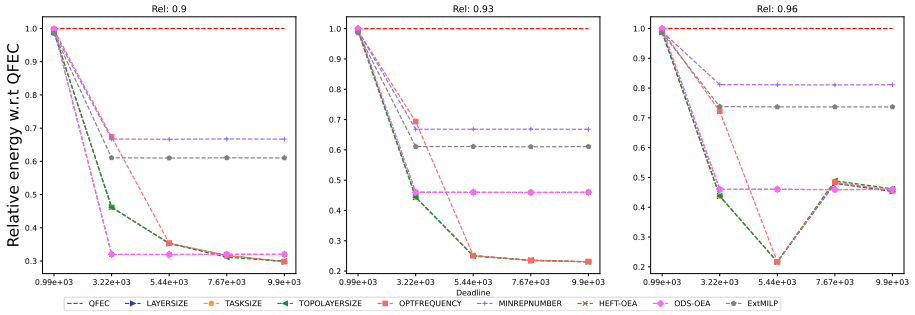

Figure 1249: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

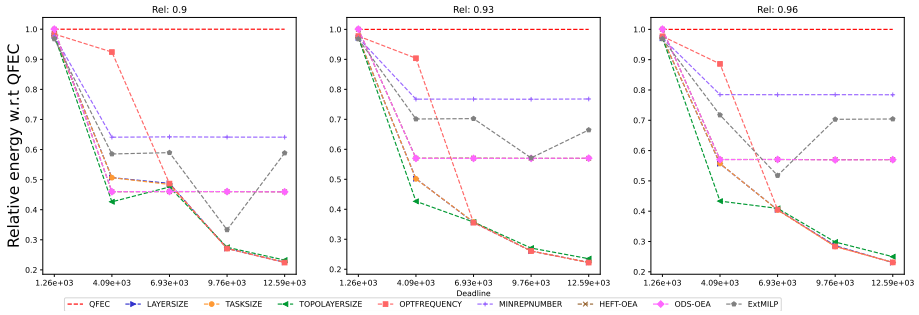

Figure 1250: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

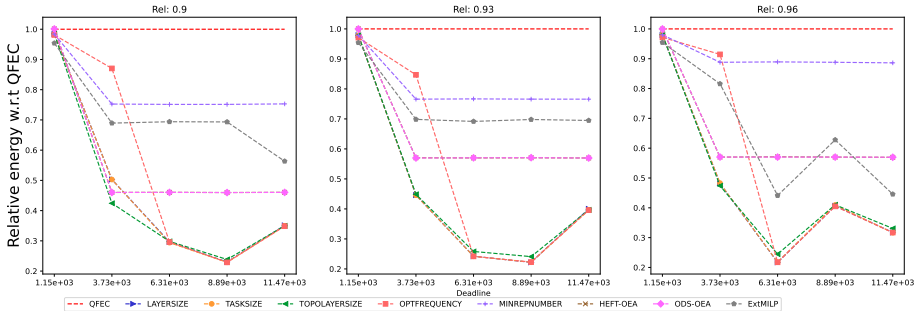

Figure 1251: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

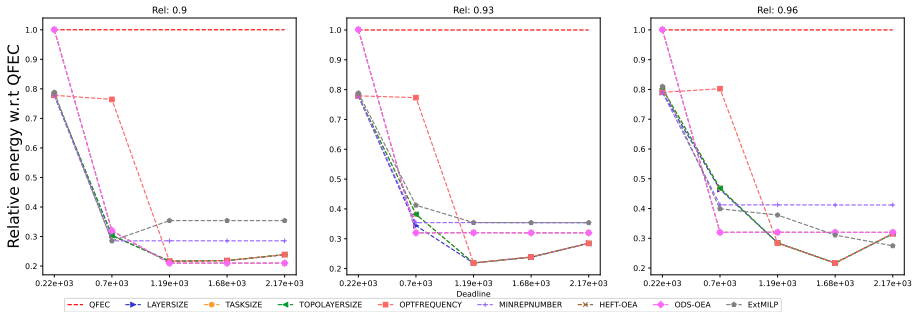

Figure 1252: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

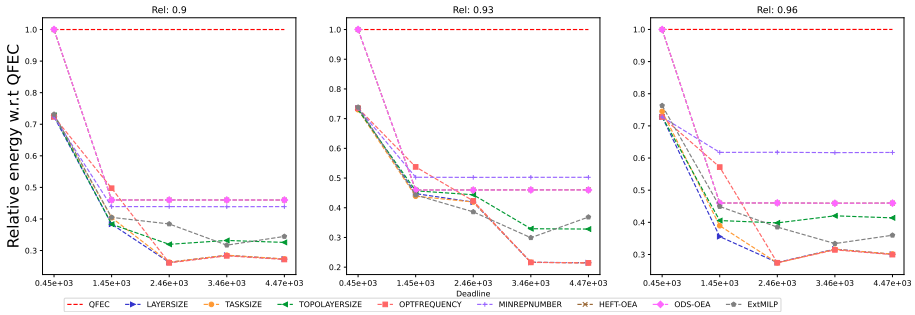

Figure 1253: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

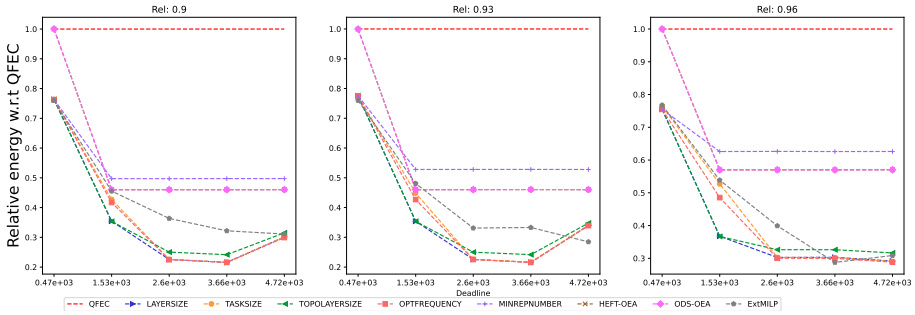

Figure 1254: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

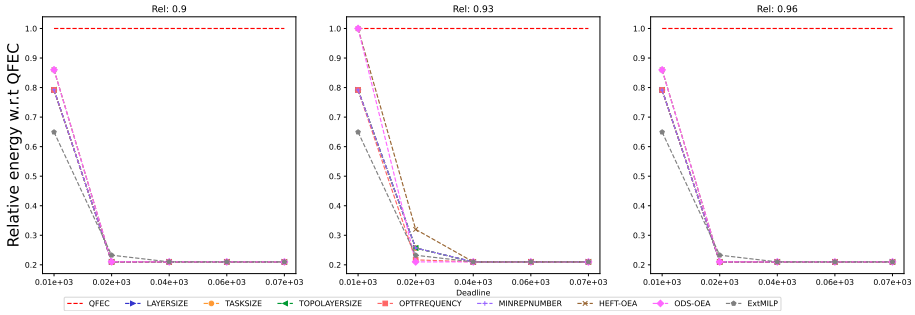

Figure 1255: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

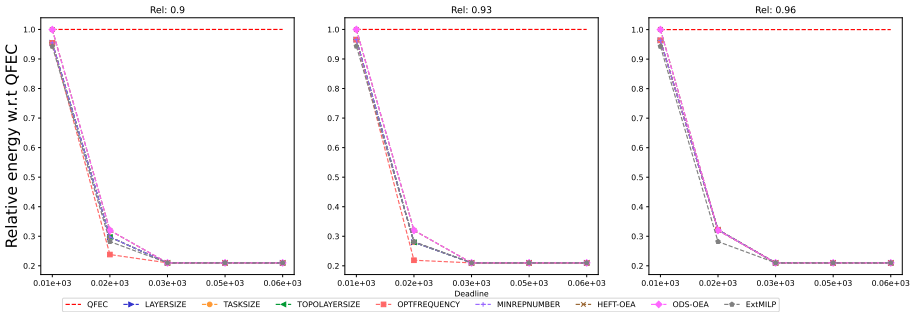

Figure 1256: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

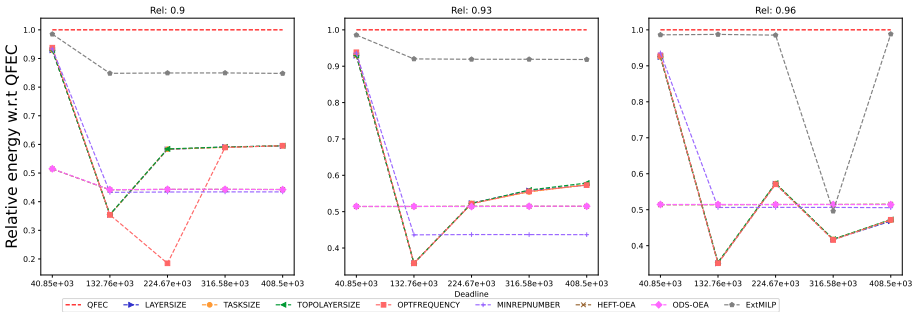

Figure 1257: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

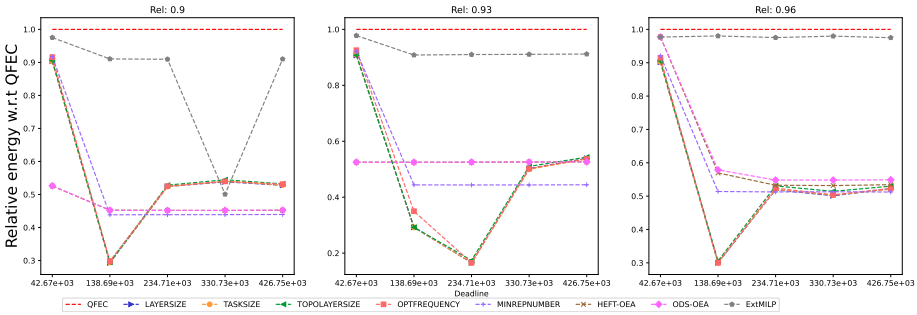

Figure 1258: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

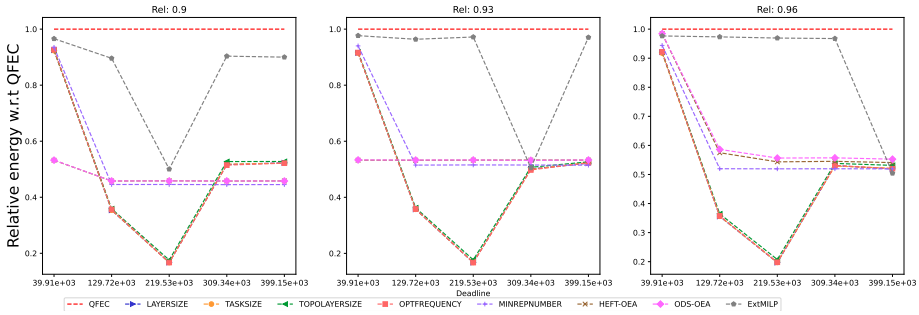

Figure 1259: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

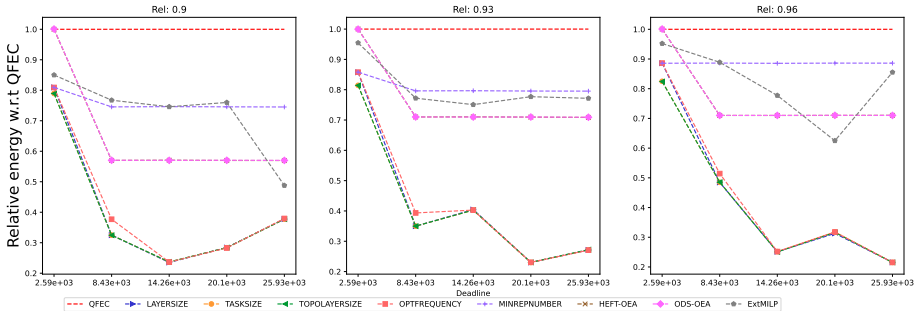

Figure 1260: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

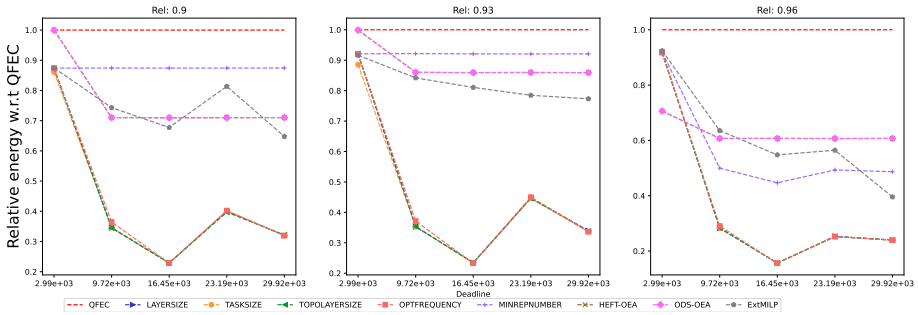

Figure 1261: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.6.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

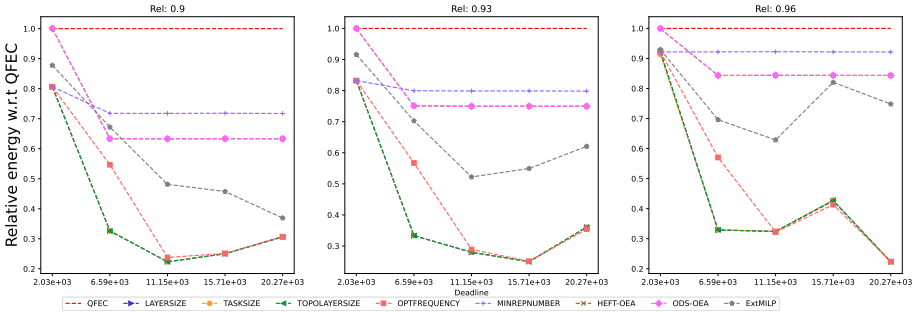

Figure 1262: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

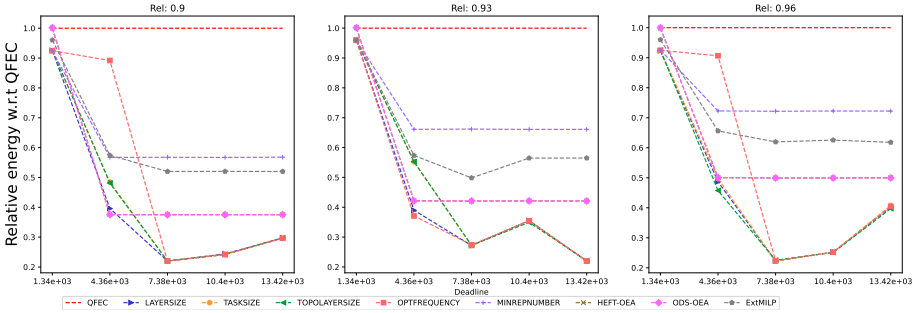

Figure 1263: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

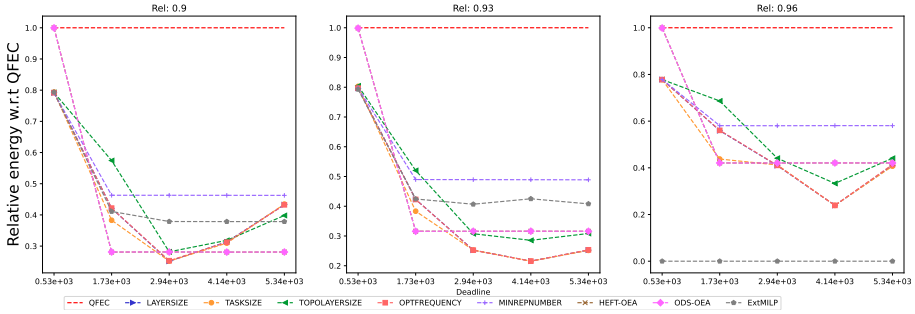

Figure 1264: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

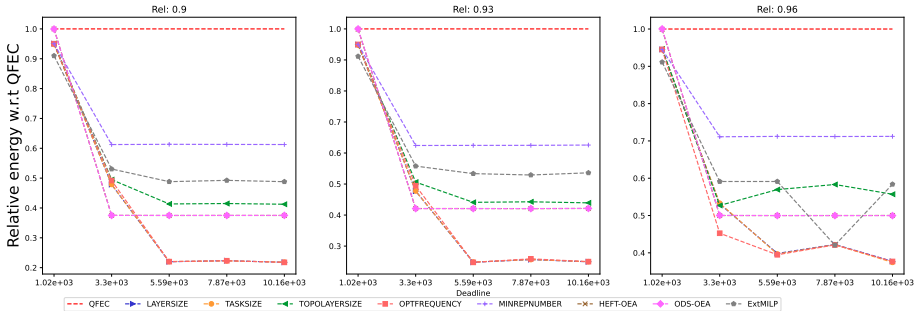

Figure 1265: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

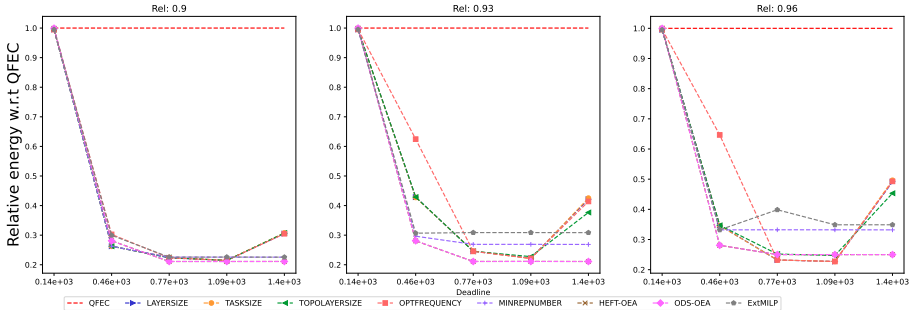

Figure 1266: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

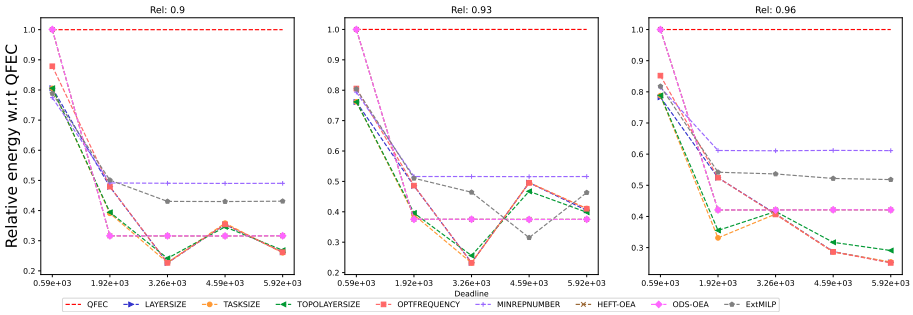

Figure 1267: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

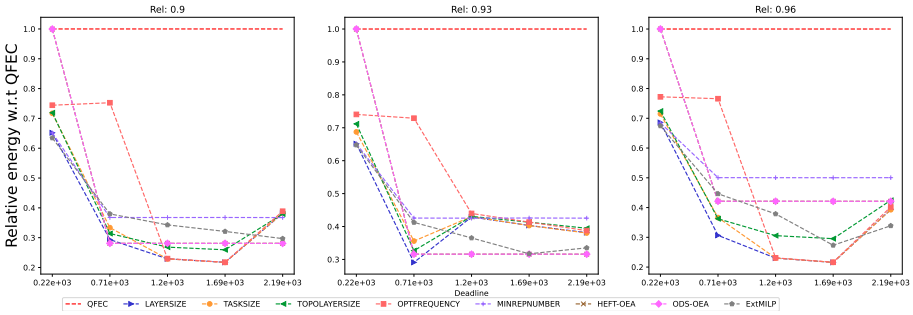

Figure 1268: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

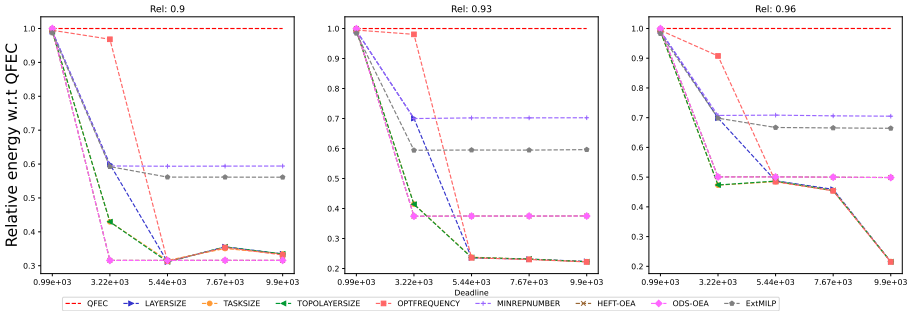

Figure 1269: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

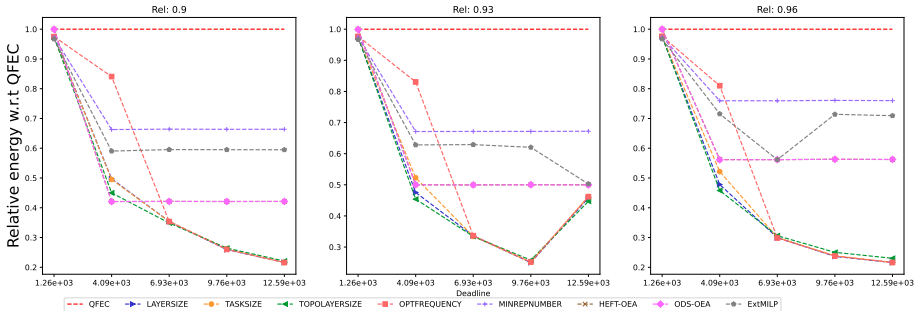

Figure 1270: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

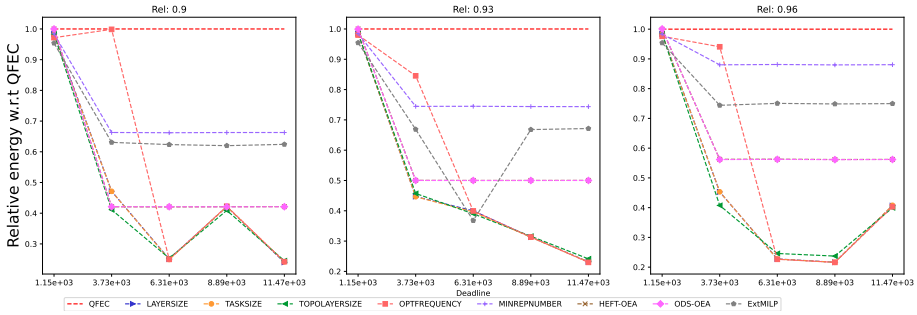

Figure 1271: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

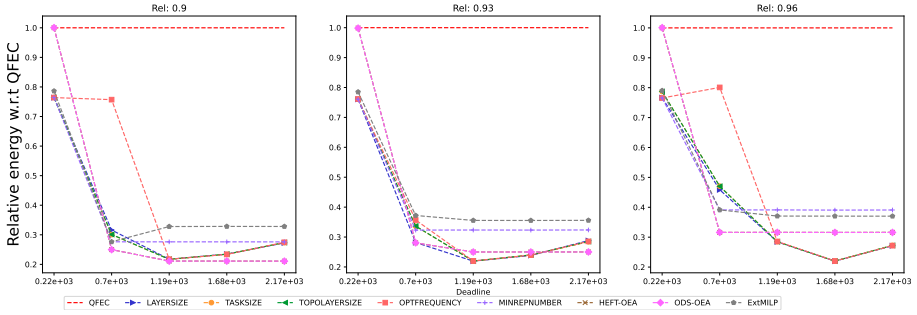

Figure 1272: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

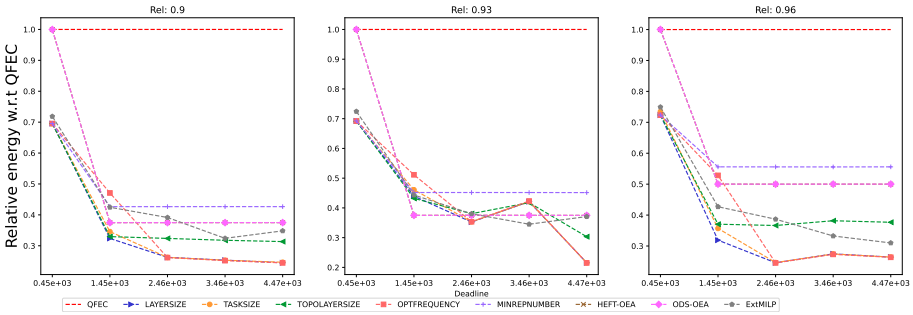

Figure 1273: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

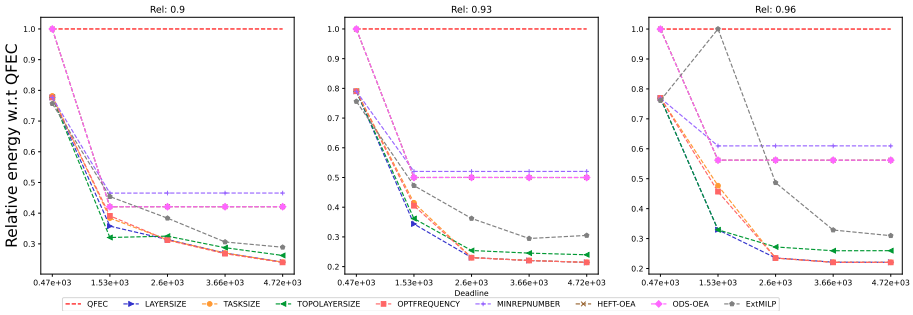

Figure 1274: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

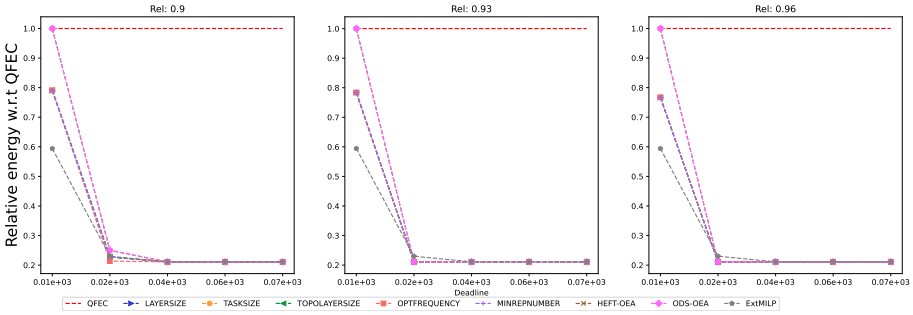

Figure 1275: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

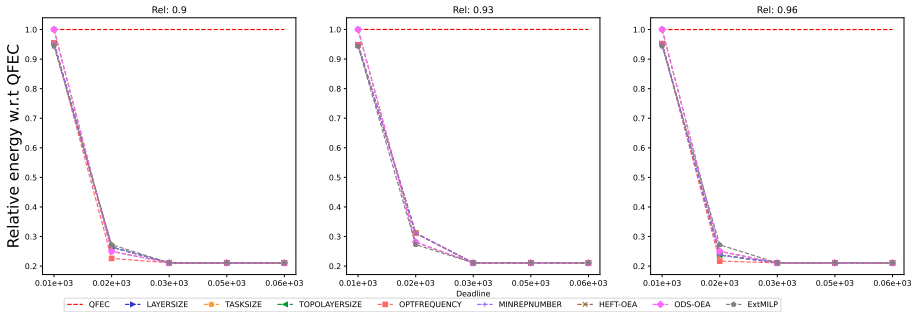

Figure 1276: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

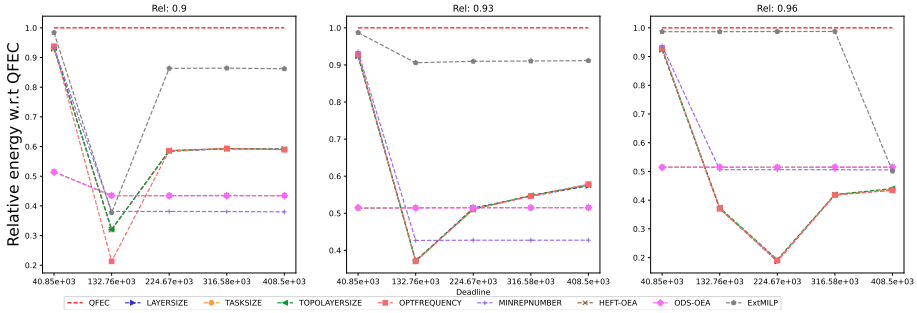

Figure 1277: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

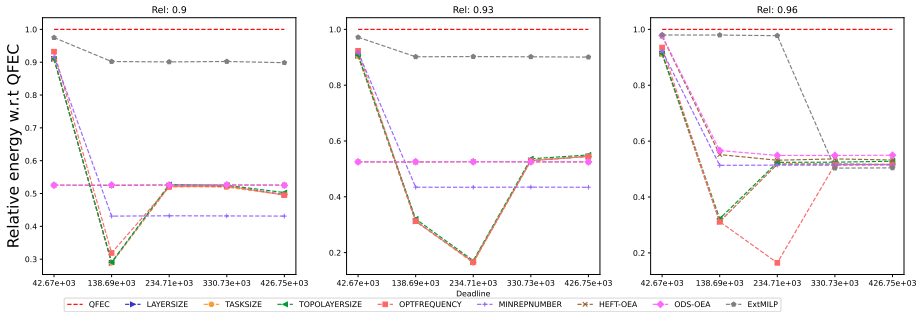

Figure 1278: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

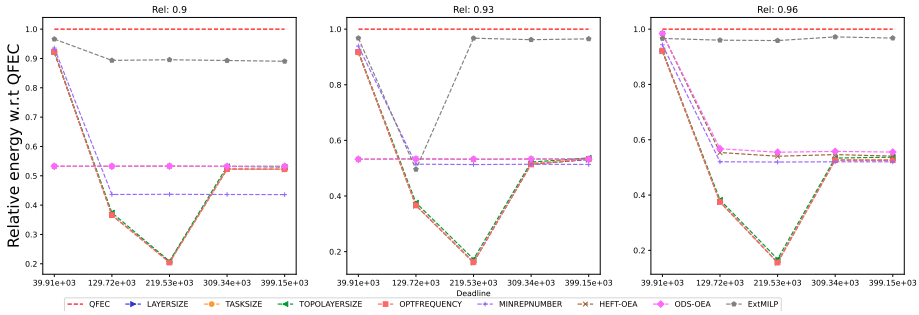

Figure 1279: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

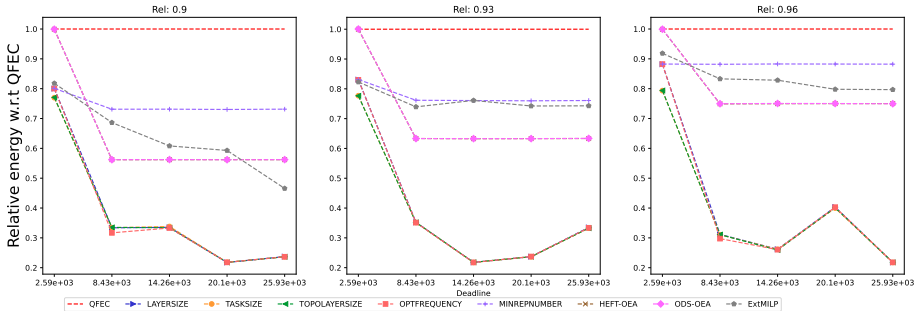

Figure 1280: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

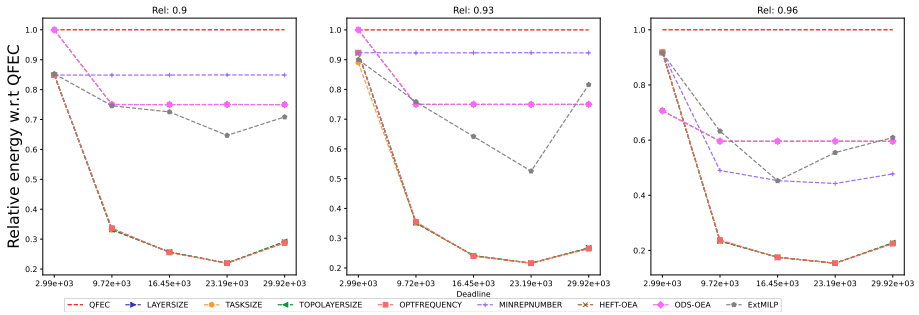

Figure 1281: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.7 $BC/WC = 0.7$

F.7.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

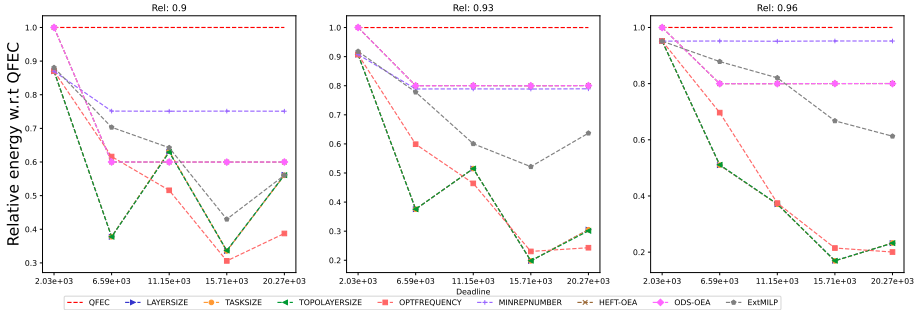

Figure 1282: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

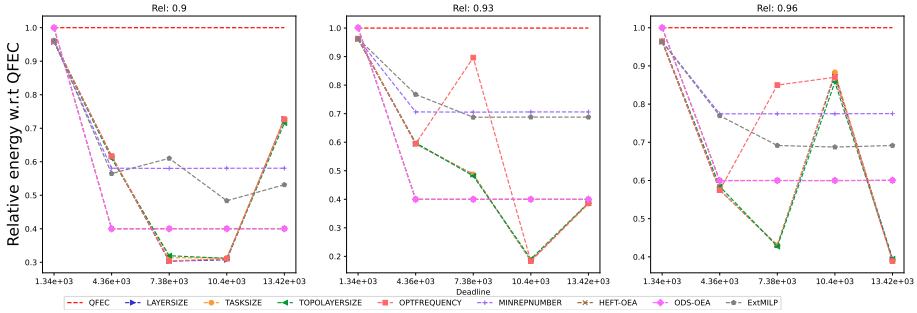

Figure 1283: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

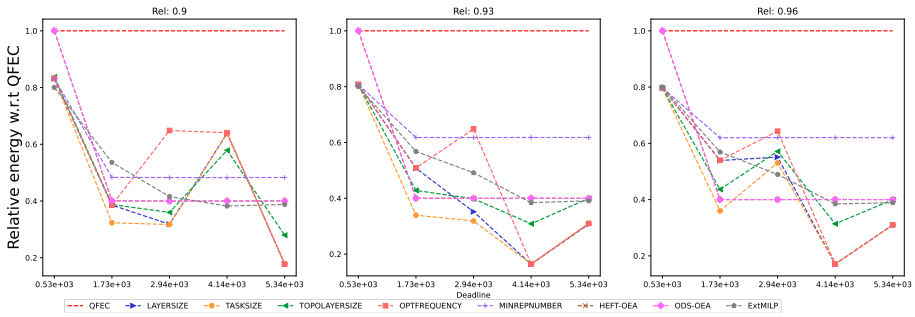

Figure 1284: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

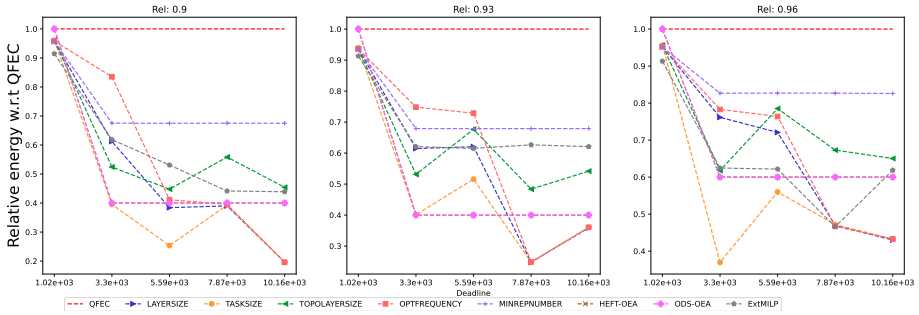

Figure 1285: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

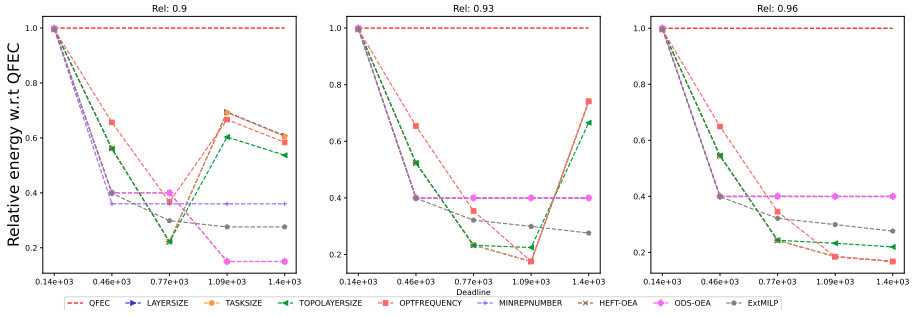

Figure 1286: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

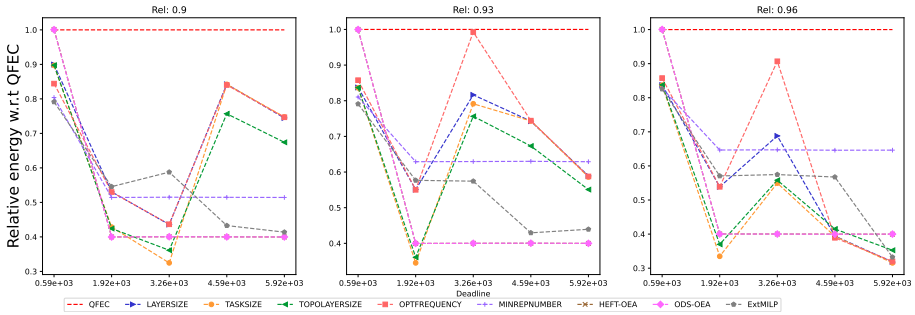

Figure 1287: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

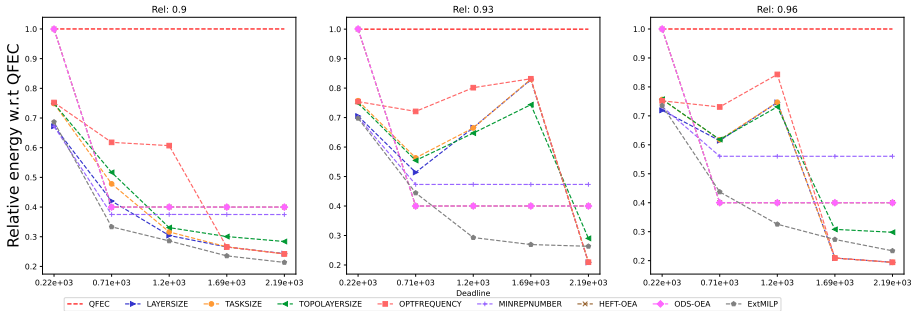

Figure 1288: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

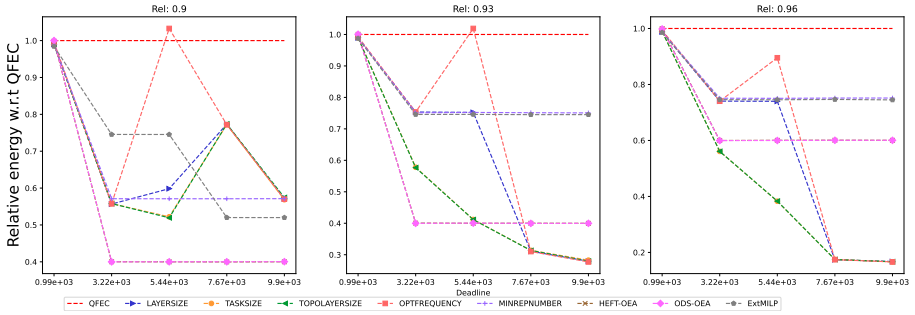

Figure 1289: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

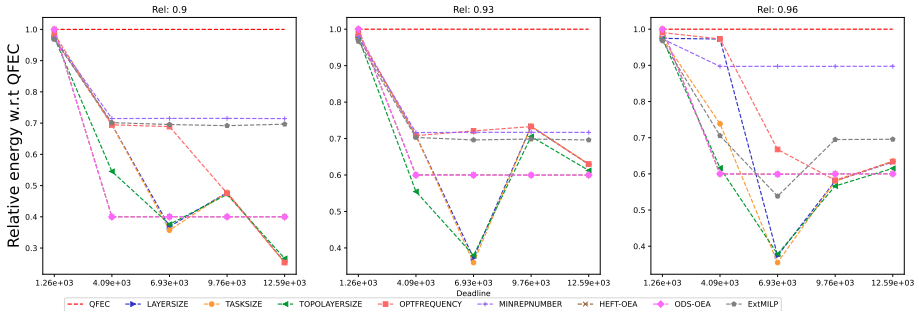

Figure 1290: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

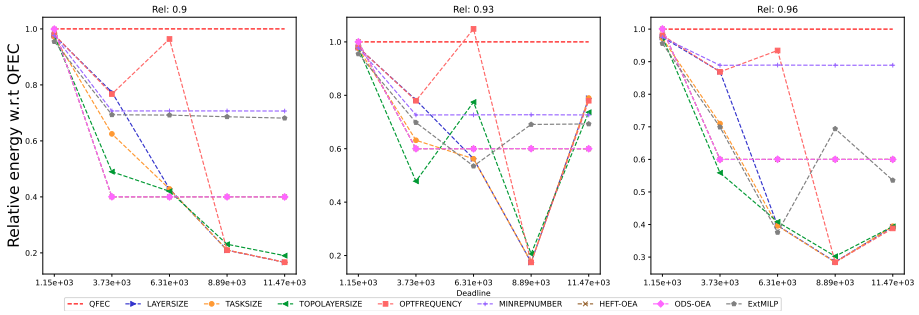

Figure 1291: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

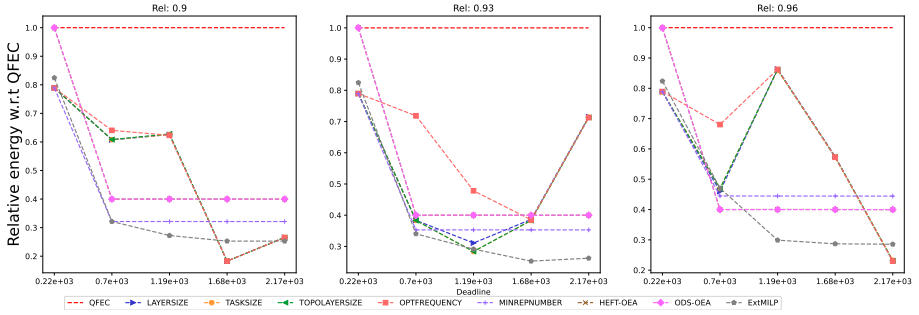

Figure 1292: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

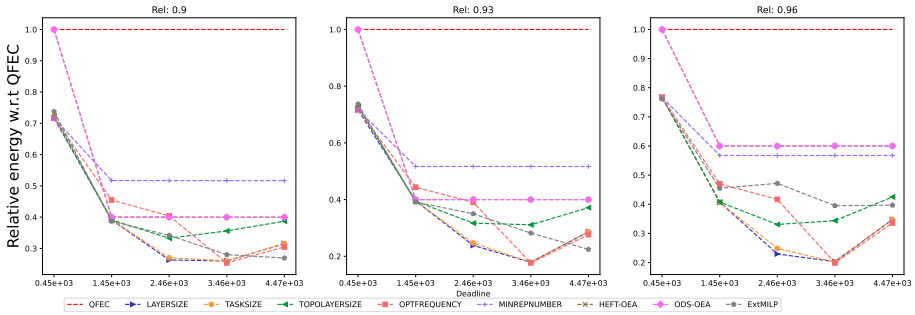

Figure 1293: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

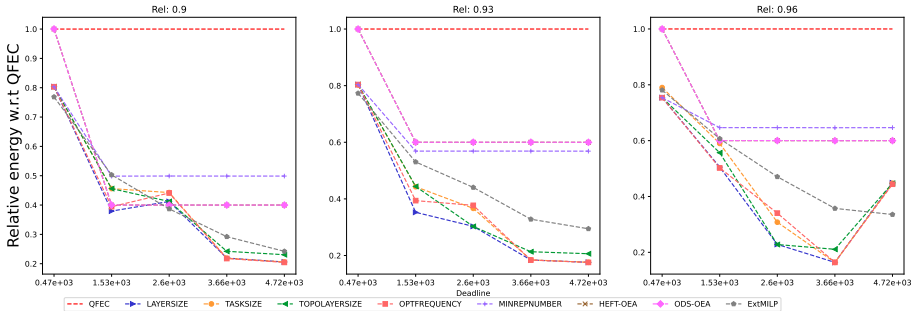

Figure 1294: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

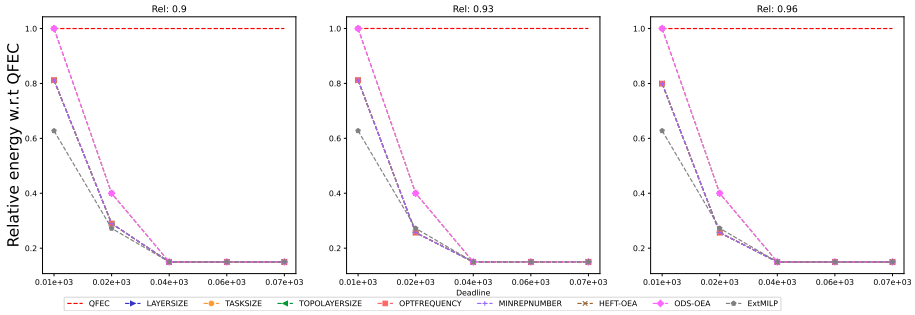

Figure 1295: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

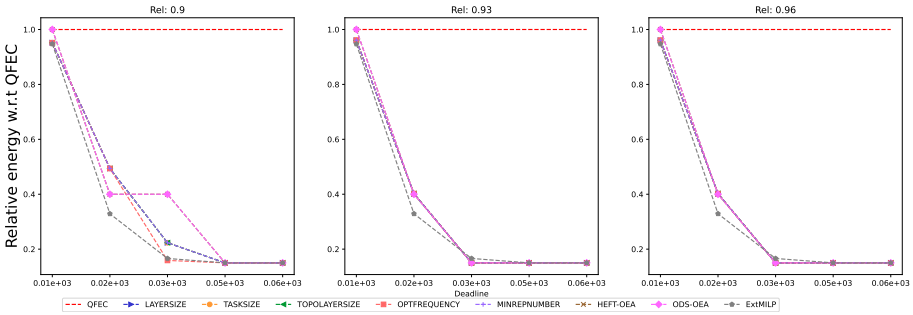

Figure 1296: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

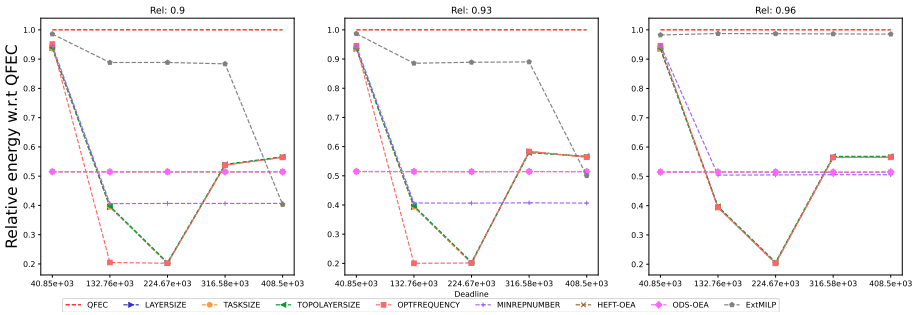

Figure 1297: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

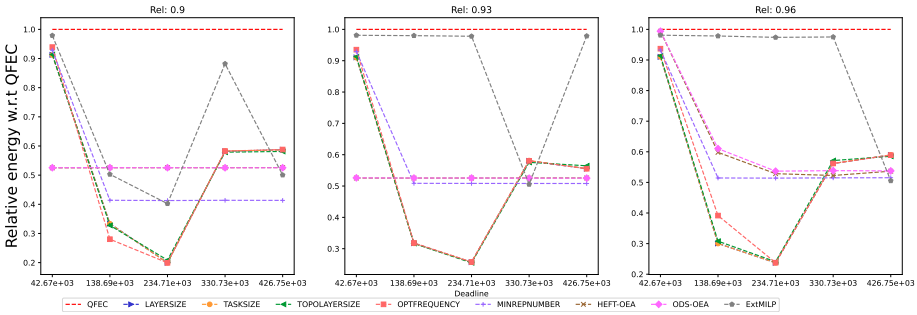

Figure 1298: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

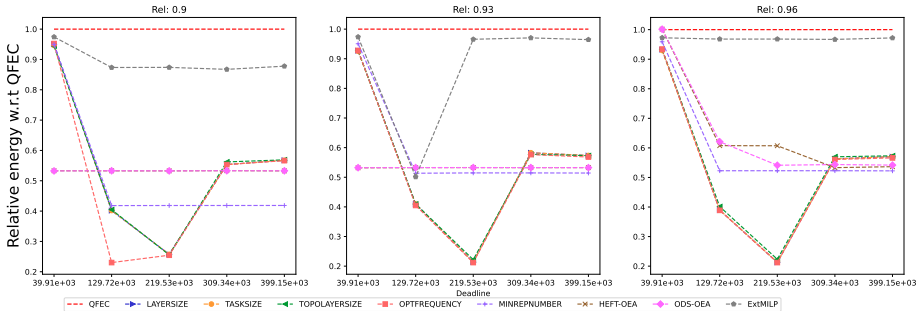

Figure 1299: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

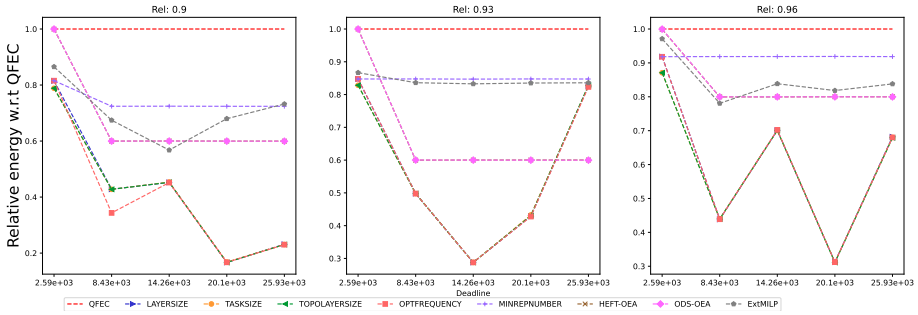

Figure 1300: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

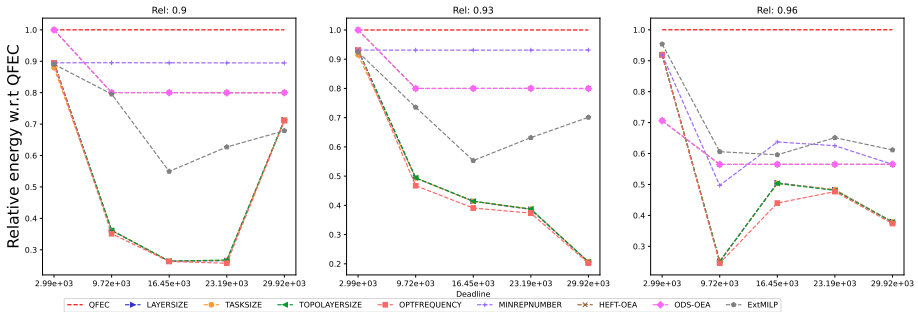

Figure 1301: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.7.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

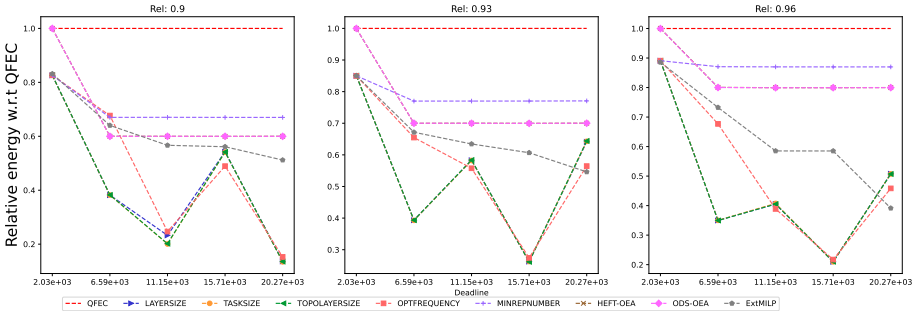

Figure 1302: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

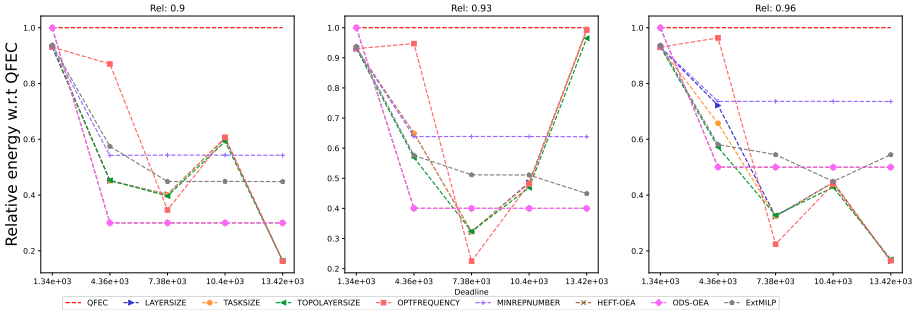

Figure 1303: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

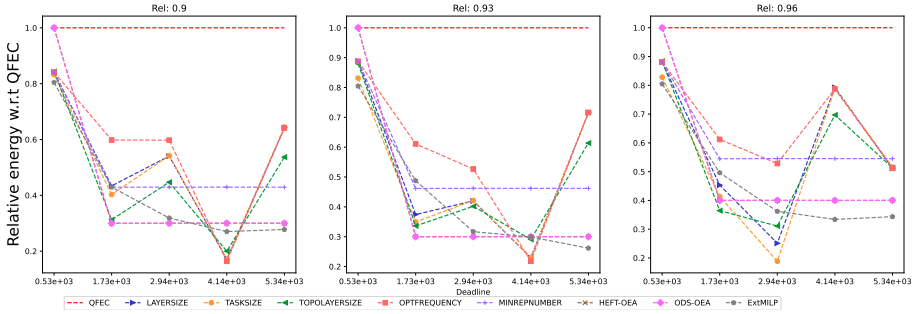

Figure 1304: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

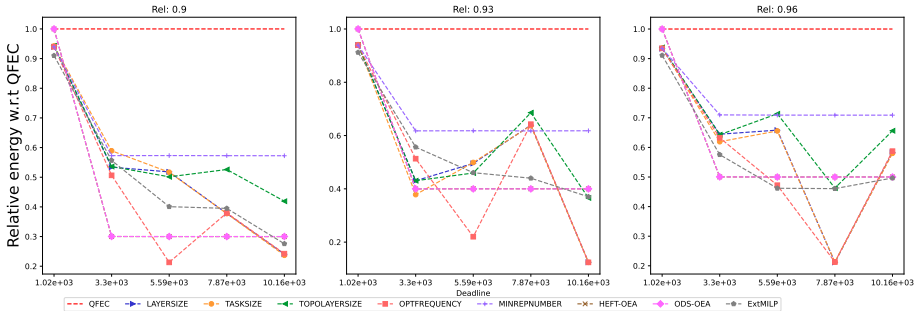

Figure 1305: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

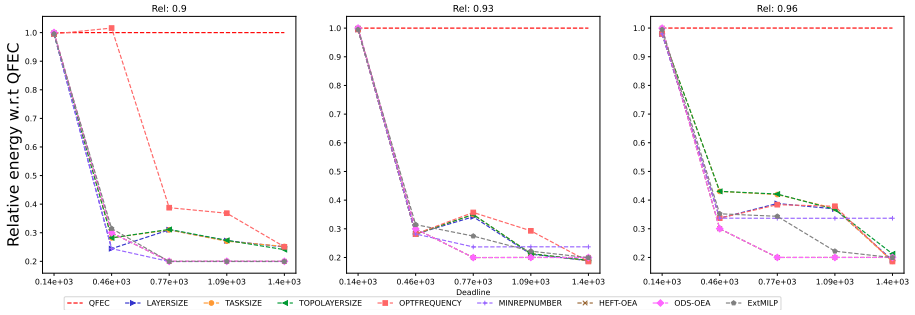

Figure 1306: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

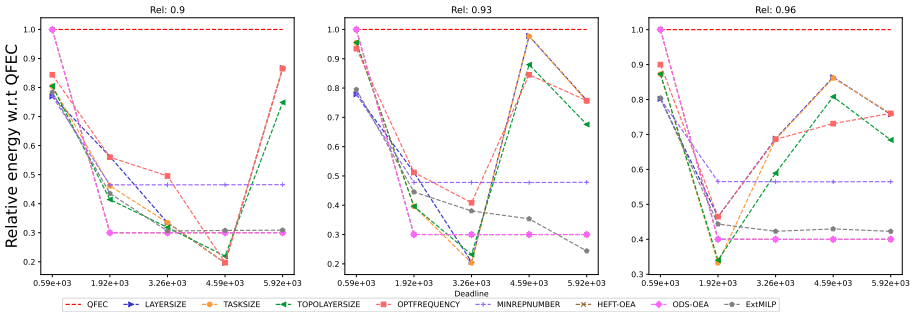

Figure 1307: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

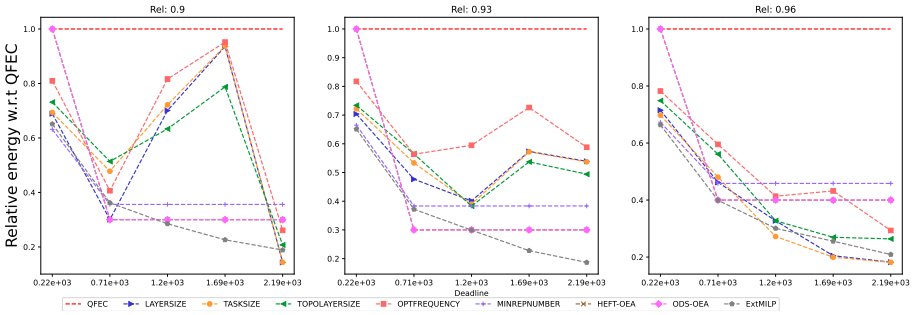

Figure 1308: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

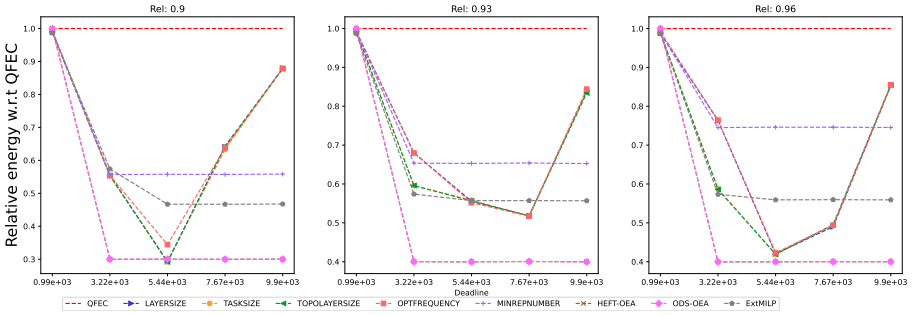

Figure 1309: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

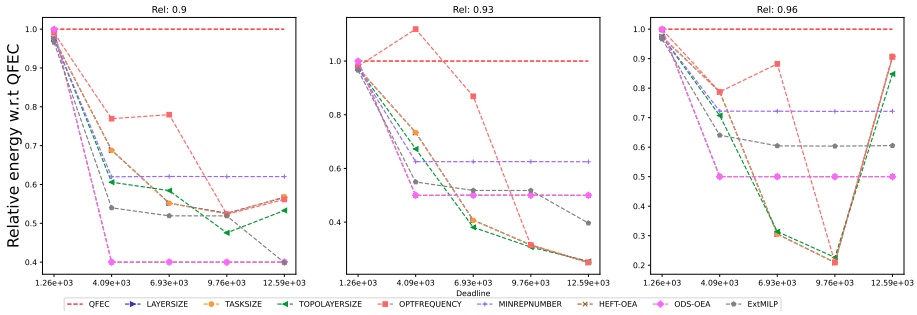

Figure 1310: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

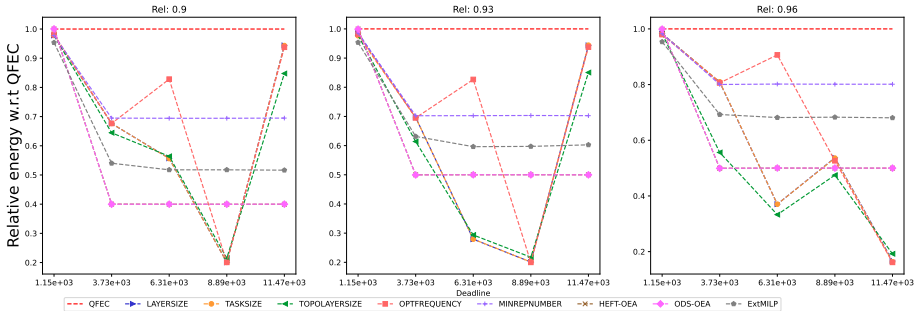

Figure 1311: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

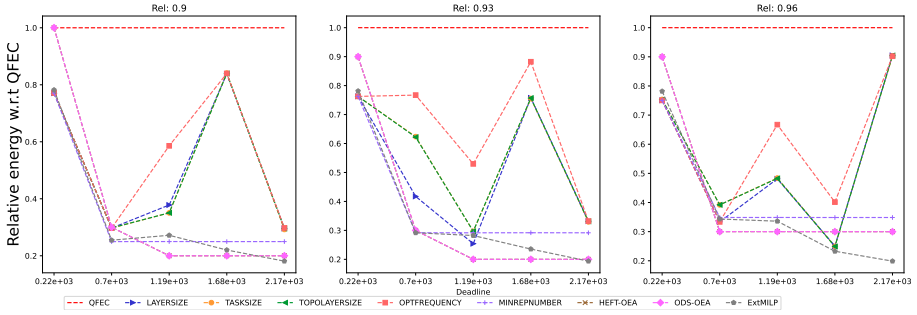

Figure 1312: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

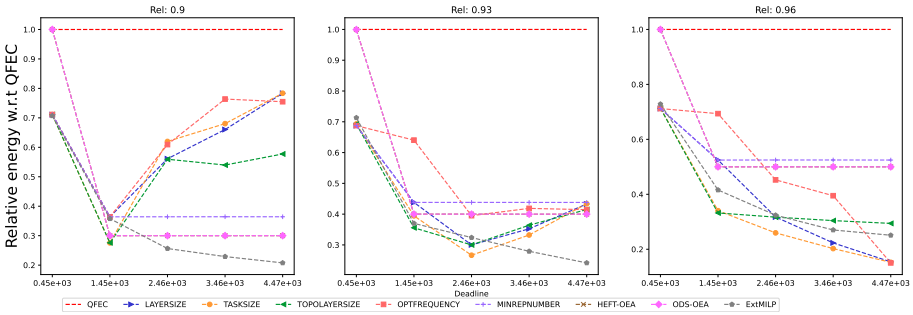

Figure 1313: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

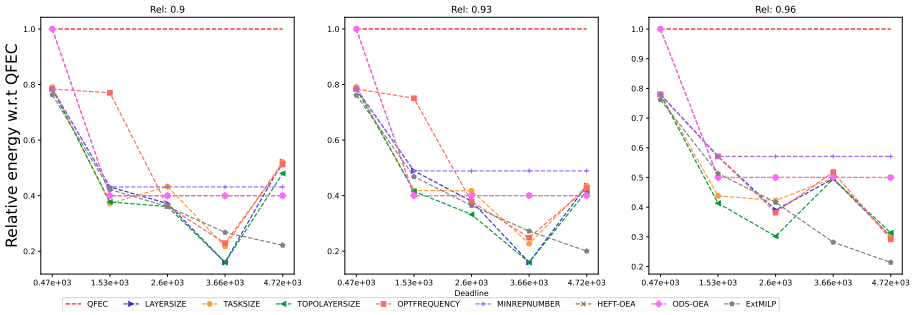

Figure 1314: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

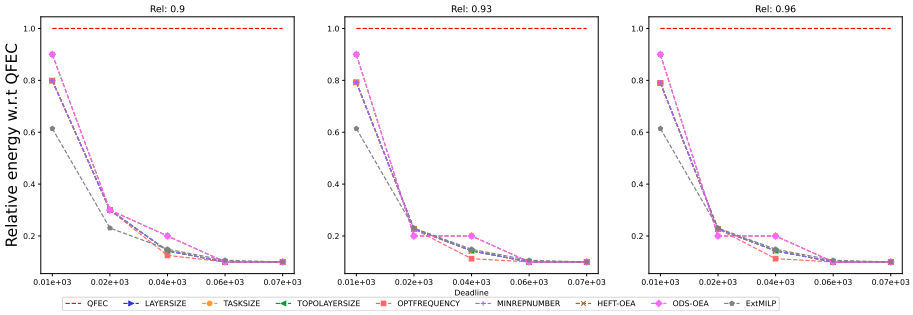

Figure 1315: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

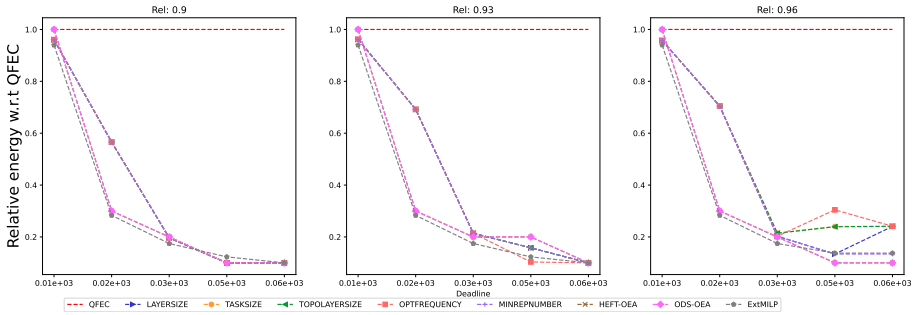

Figure 1316: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

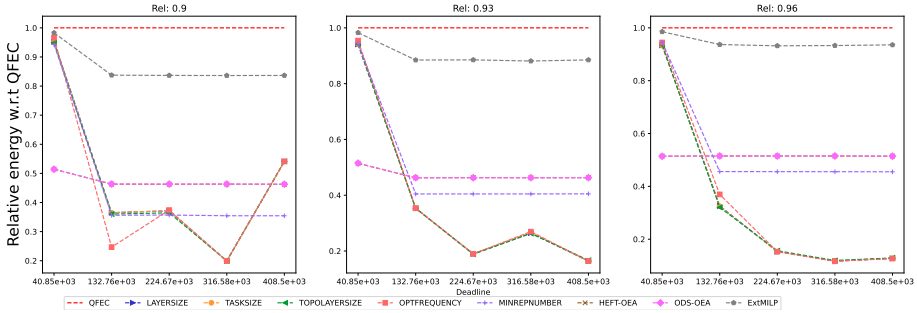

Figure 1317: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

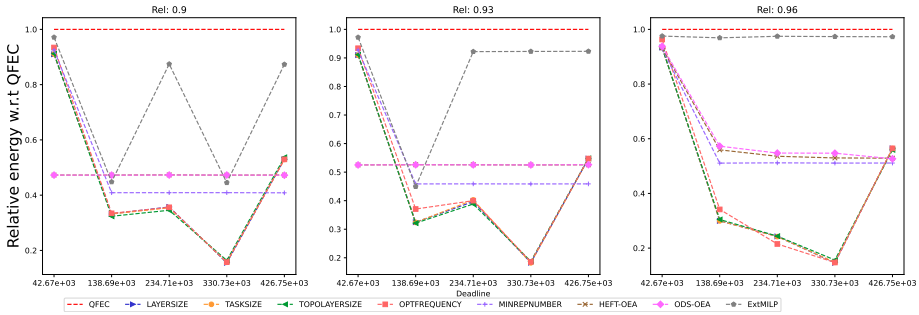

Figure 1318: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

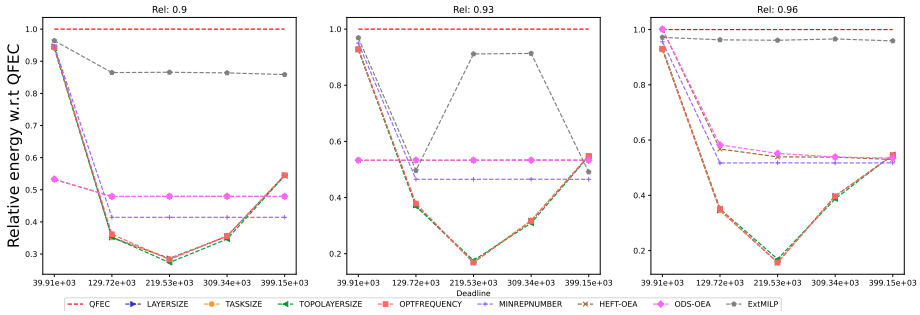

Figure 1319: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

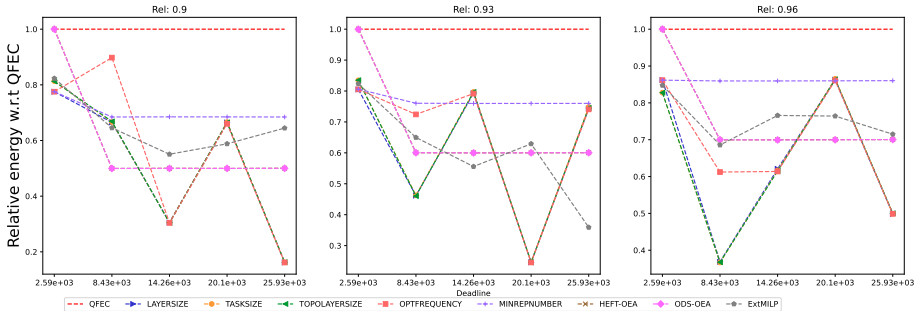

Figure 1320: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

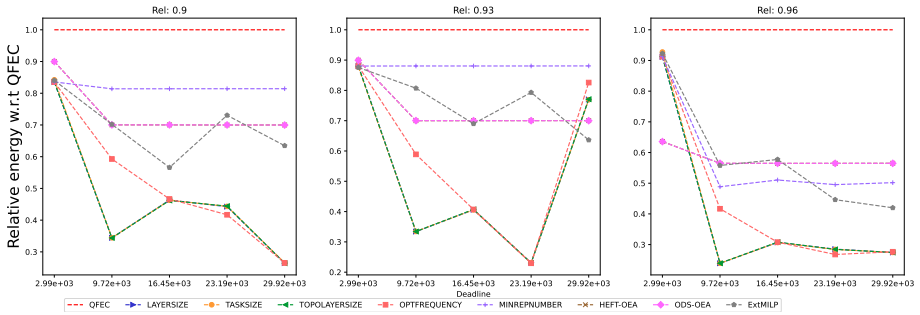

Figure 1321: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.7.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

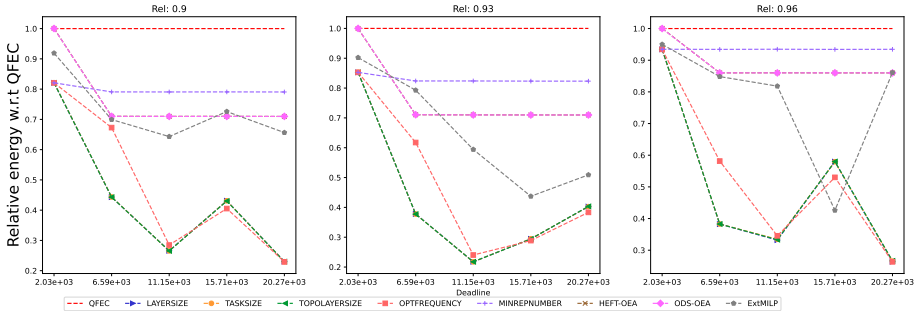

Figure 1322: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

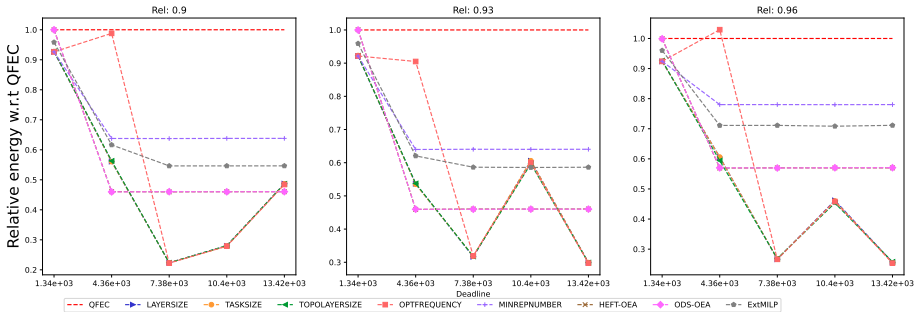

Figure 1323: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

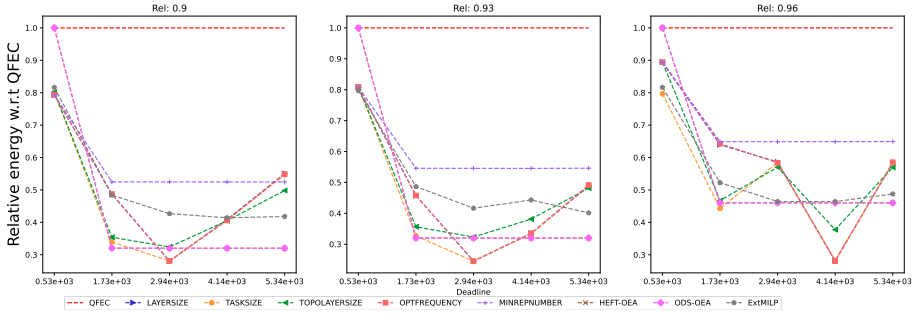

Figure 1324: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

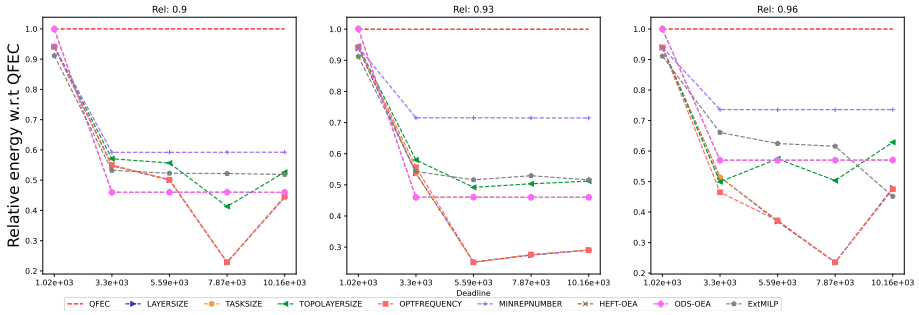

Figure 1325: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

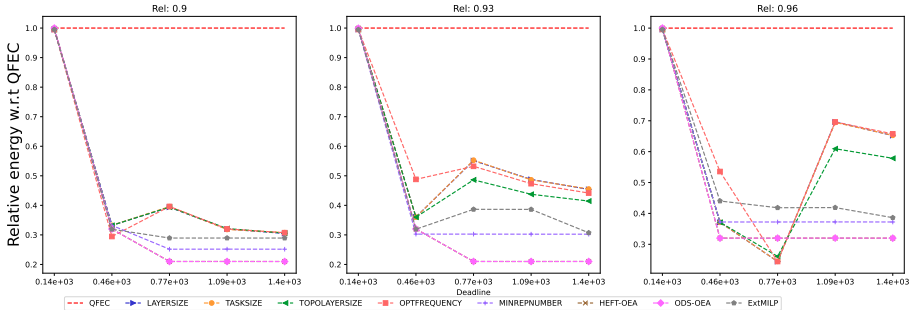

Figure 1326: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

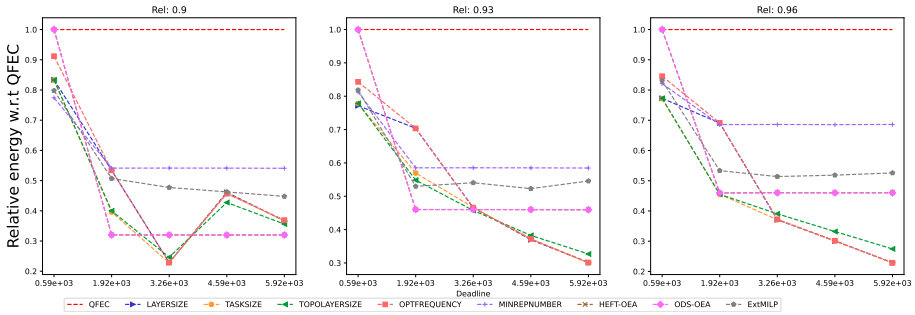

Figure 1327: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

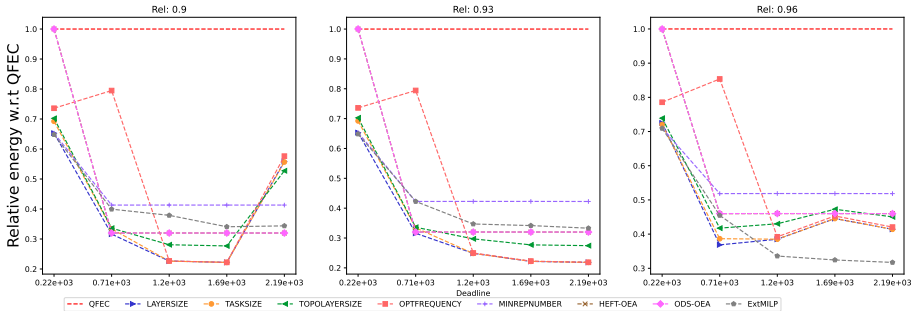

Figure 1328: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

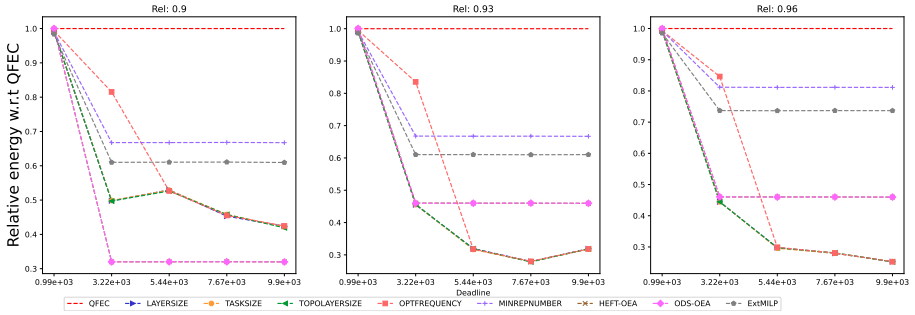

Figure 1329: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

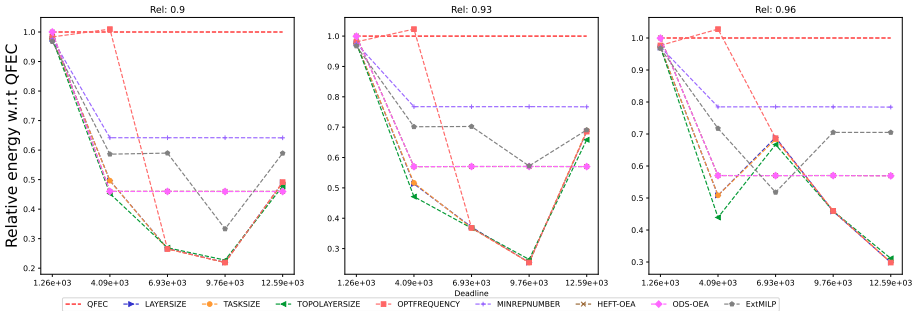

Figure 1330: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

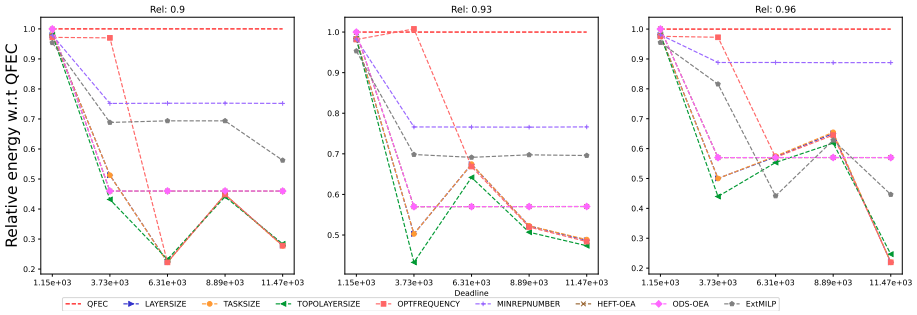

Figure 1331: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

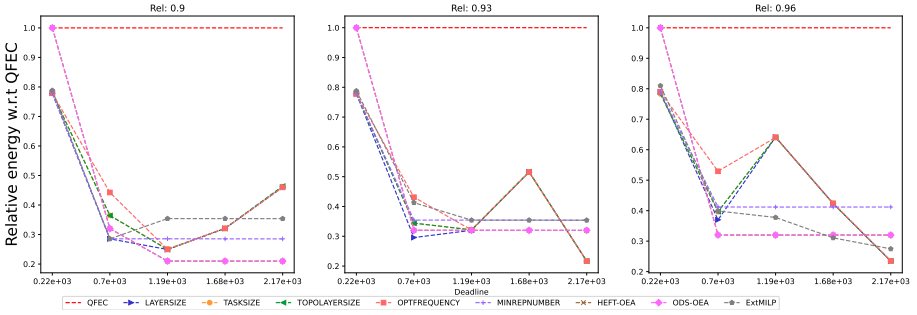

Figure 1332: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

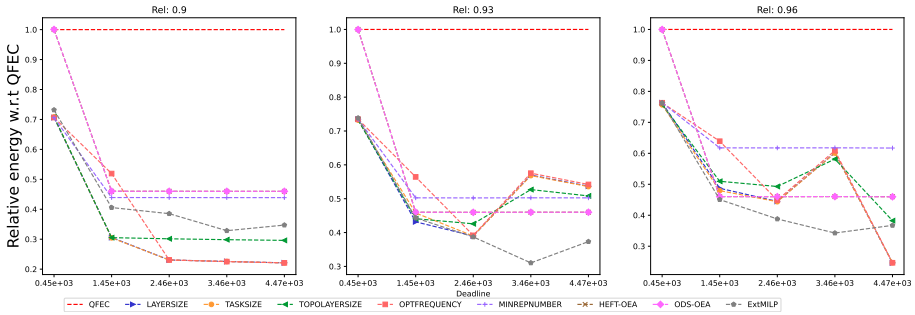

Figure 1333: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

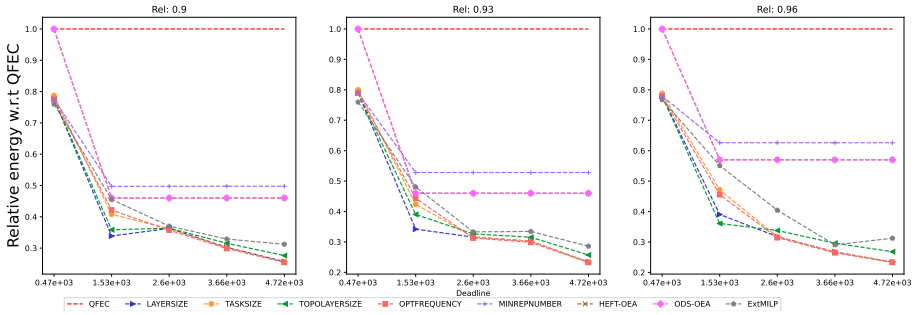

Figure 1334: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

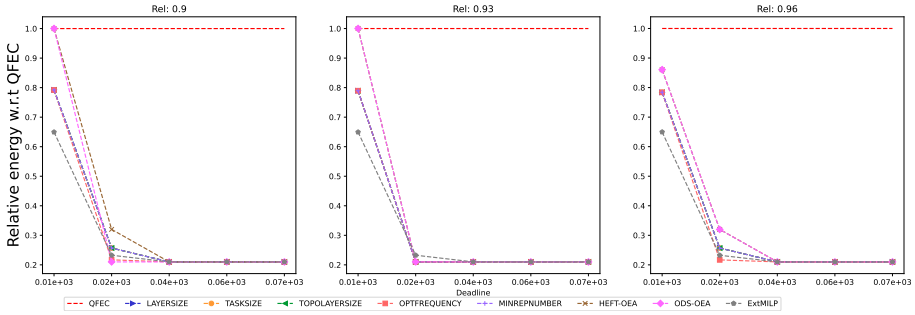

Figure 1335: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

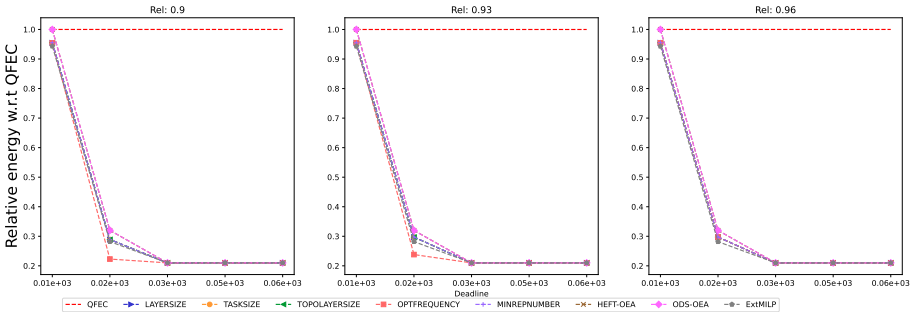

Figure 1336: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

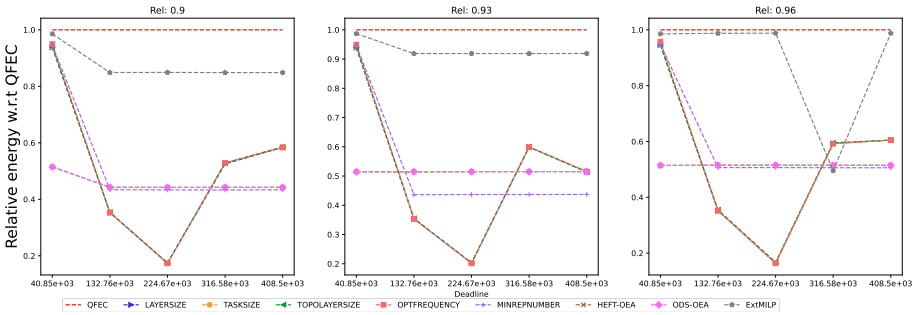

Figure 1337: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

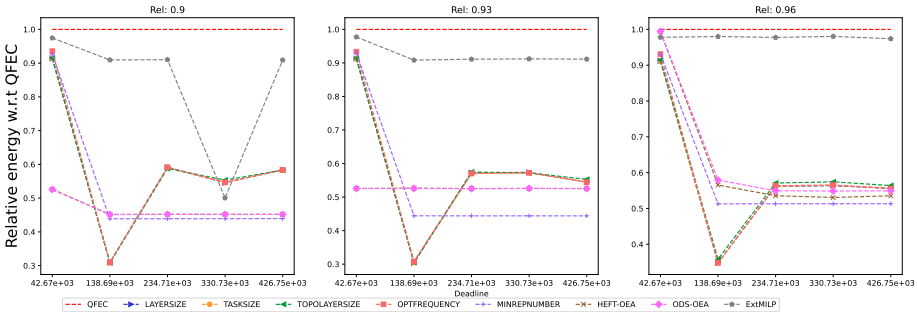

Figure 1338: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

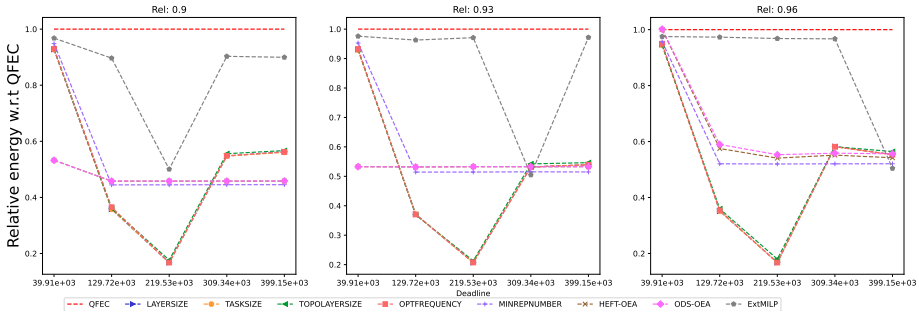

Figure 1339: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

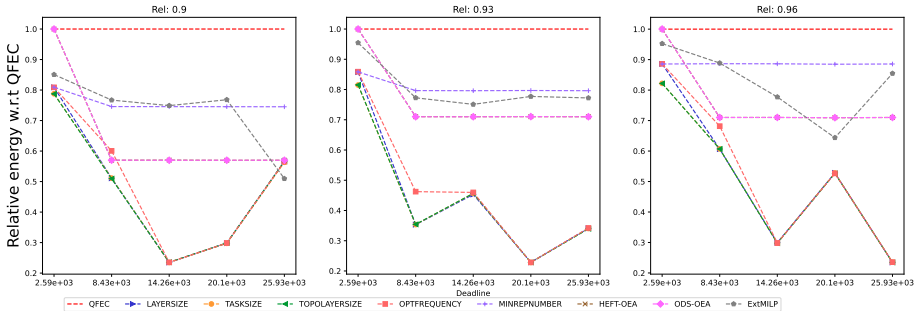

Figure 1340: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

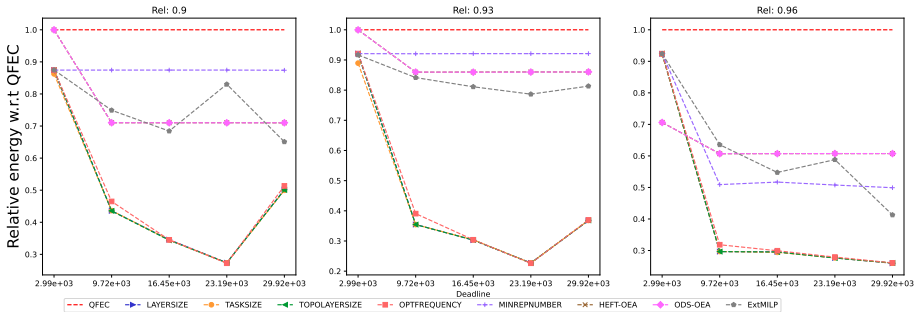

Figure 1341: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.7.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

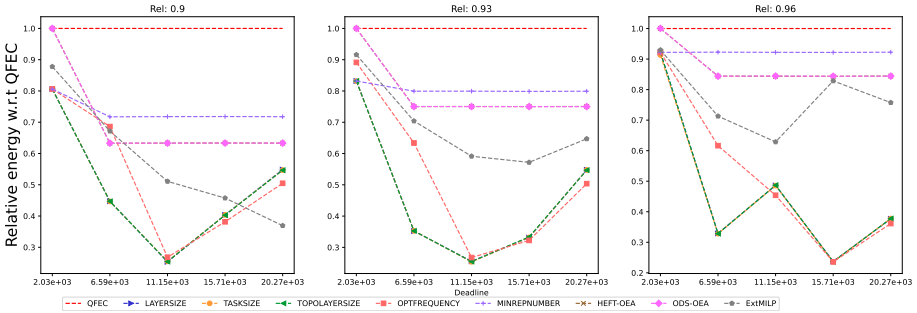

Figure 1342: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

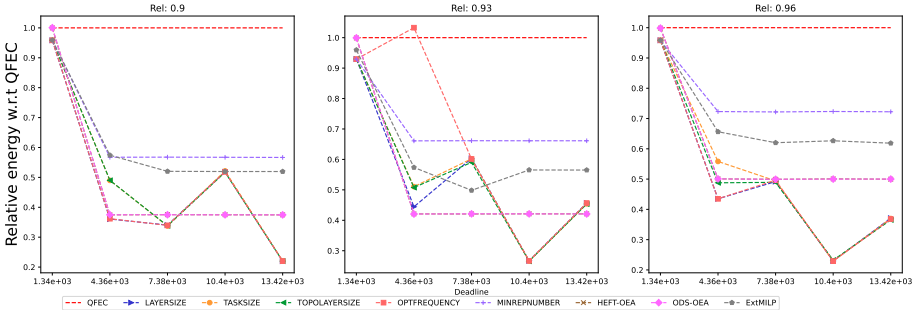

Figure 1343: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

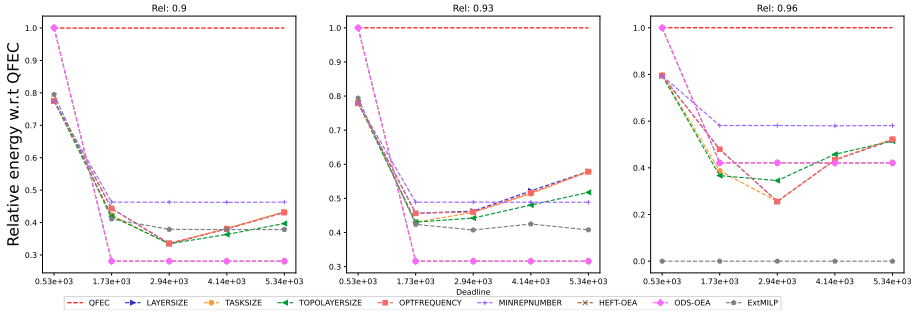

Figure 1344: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

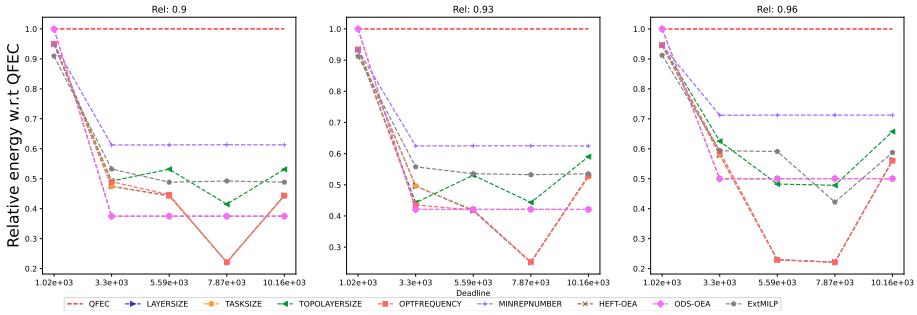

Figure 1345: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

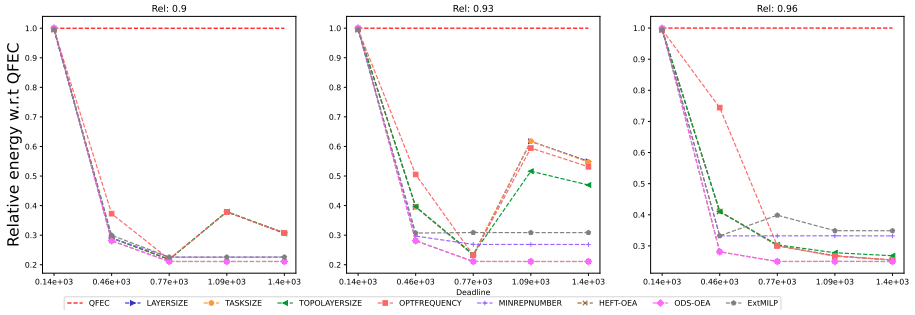

Figure 1346: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

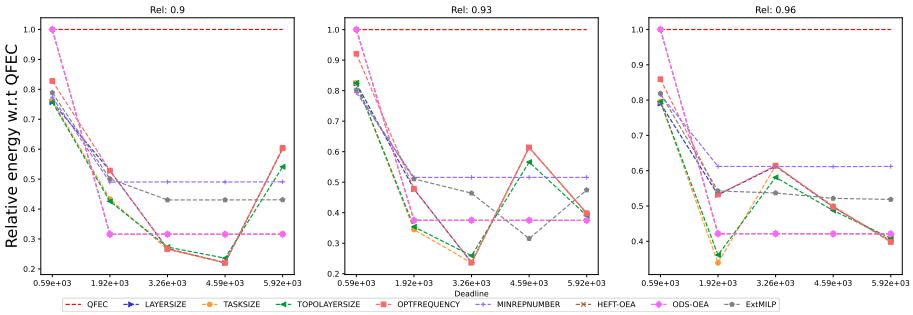

Figure 1347: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

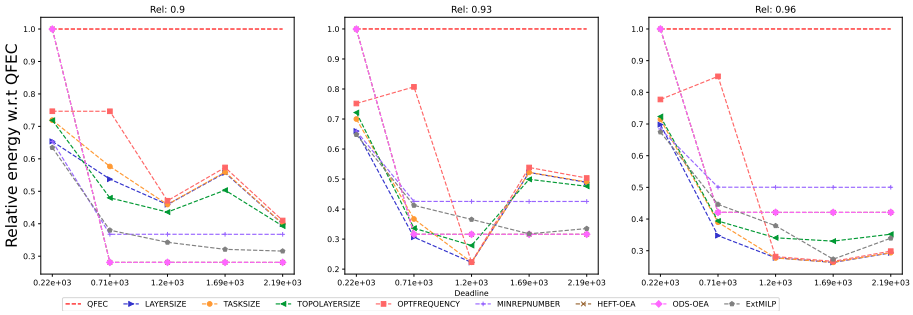

Figure 1348: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

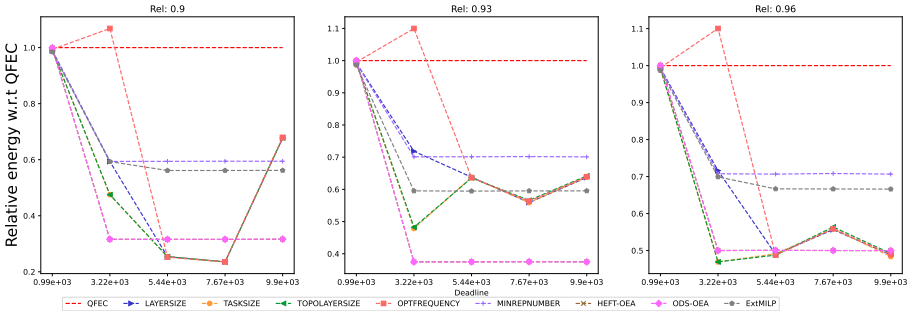

Figure 1349: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

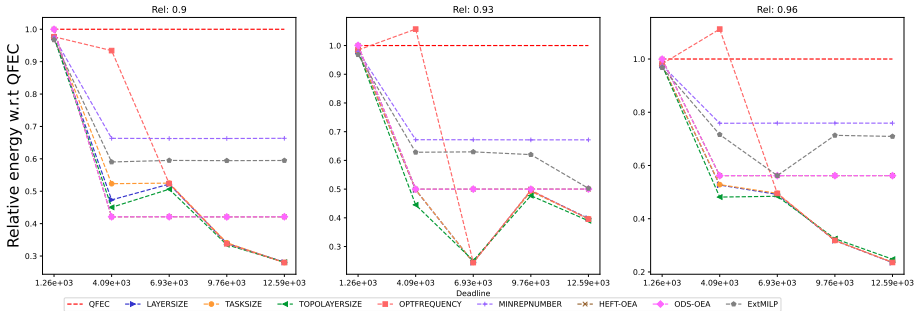

Figure 1350: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

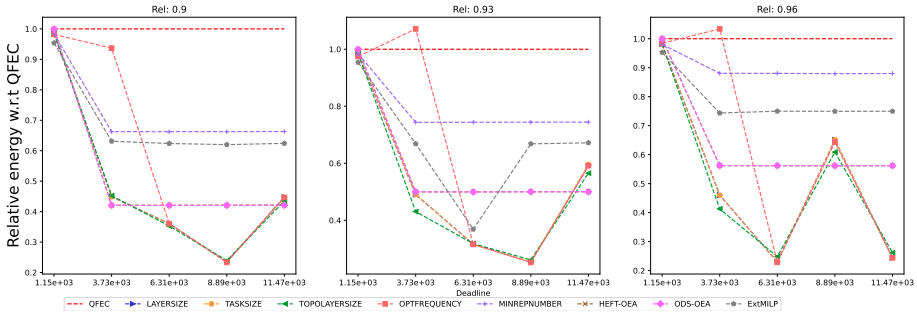

Figure 1351: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

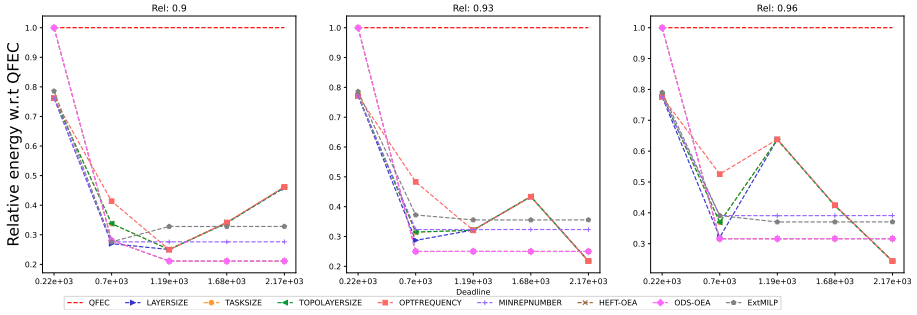

Figure 1352: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

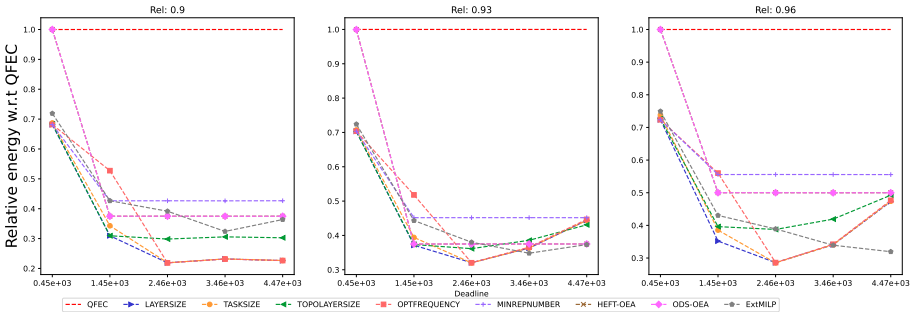

Figure 1353: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

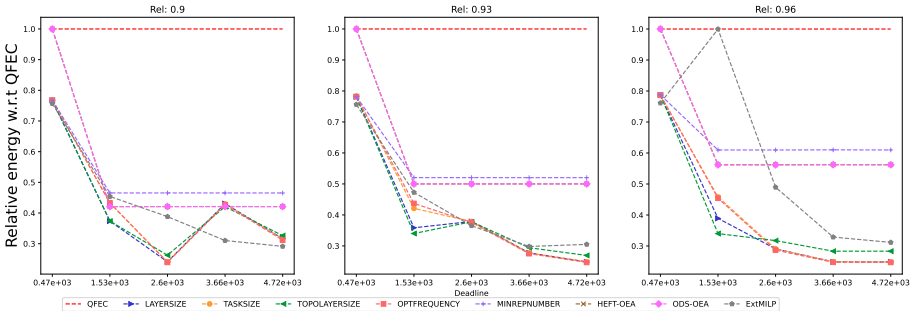

Figure 1354: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

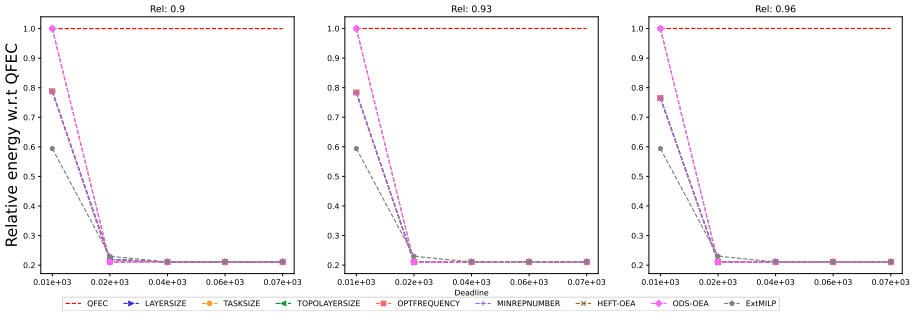

Figure 1355: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

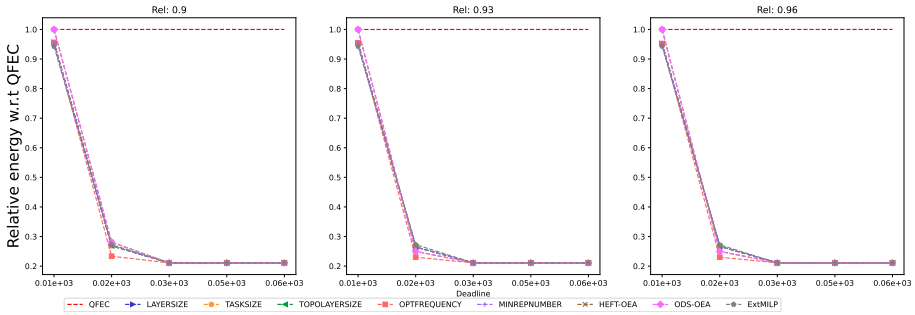

Figure 1356: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

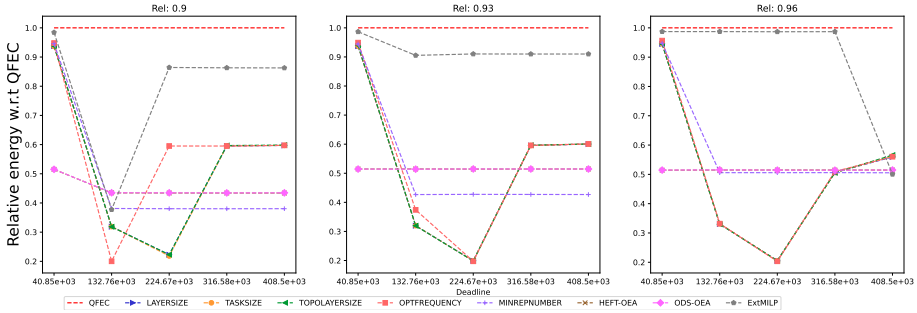

Figure 1357: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

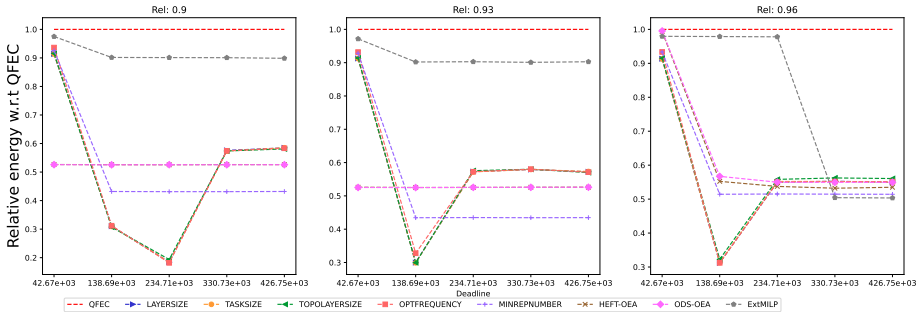

Figure 1358: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

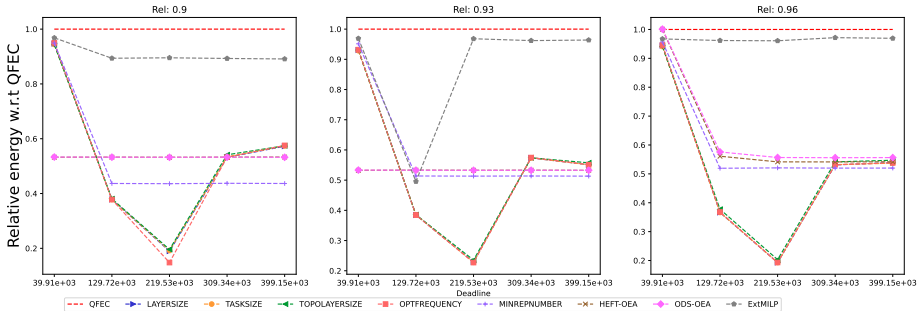

Figure 1359: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

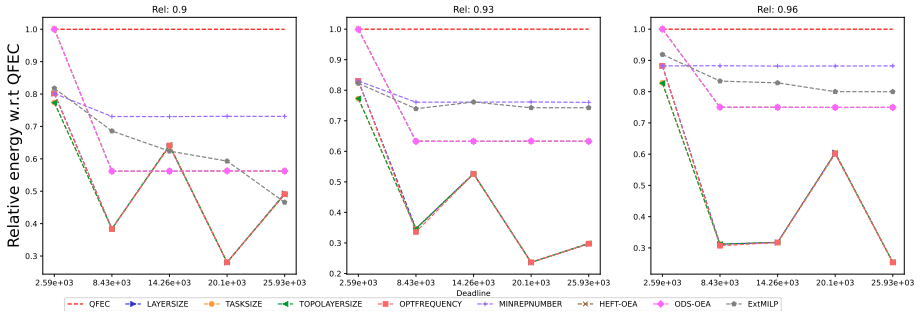

Figure 1360: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

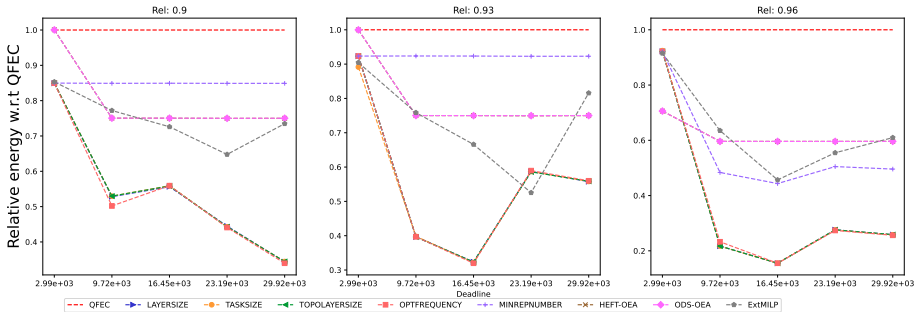

Figure 1361: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.8 $BC/WC = 0.8$

F.8.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

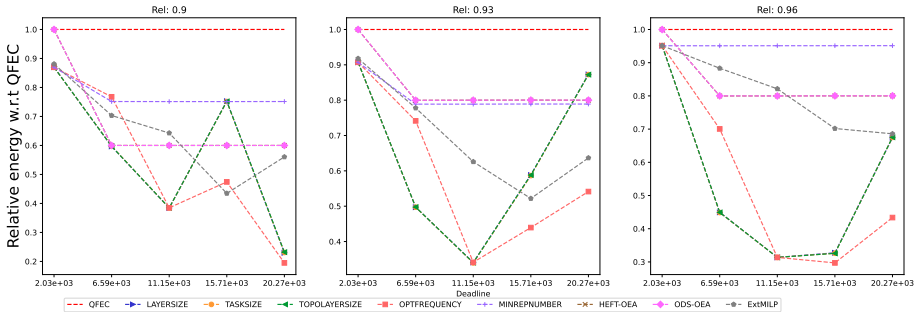

Figure 1362: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

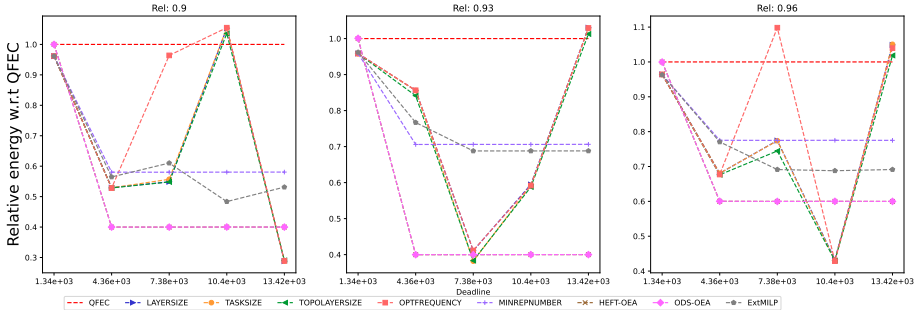

Figure 1363: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

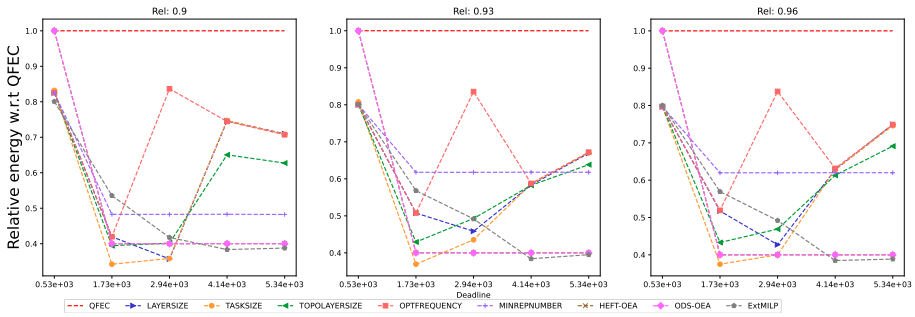

Figure 1364: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

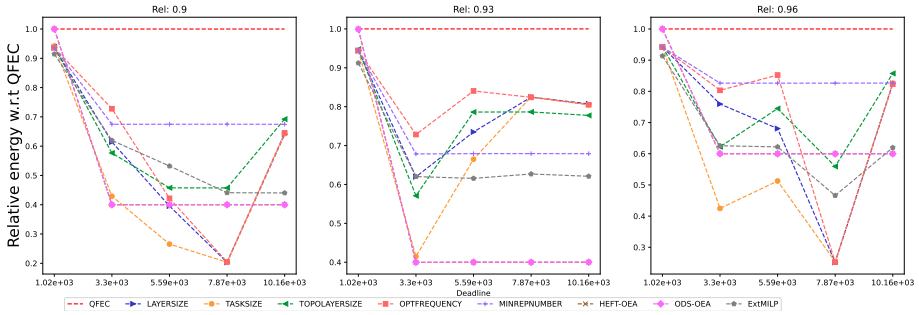

Figure 1365: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

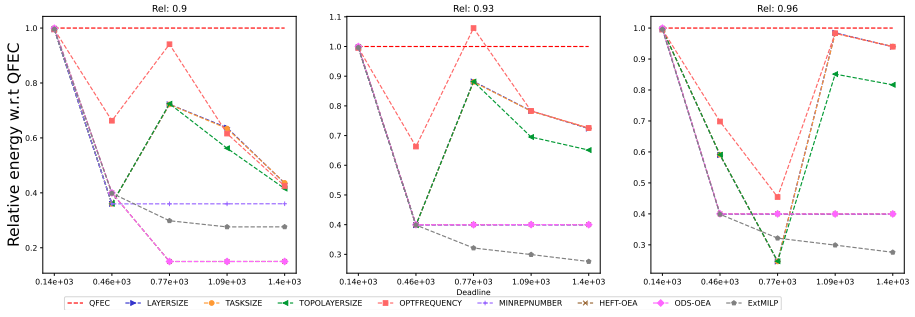

Figure 1366: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

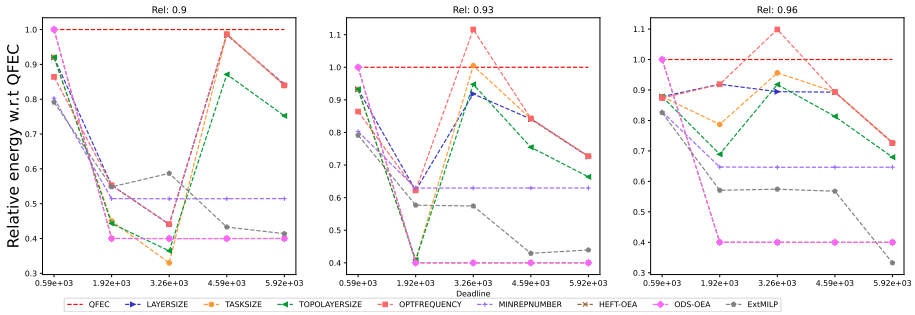

Figure 1367: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

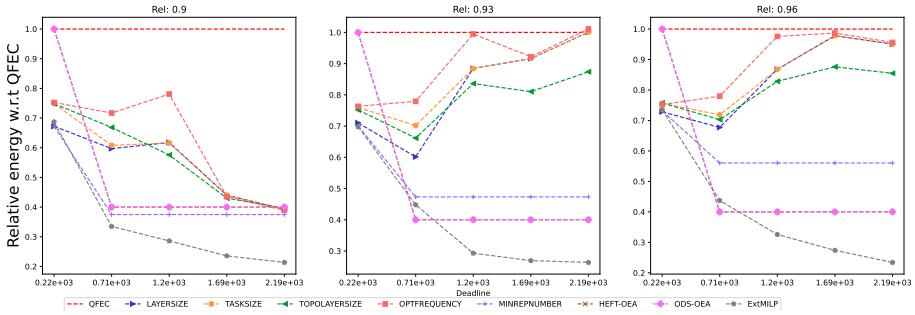

Figure 1368: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

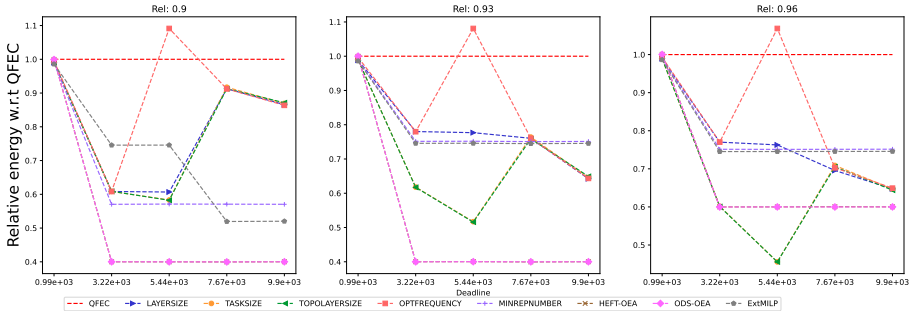

Figure 1369: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

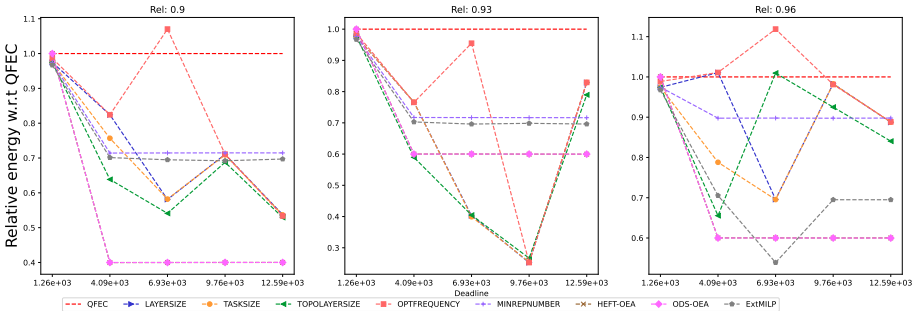

Figure 1370: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

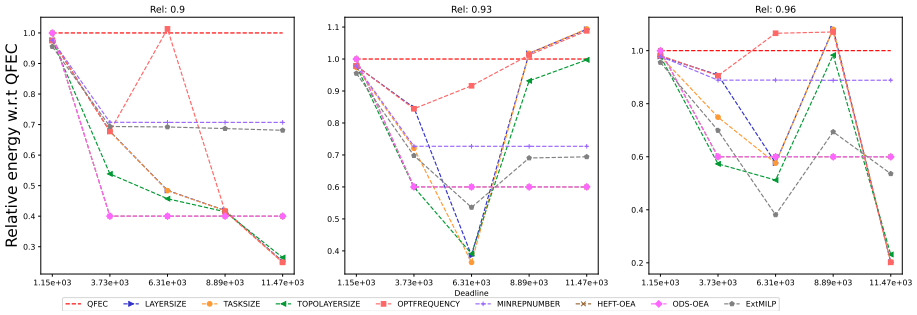

Figure 1371: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

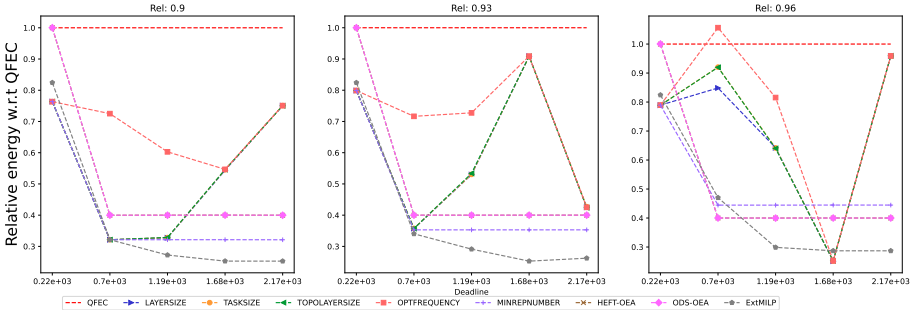

Figure 1372: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

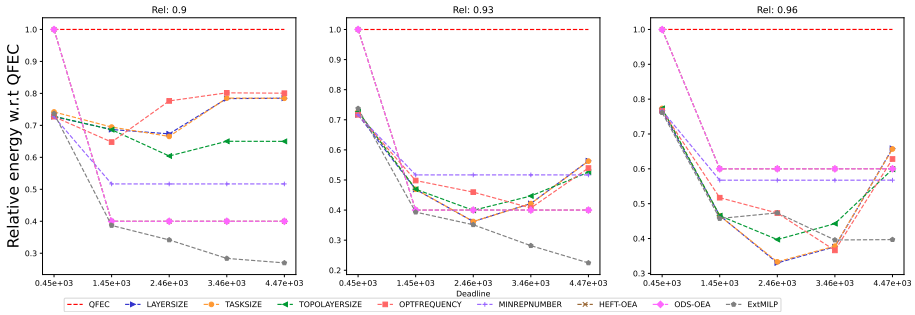

Figure 1373: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

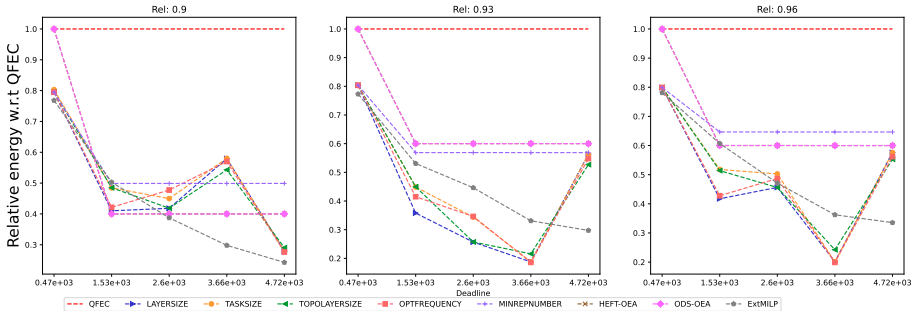

Figure 1374: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

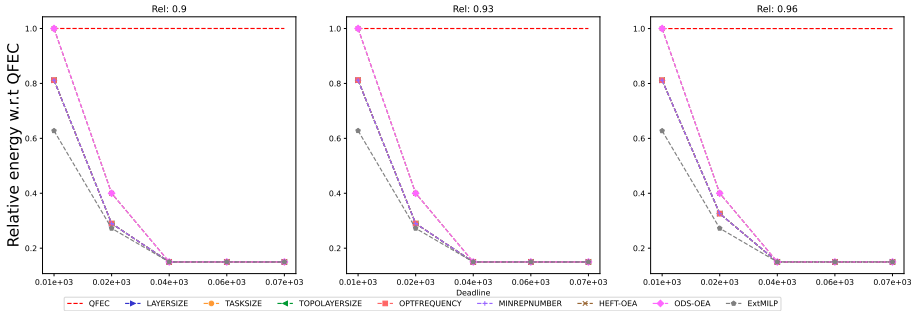

Figure 1375: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

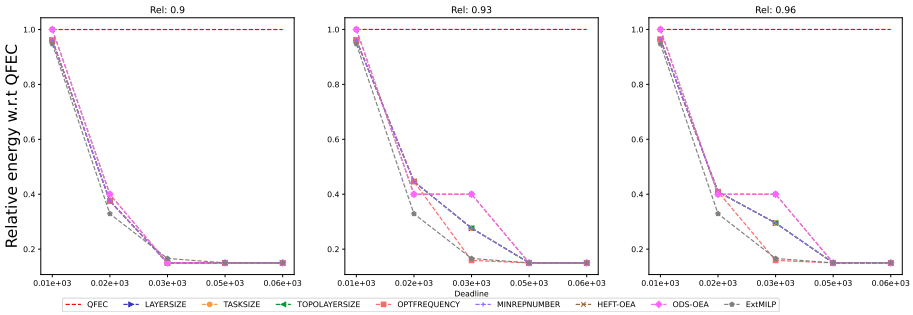

Figure 1376: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

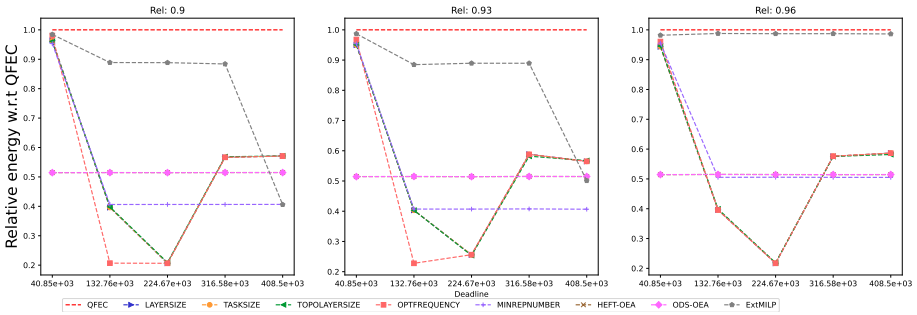

Figure 1377: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

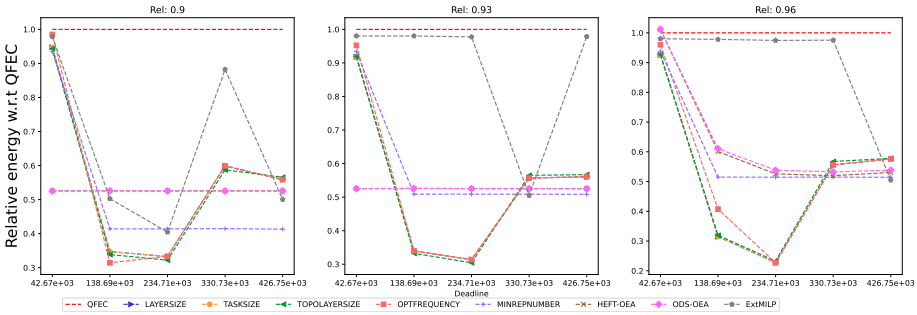

Figure 1378: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

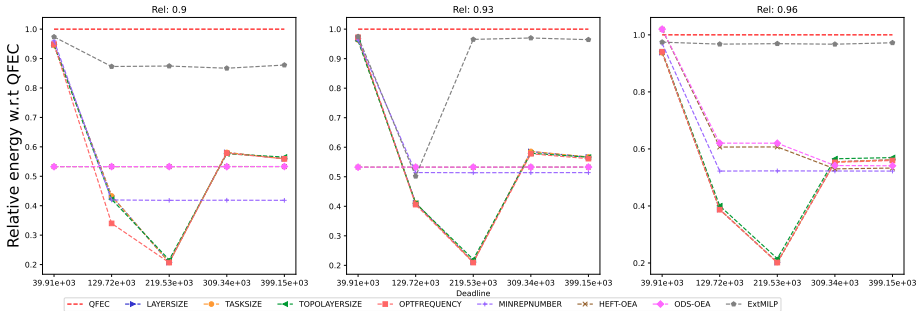

Figure 1379: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

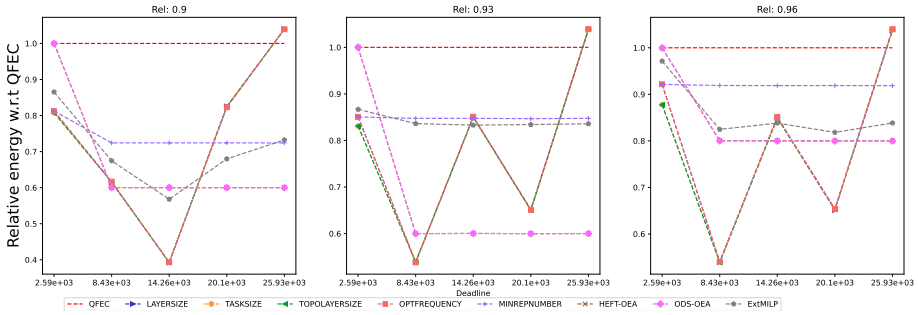

Figure 1380: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

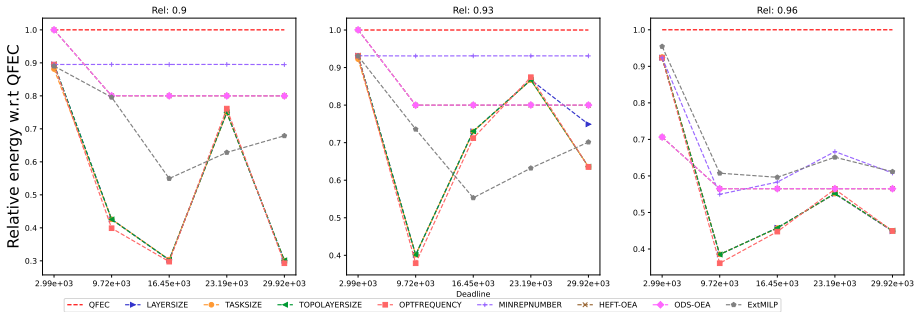

Figure 1381: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.8.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

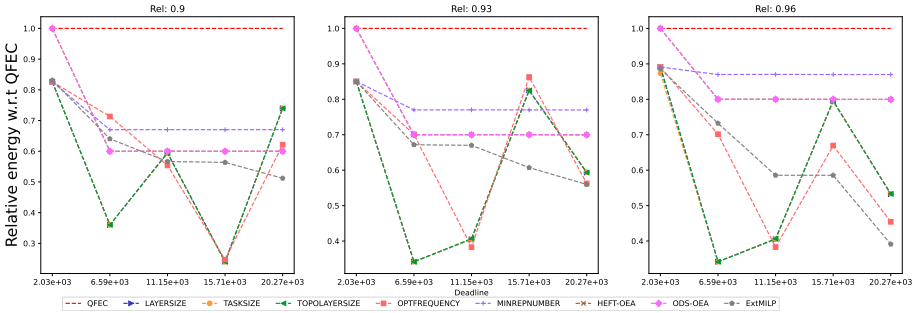

Figure 1382: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

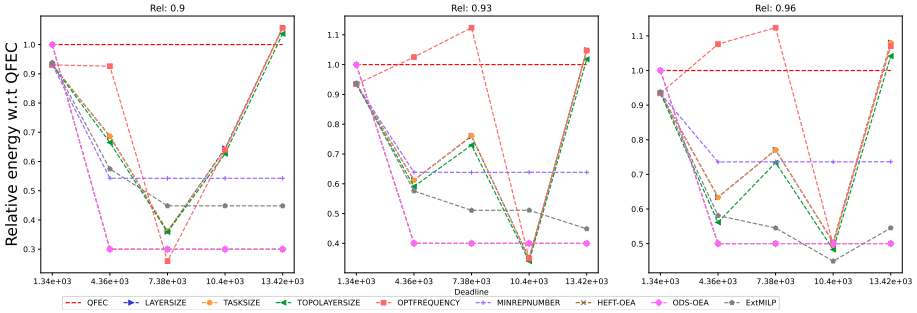

Figure 1383: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

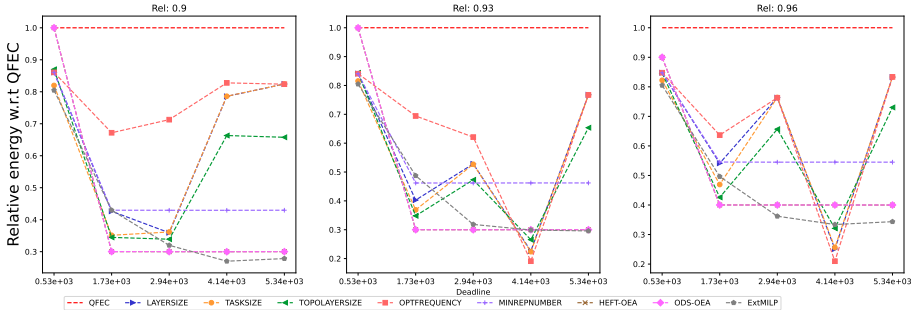

Figure 1384: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

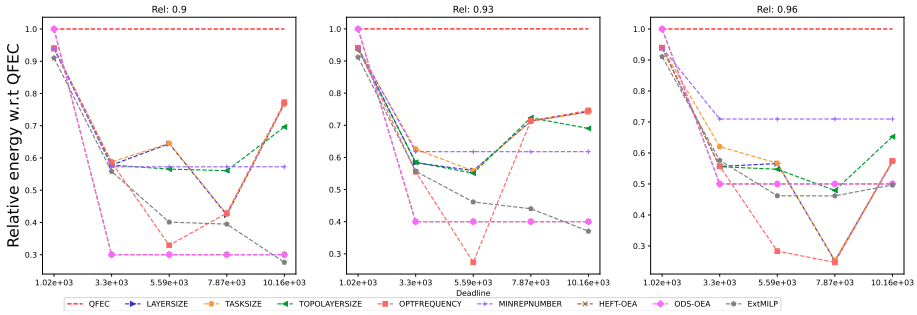

Figure 1385: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

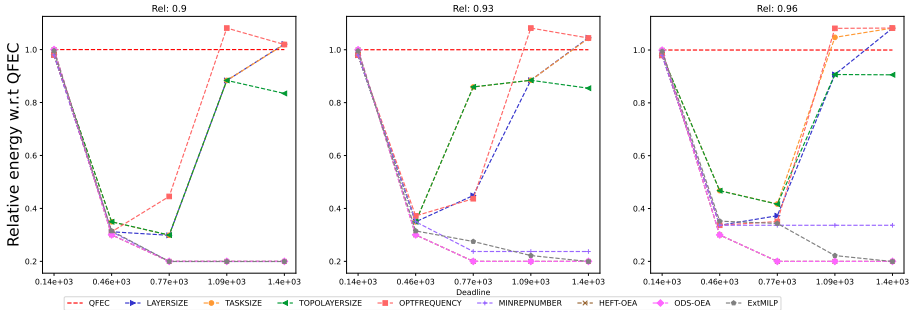

Figure 1386: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

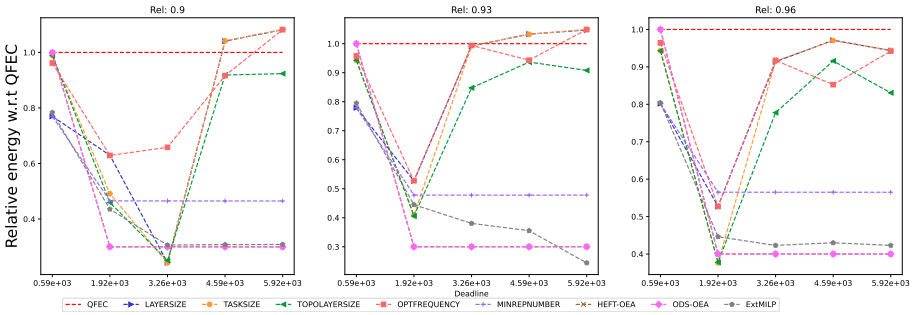

Figure 1387: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

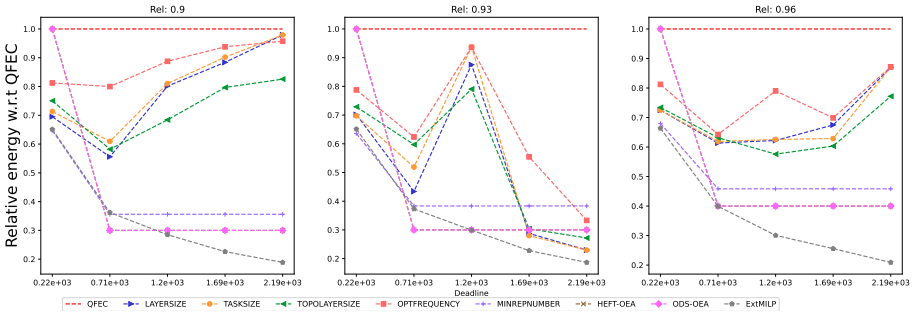

Figure 1388: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

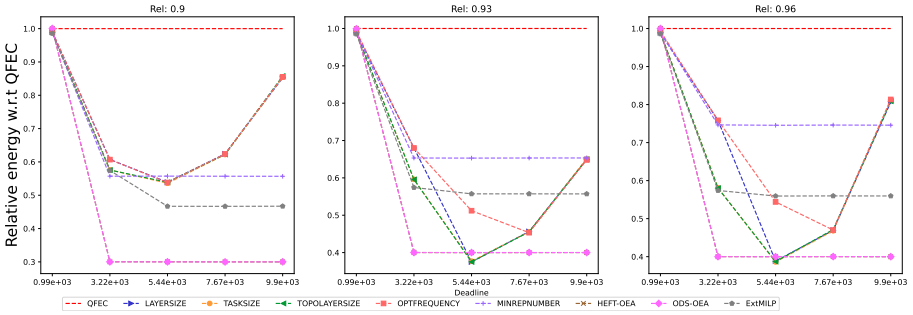

Figure 1389: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

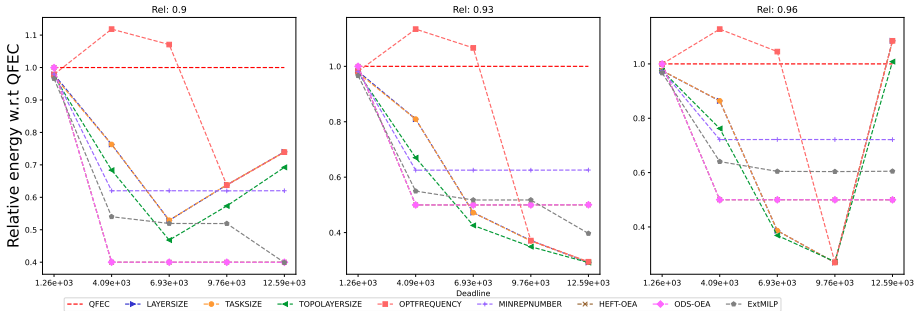

Figure 1390: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

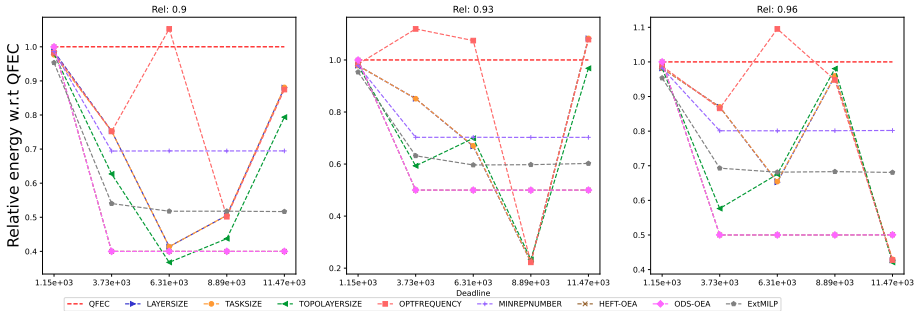

Figure 1391: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

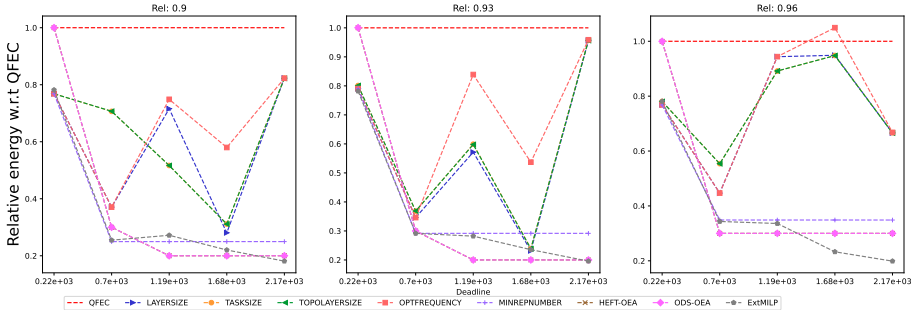

Figure 1392: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

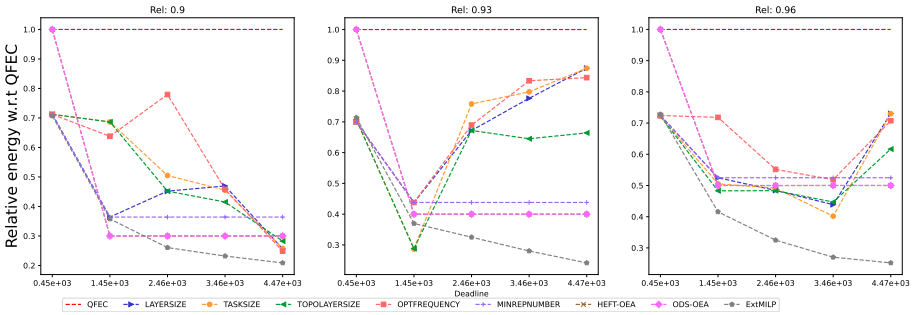

Figure 1393: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

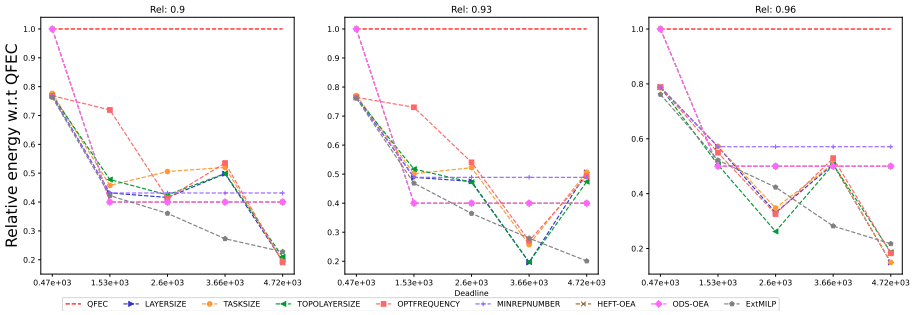

Figure 1394: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

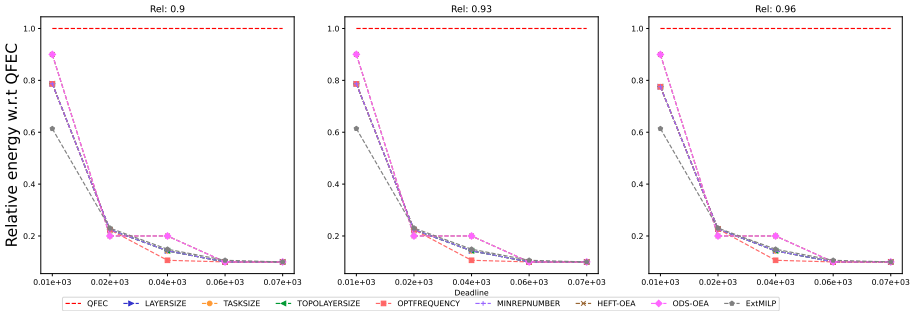

Figure 1395: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

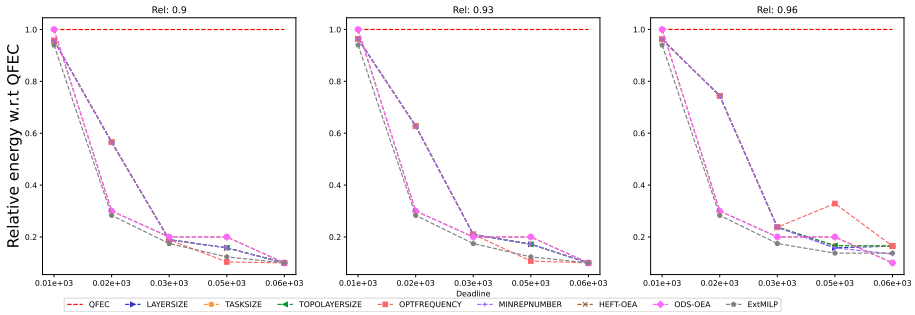

Figure 1396: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

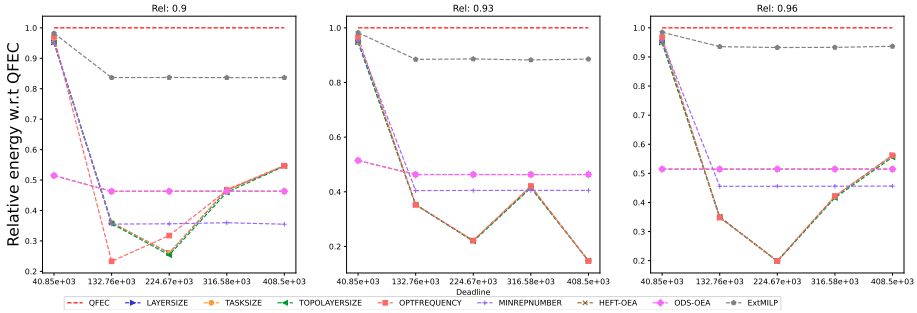

Figure 1397: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

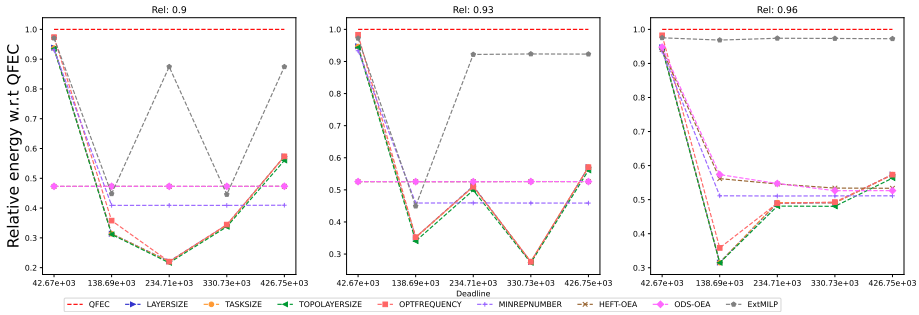

Figure 1398: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

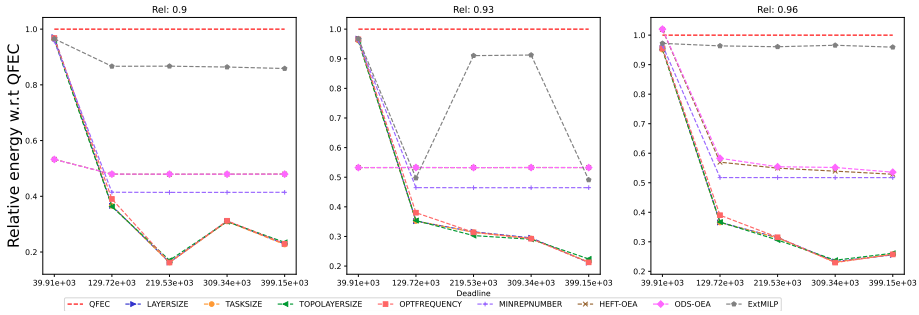

Figure 1399: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

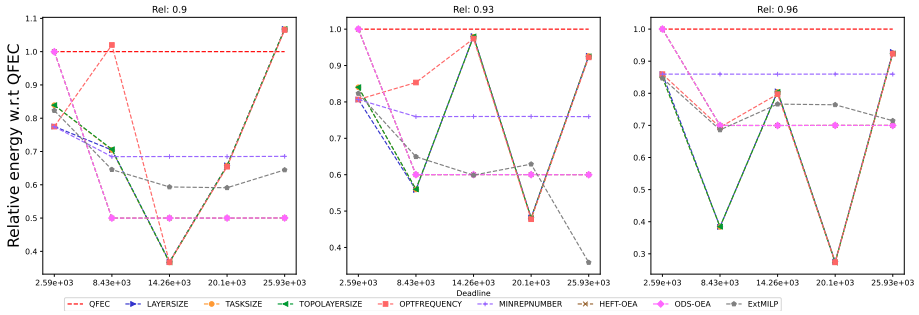

Figure 1400: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

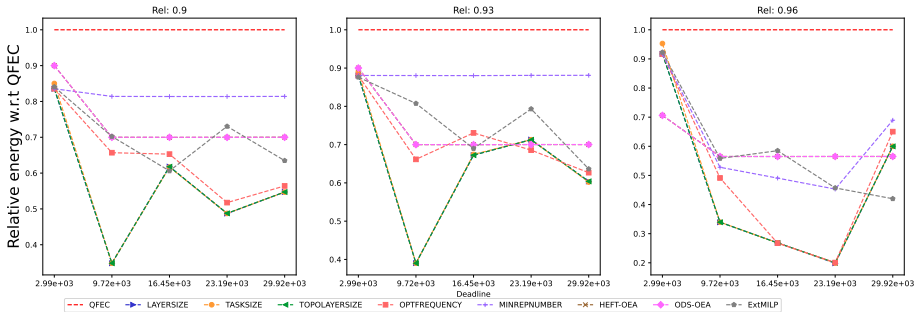

Figure 1401: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.8.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

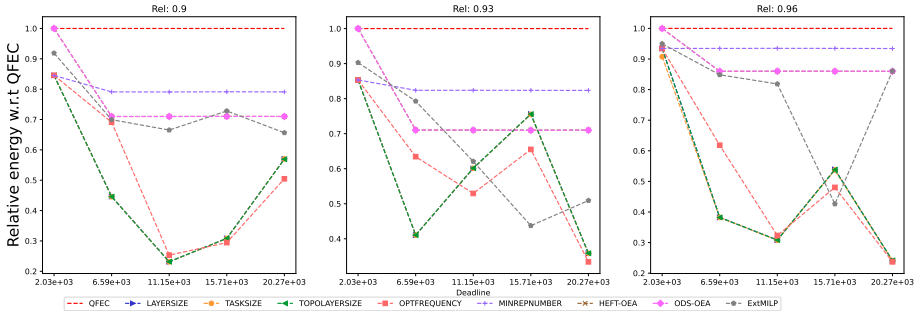

Figure 1402: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

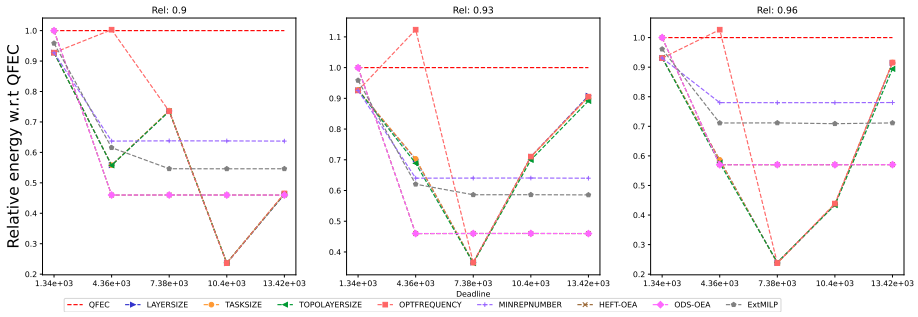

Figure 1403: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

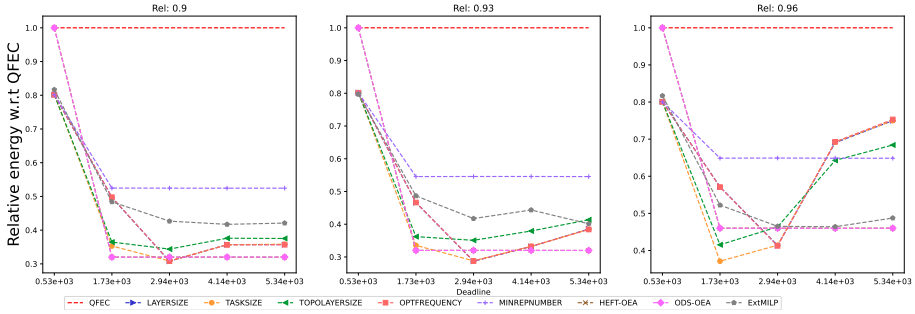

Figure 1404: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

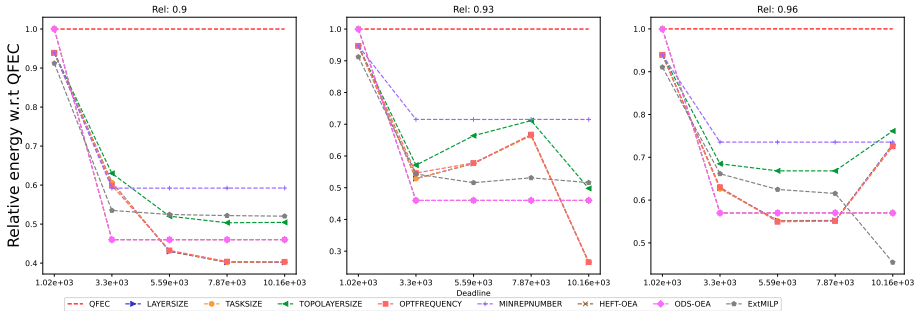

Figure 1405: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

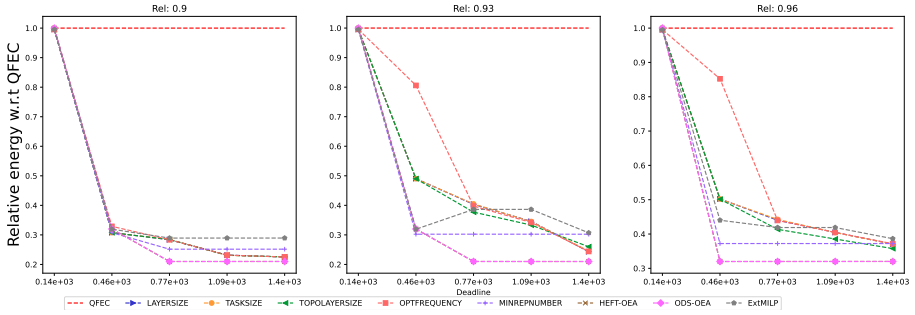

Figure 1406: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

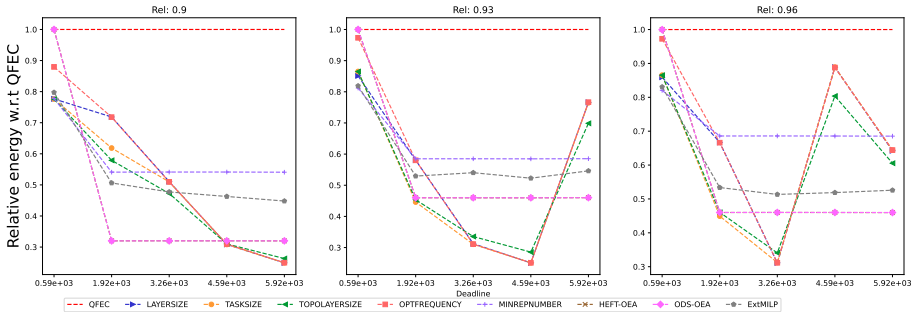

Figure 1407: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

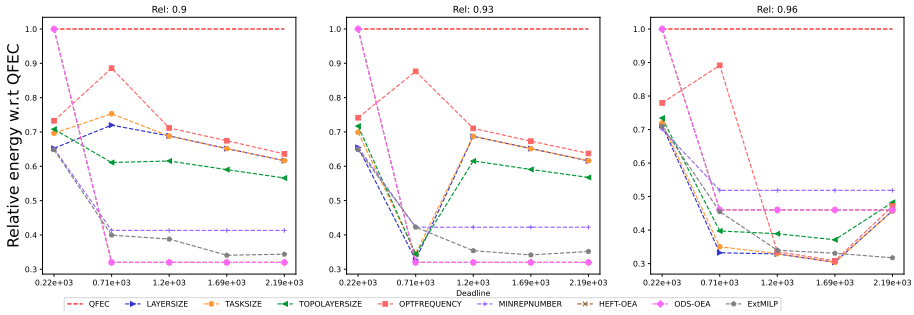

Figure 1408: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

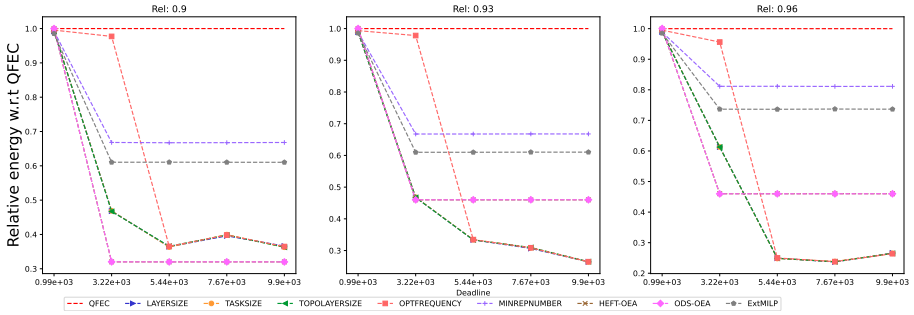

Figure 1409: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

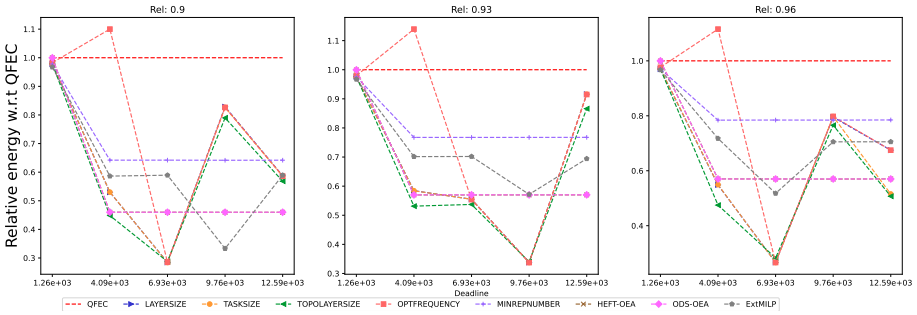

Figure 1410: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

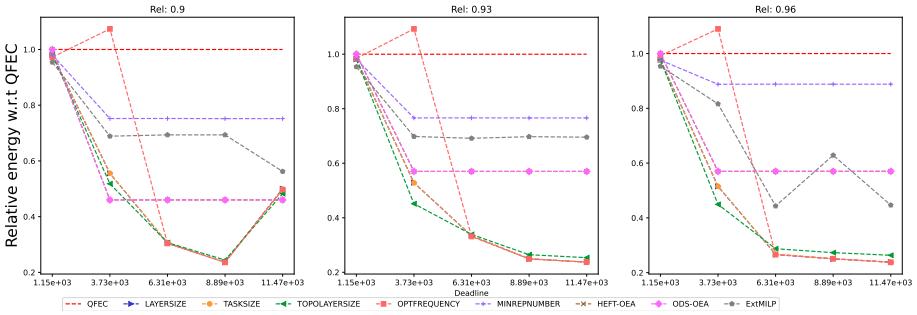

Figure 1411: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

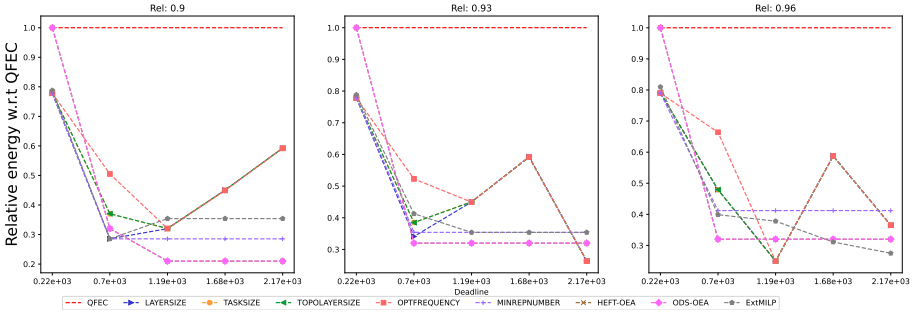

Figure 1412: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

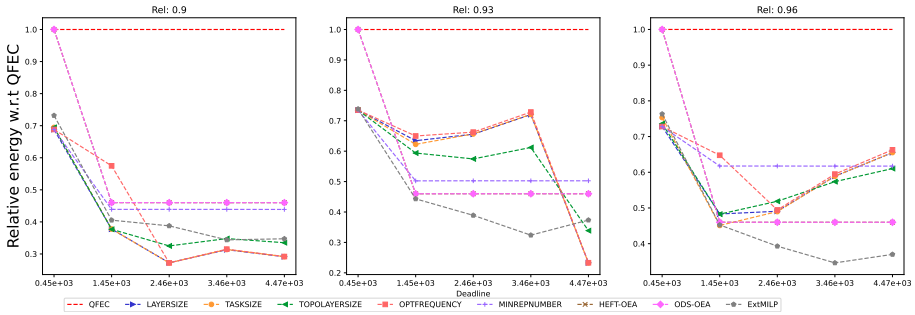

Figure 1413: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

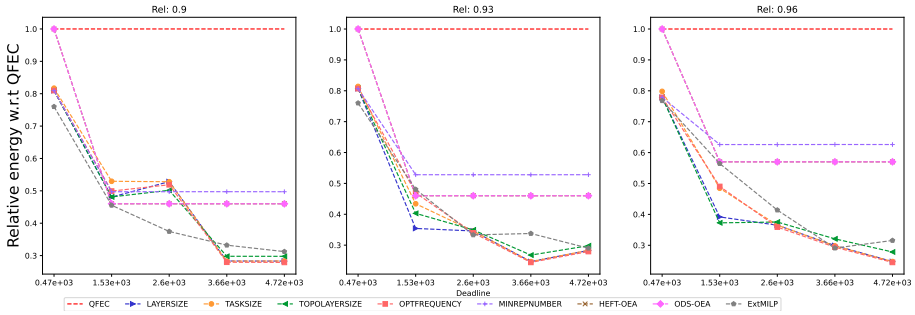

Figure 1414: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

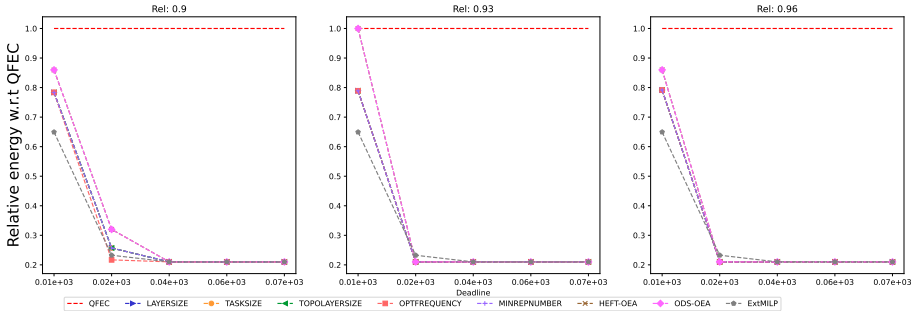

Figure 1415: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

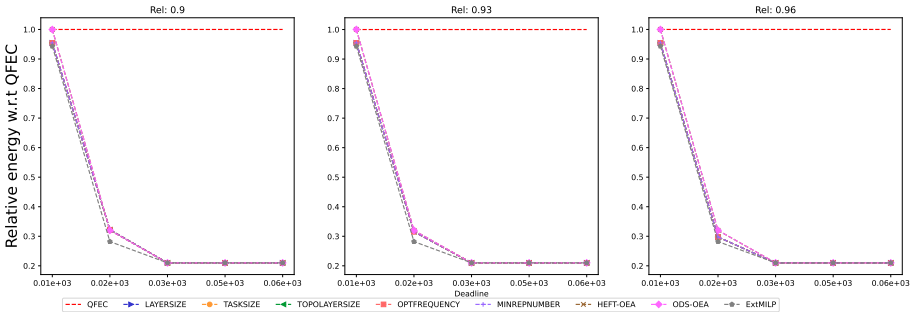

Figure 1416: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

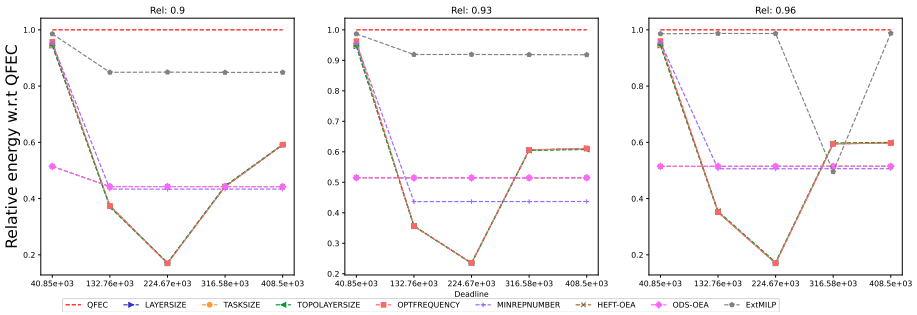

Figure 1417: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

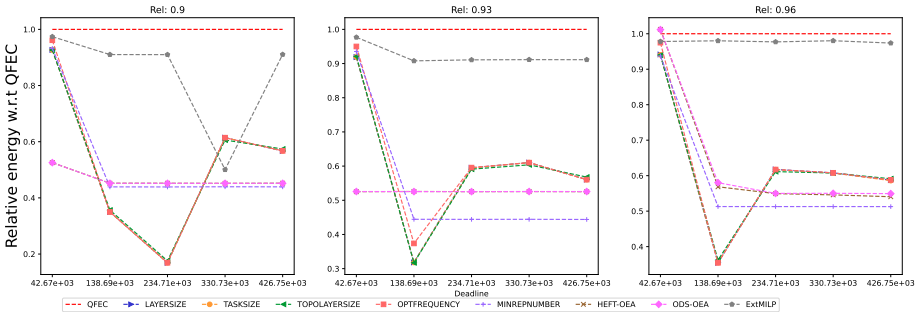

Figure 1418: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

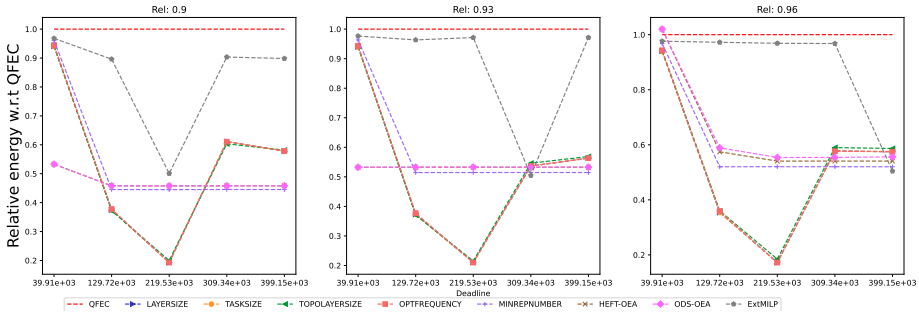

Figure 1419: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

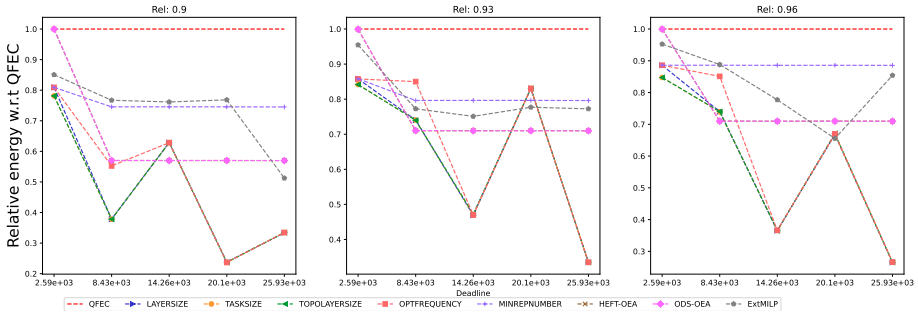

Figure 1420: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

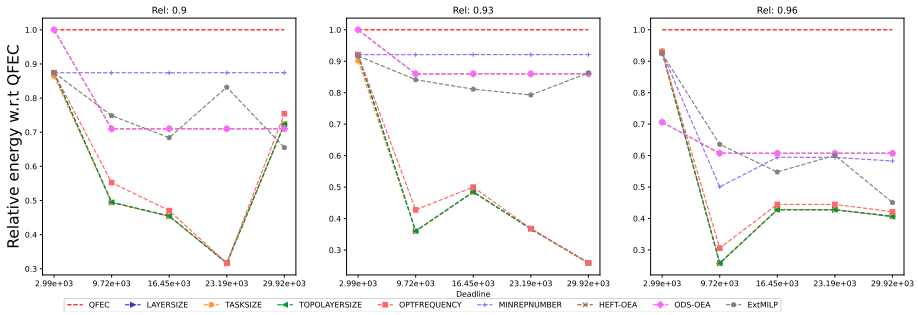

Figure 1421: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.8.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

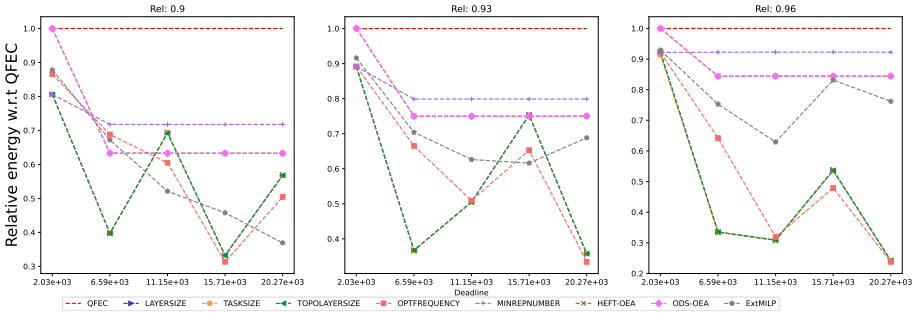

Figure 1422: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

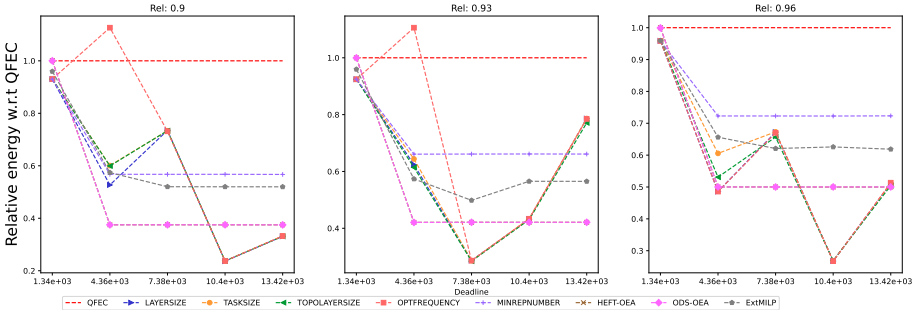

Figure 1423: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

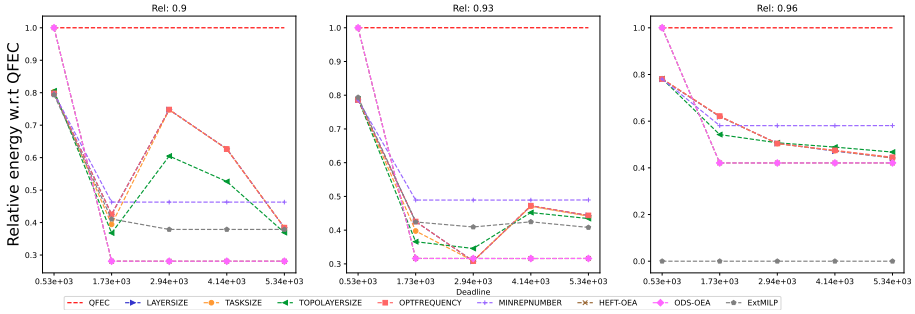

Figure 1424: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

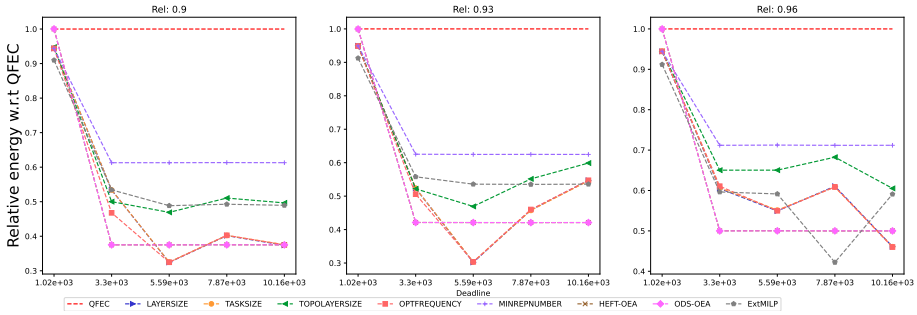

Figure 1425: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

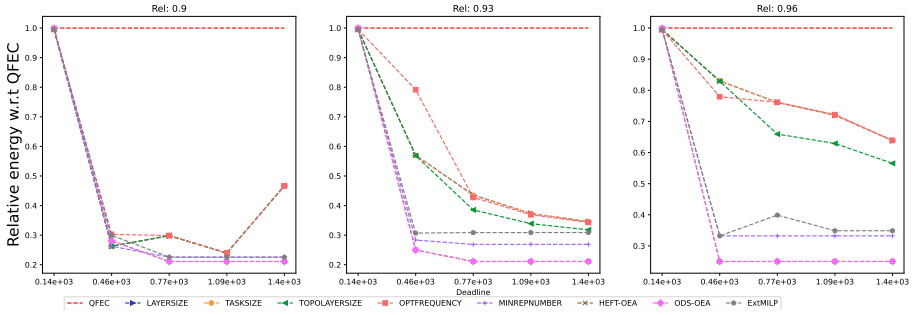

Figure 1426: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

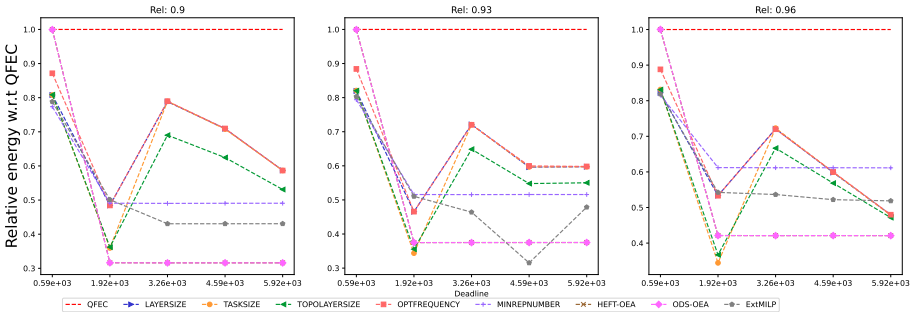

Figure 1427: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

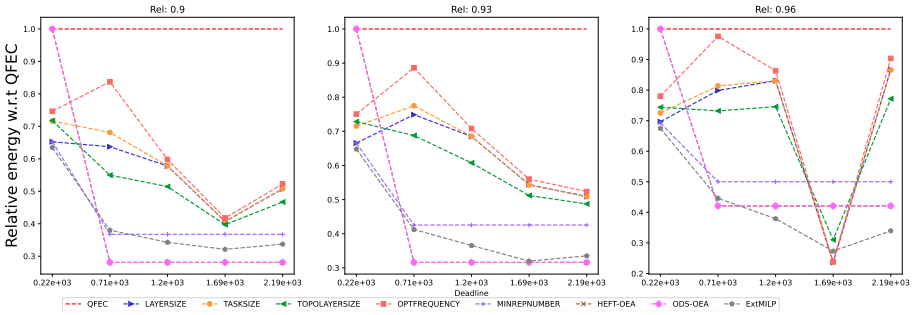

Figure 1428: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

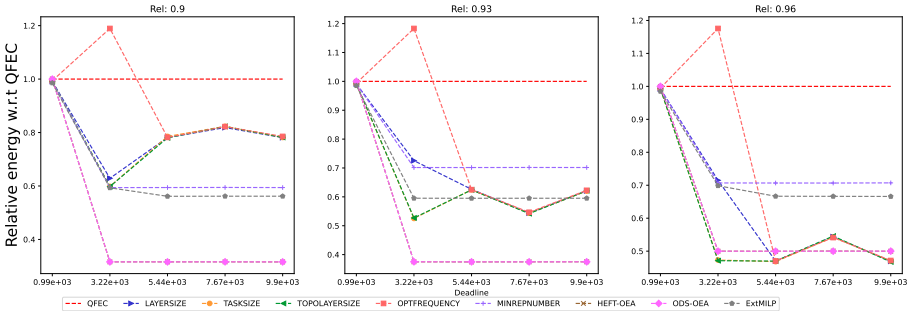

Figure 1429: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

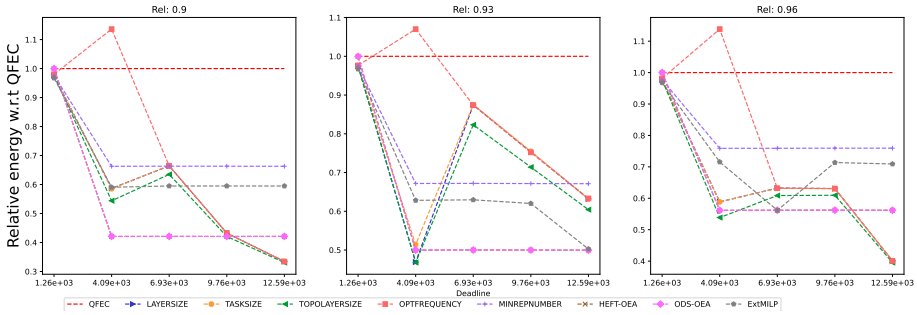

Figure 1430: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

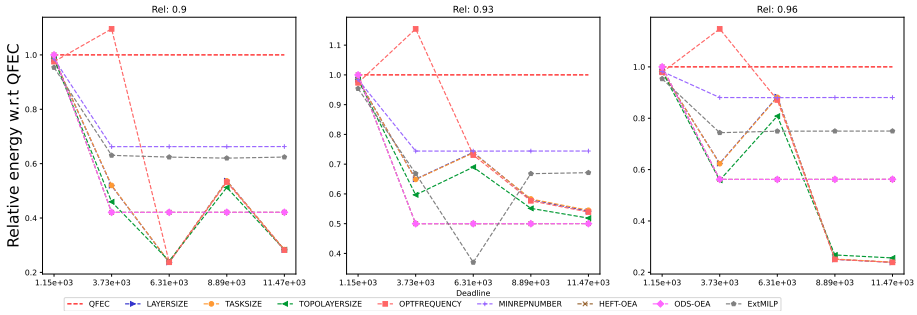

Figure 1431: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

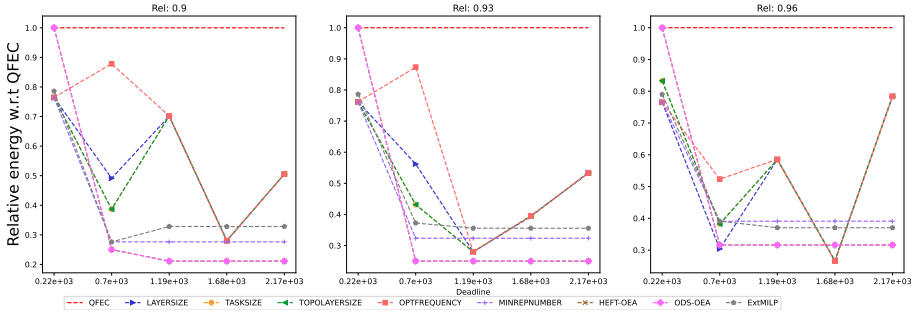

Figure 1432: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

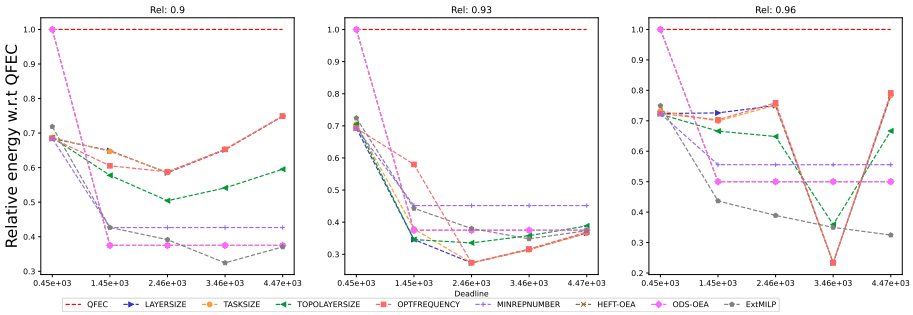

Figure 1433: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

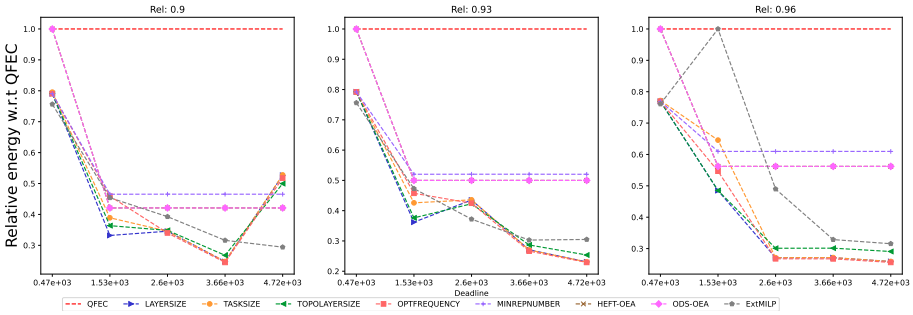

Figure 1434: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

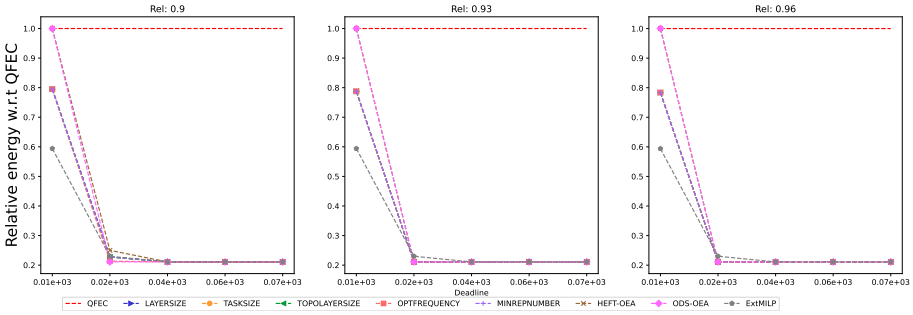

Figure 1435: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

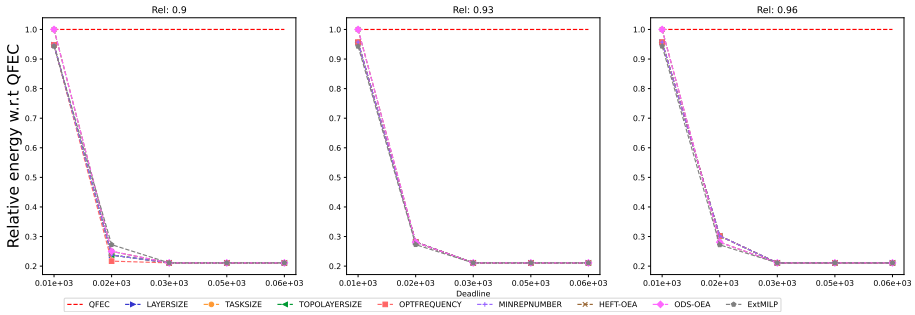

Figure 1436: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

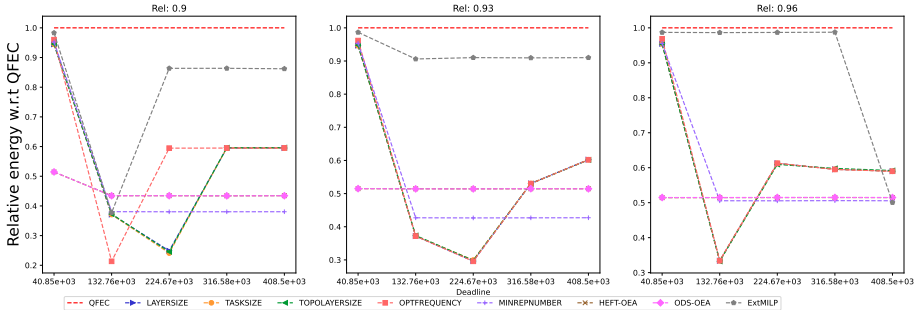

Figure 1437: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

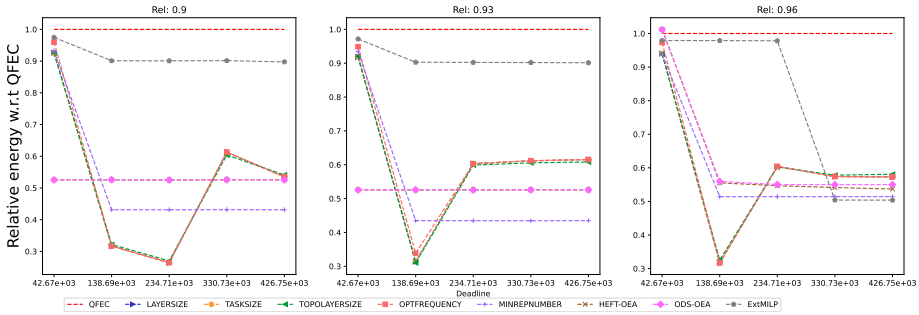

Figure 1438: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

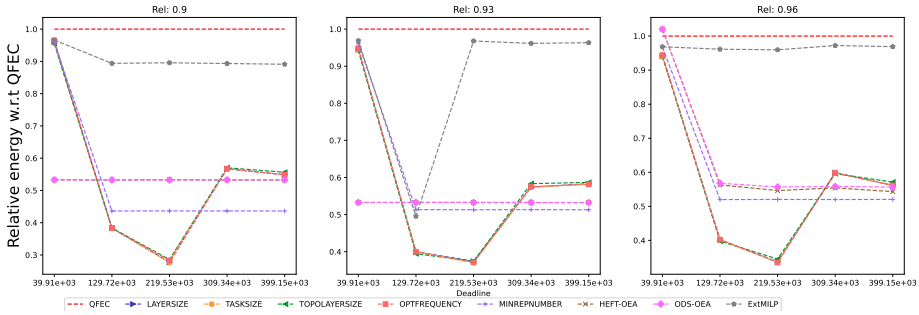

Figure 1439: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

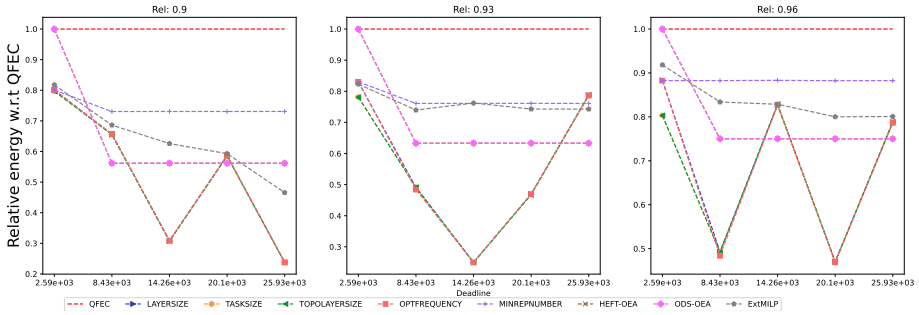

Figure 1440: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

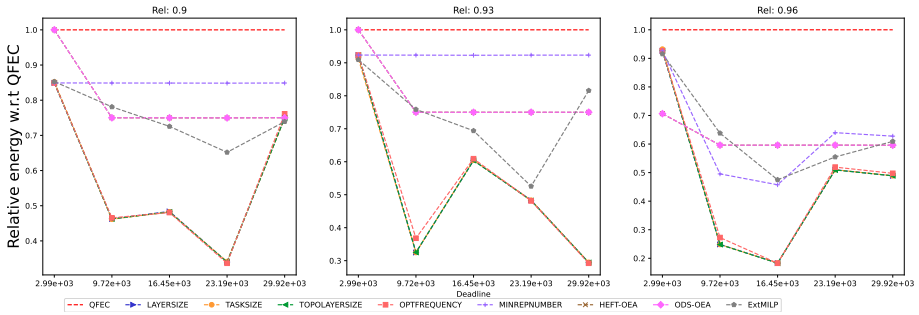

Figure 1441: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.9 $BC/WC = 0.9$

F.9.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

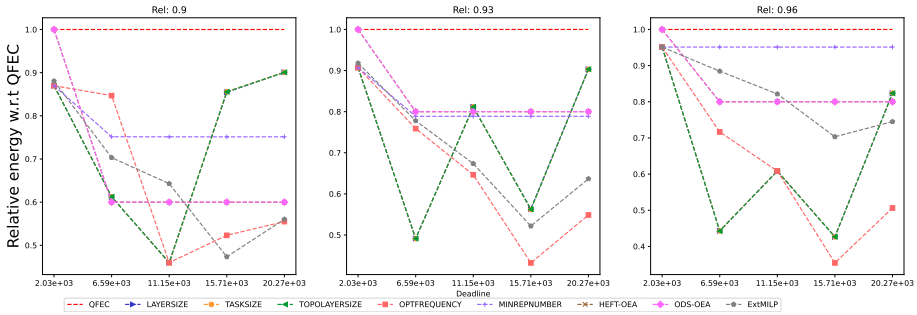

Figure 1442: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

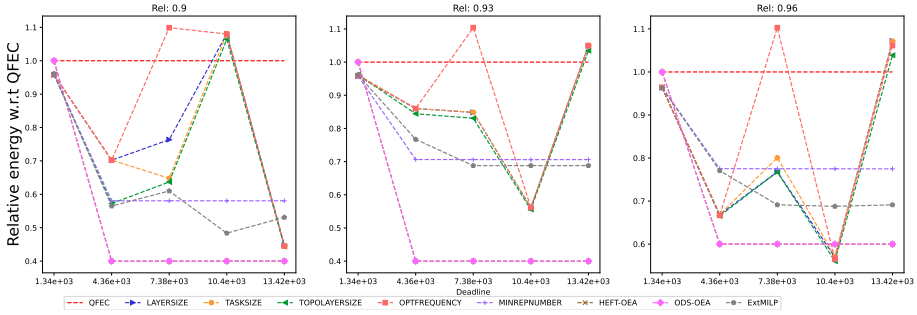

Figure 1443: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

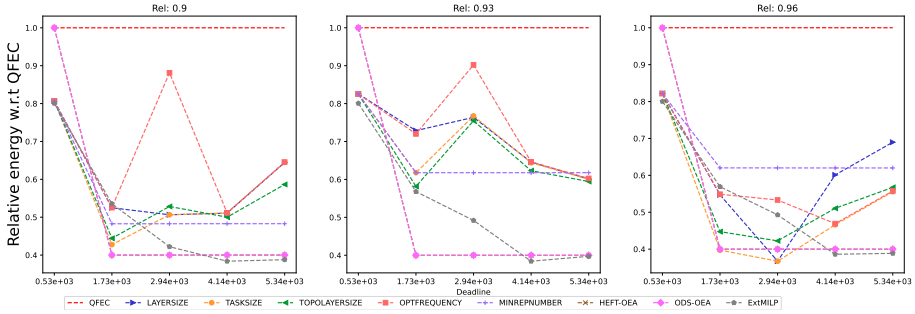

Figure 1444: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

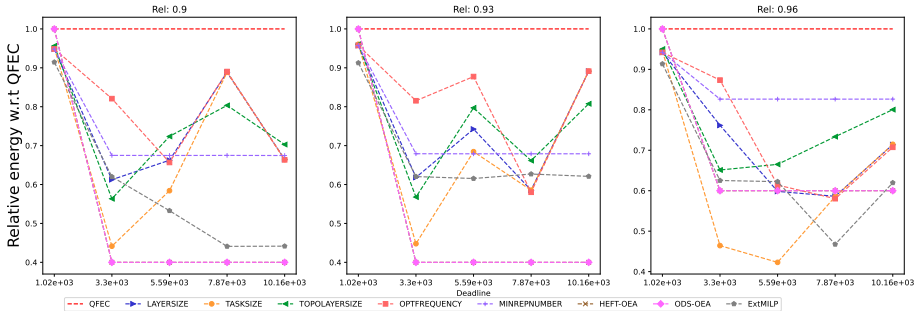

Figure 1445: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

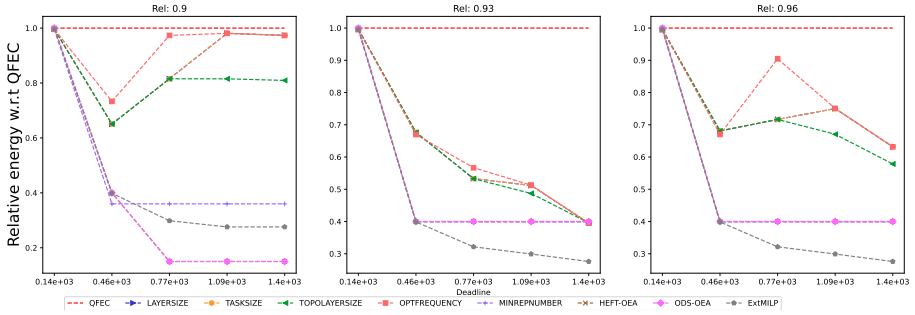

Figure 1446: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

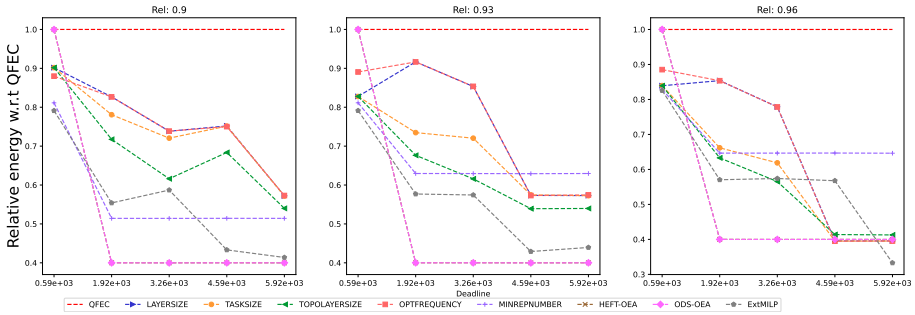

Figure 1447: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

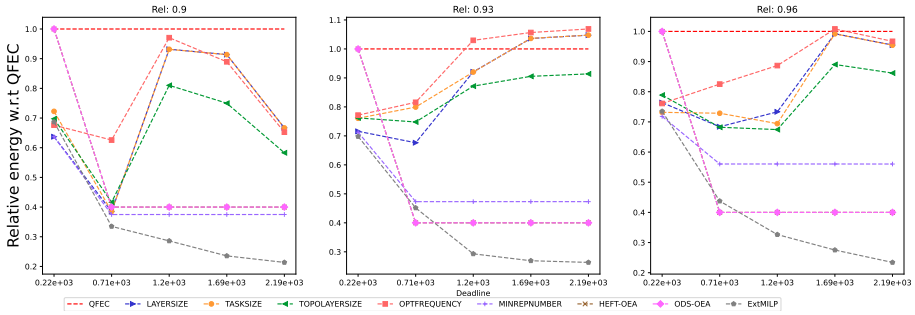

Figure 1448: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

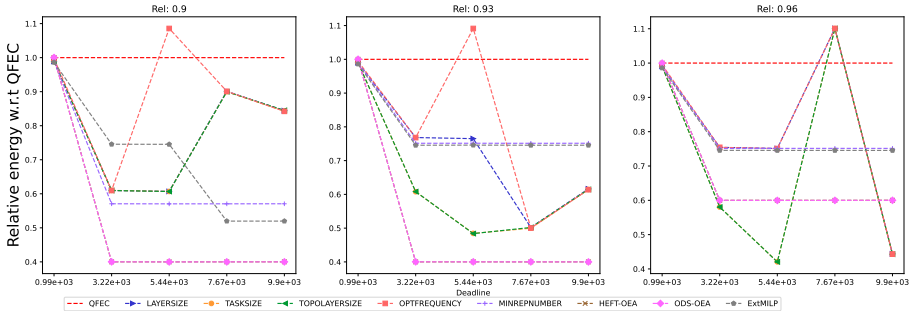

Figure 1449: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

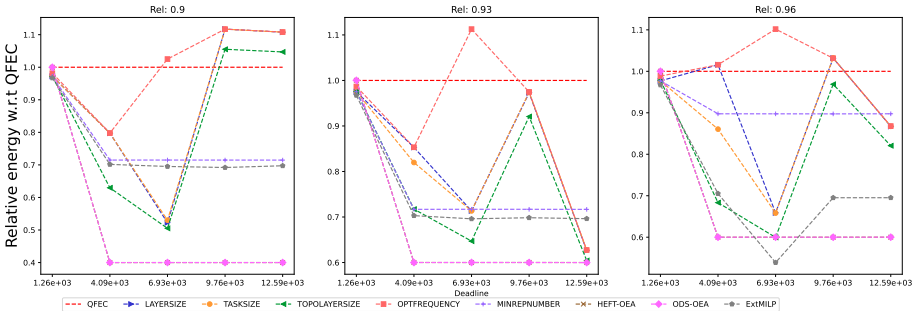

Figure 1450: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

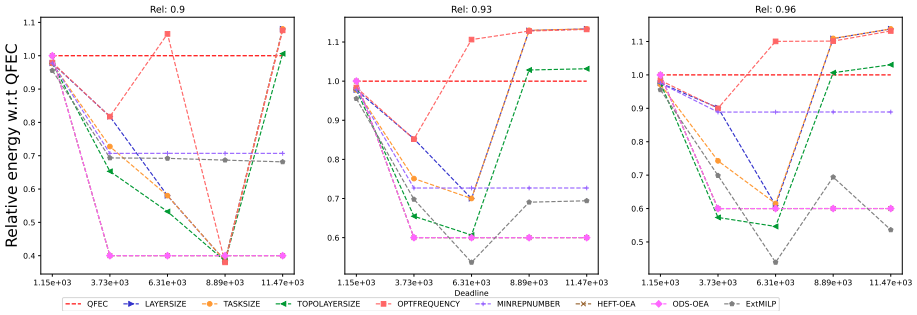

Figure 1451: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

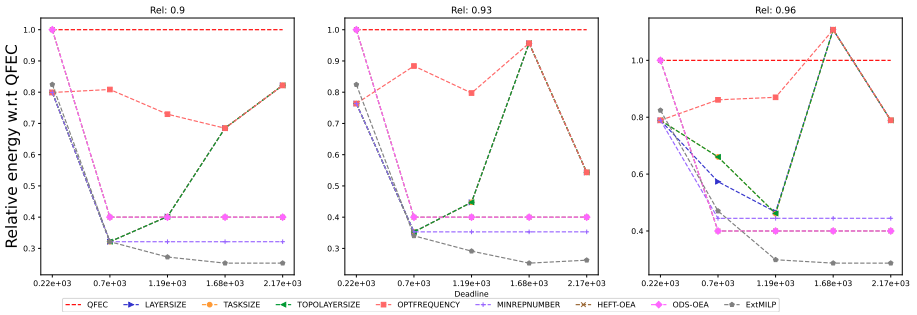

Figure 1452: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

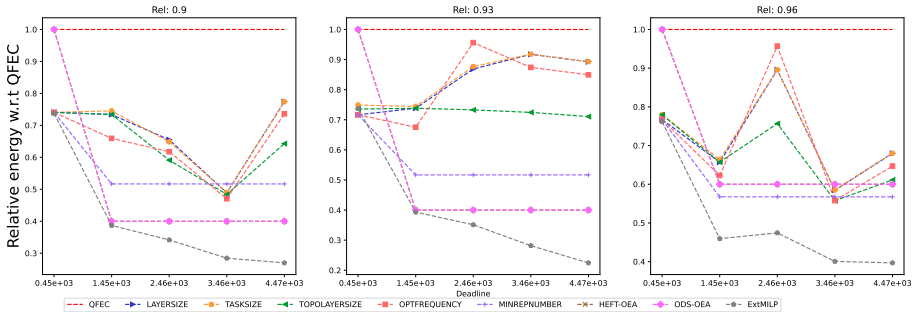

Figure 1453: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

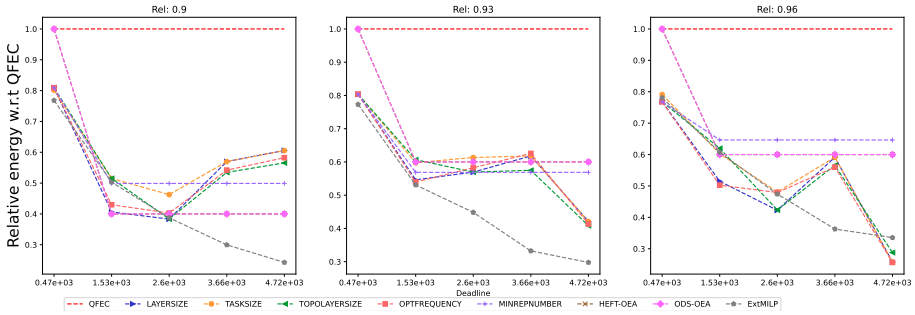

Figure 1454: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

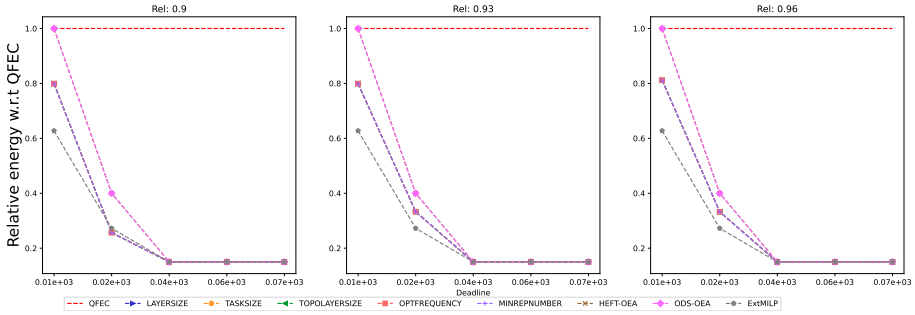

Figure 1455: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

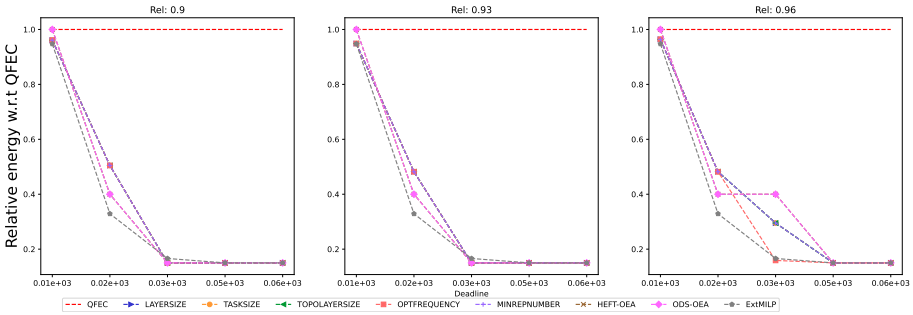

Figure 1456: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

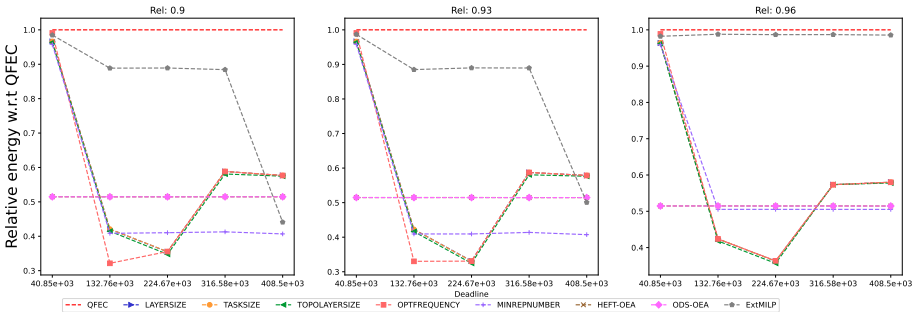

Figure 1457: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

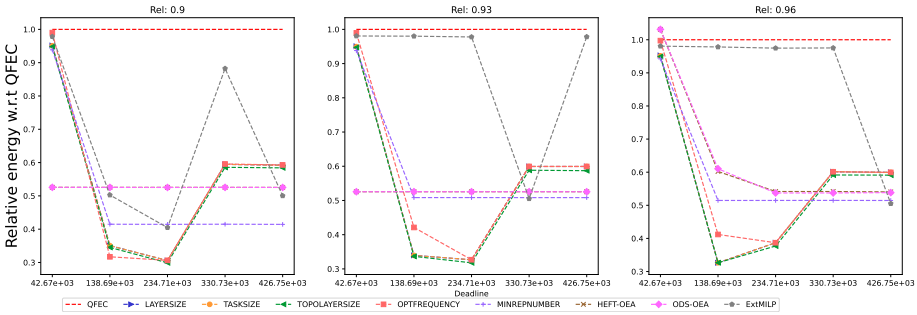

Figure 1458: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

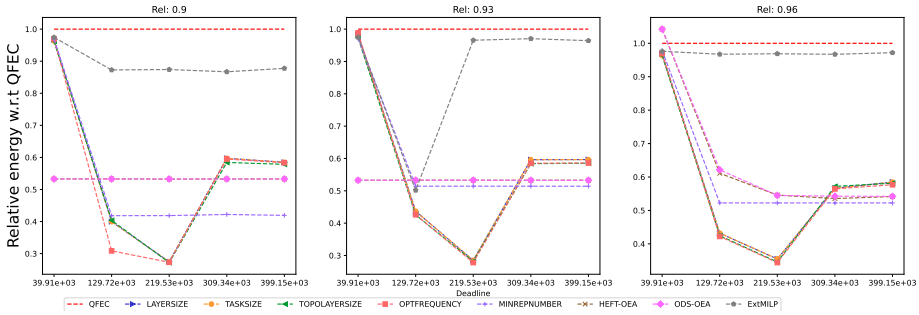

Figure 1459: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

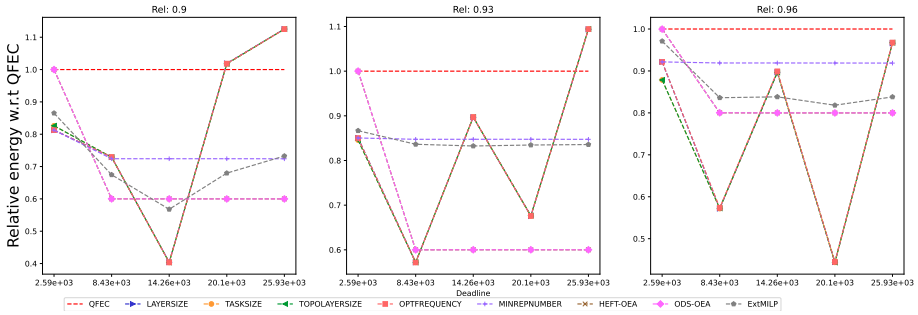

Figure 1460: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

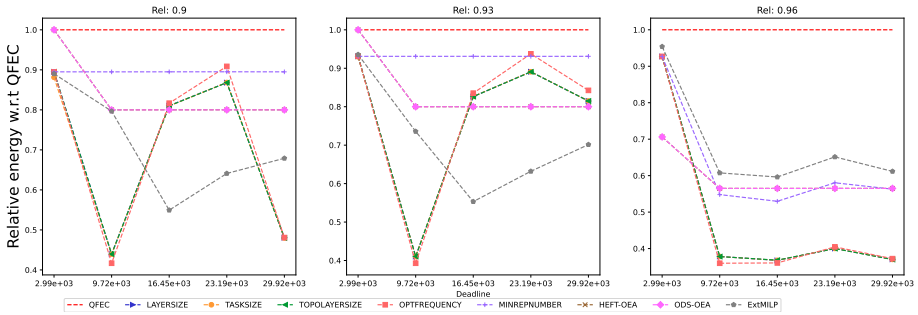

Figure 1461: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.9.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

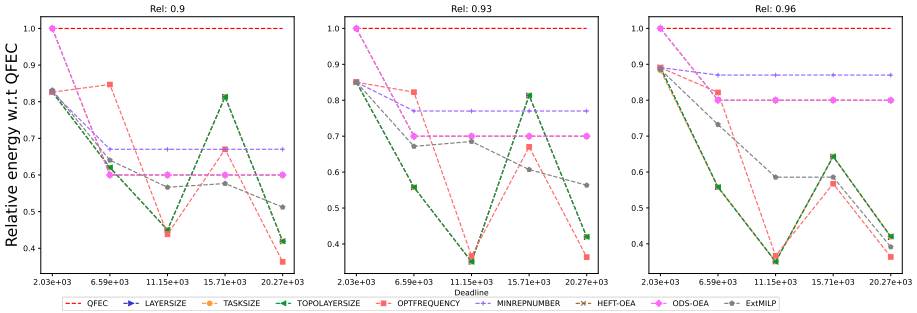

Figure 1462: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

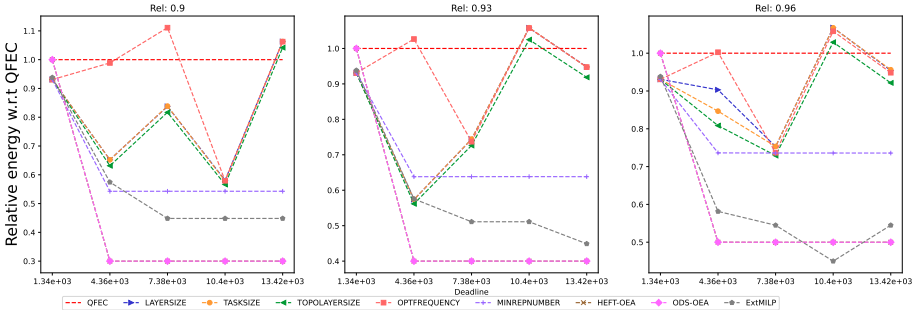

Figure 1463: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

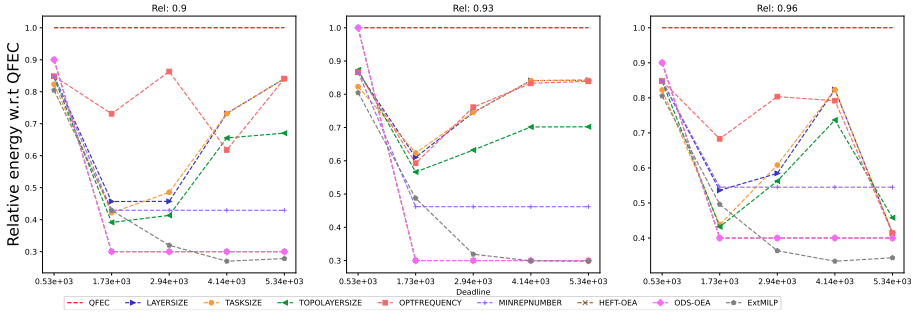

Figure 1464: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

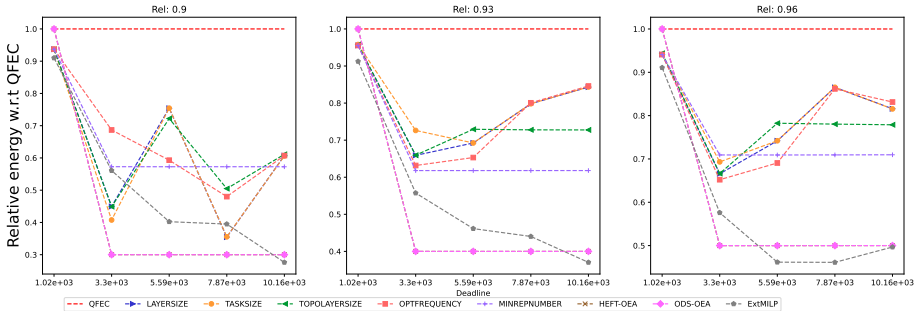

Figure 1465: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

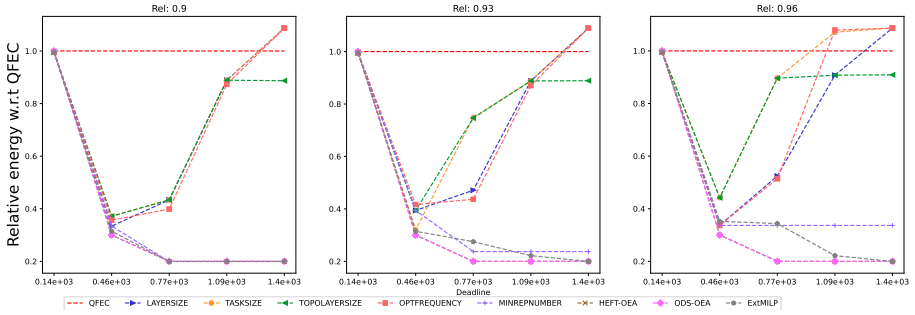

Figure 1466: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

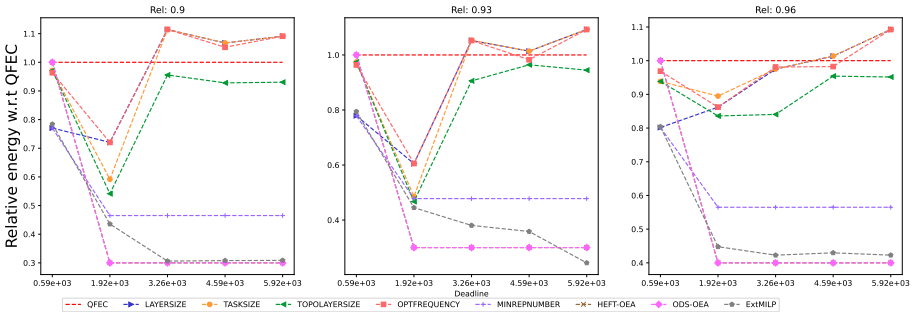

Figure 1467: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

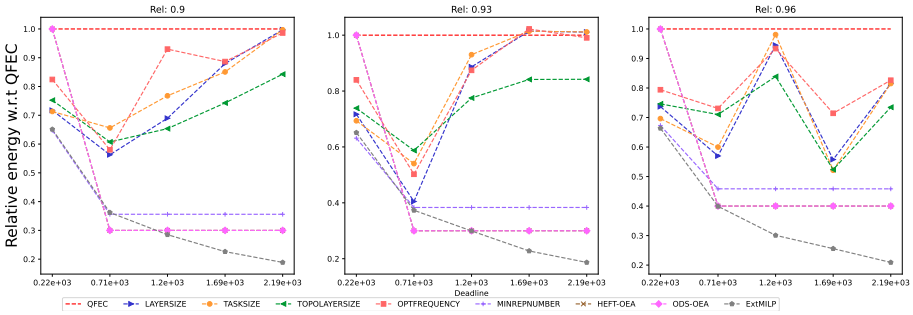

Figure 1468: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

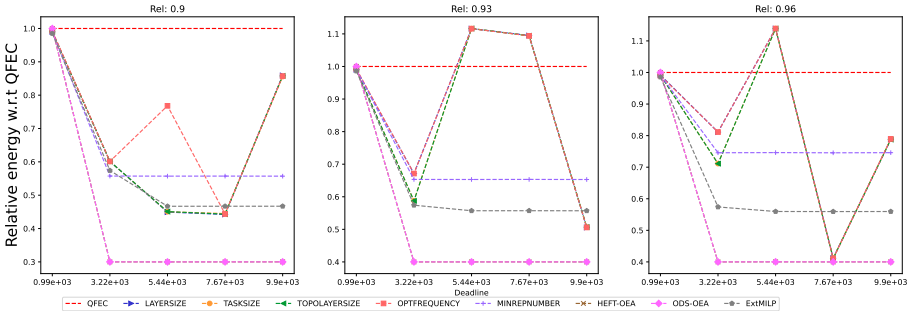

Figure 1469: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

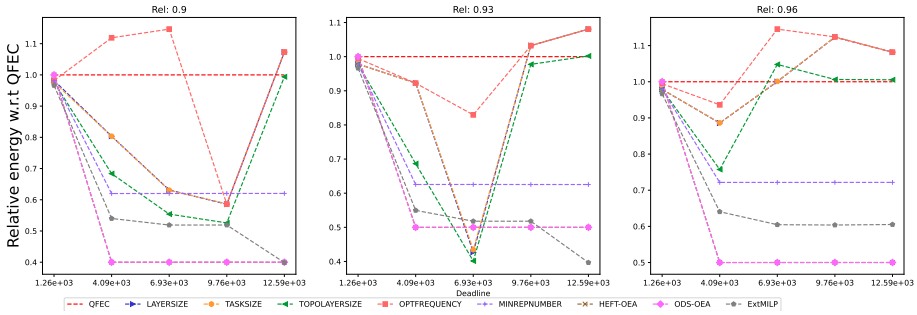

Figure 1470: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

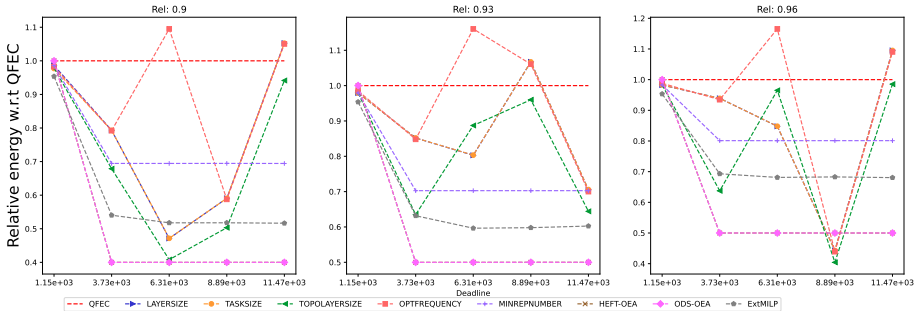

Figure 1471: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

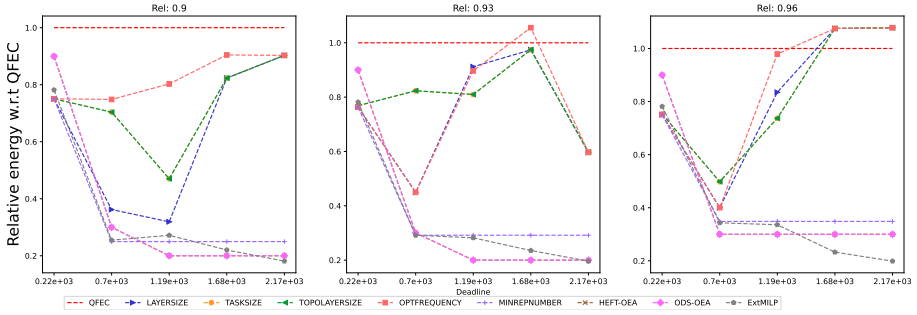

Figure 1472: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

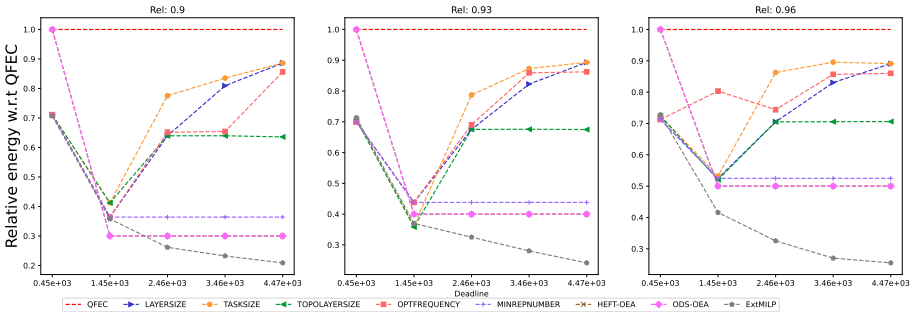

Figure 1473: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

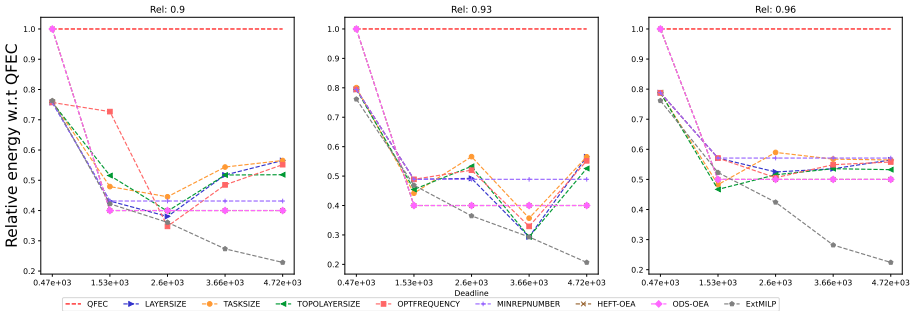

Figure 1474: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

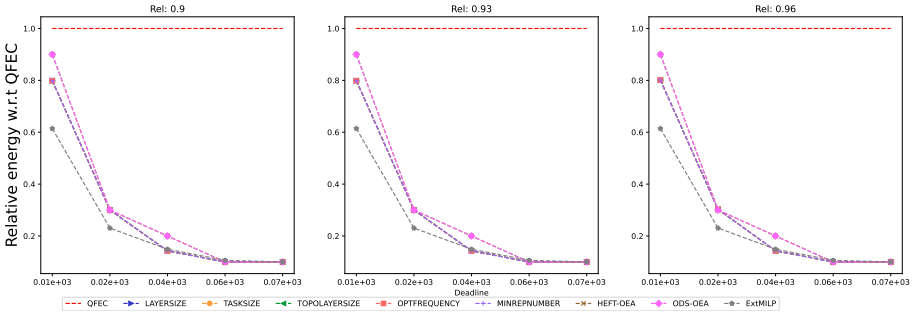

Figure 1475: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

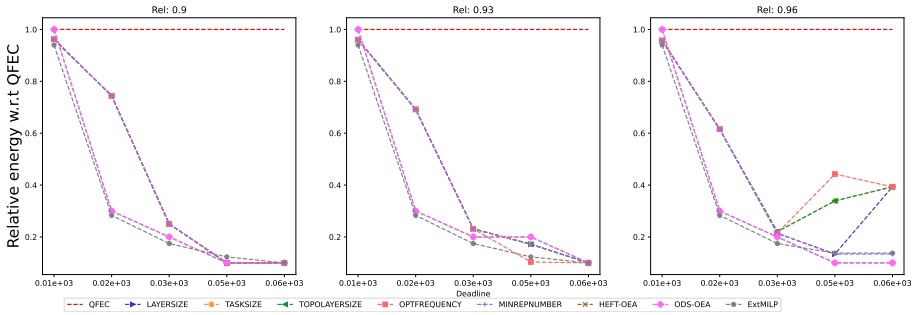

Figure 1476: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

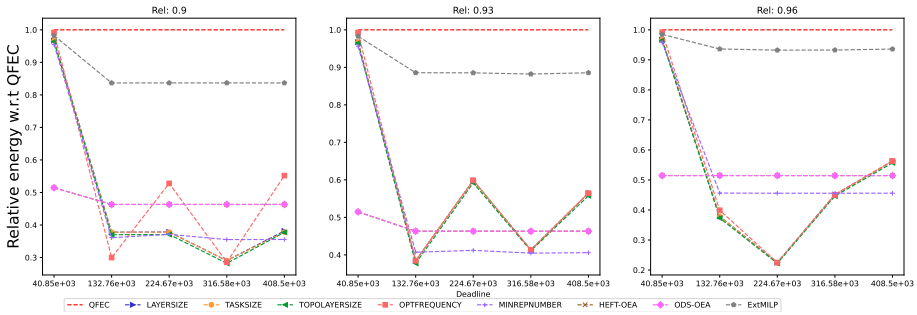

Figure 1477: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

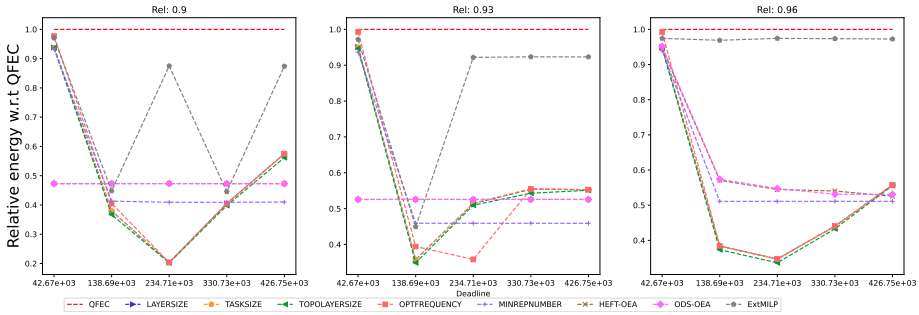

Figure 1478: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

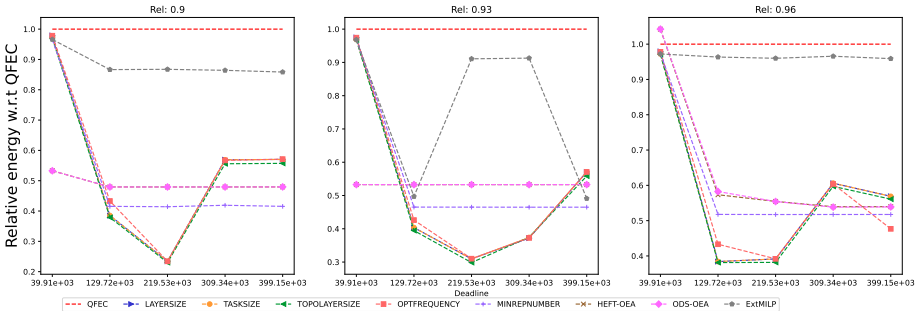

Figure 1479: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

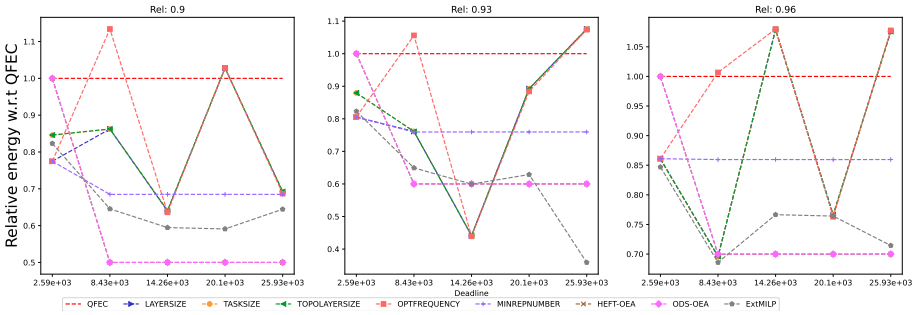

Figure 1480: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

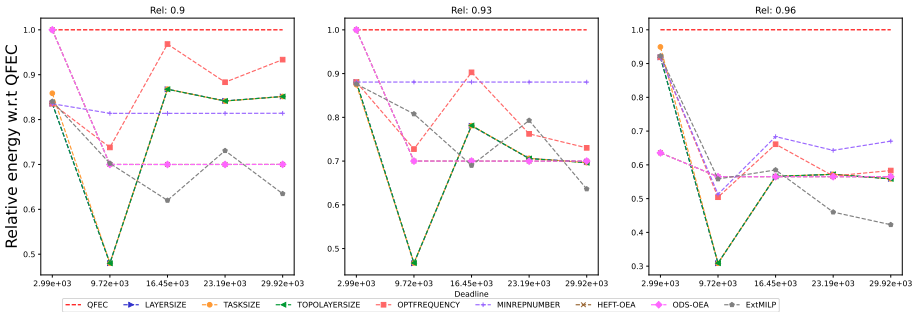

Figure 1481: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.9.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

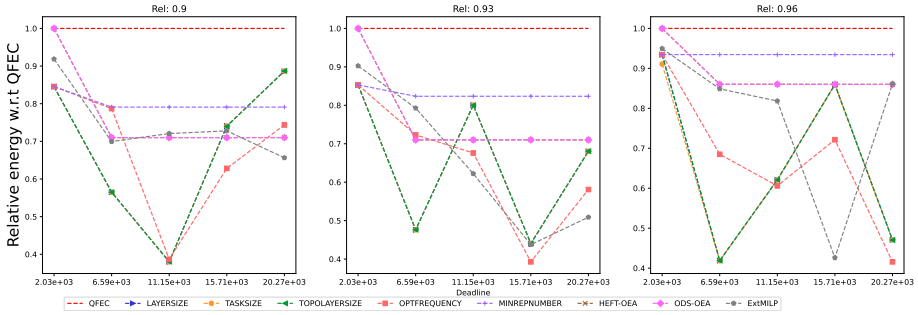

Figure 1482: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

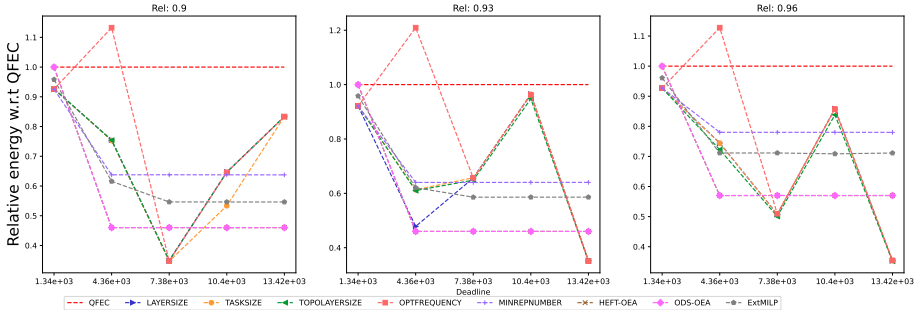

Figure 1483: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

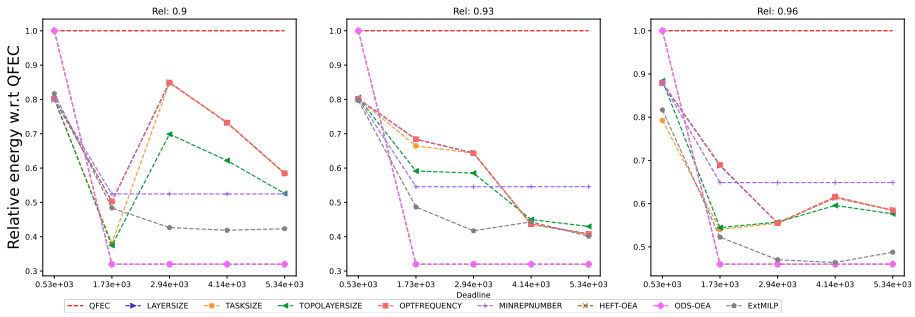

Figure 1484: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

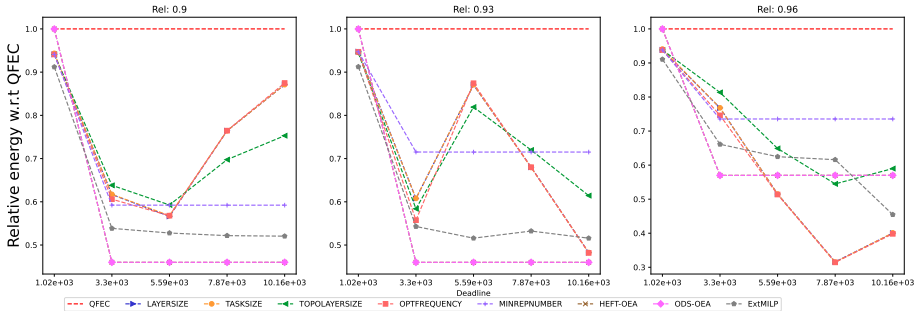

Figure 1485: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

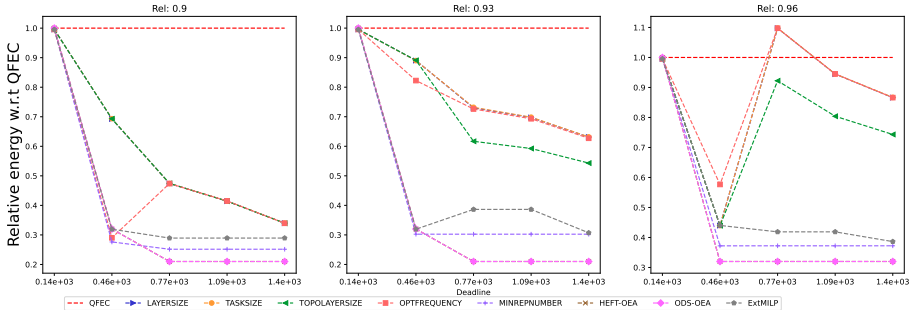

Figure 1486: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

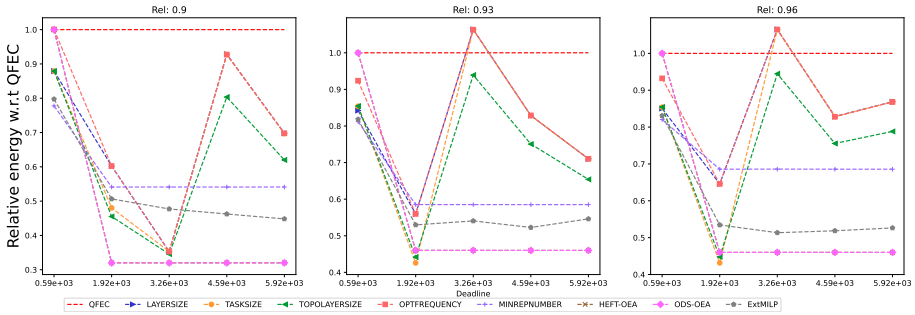

Figure 1487: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

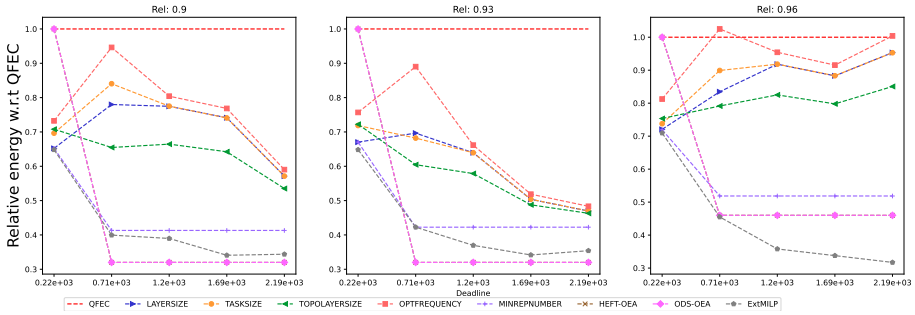

Figure 1488: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

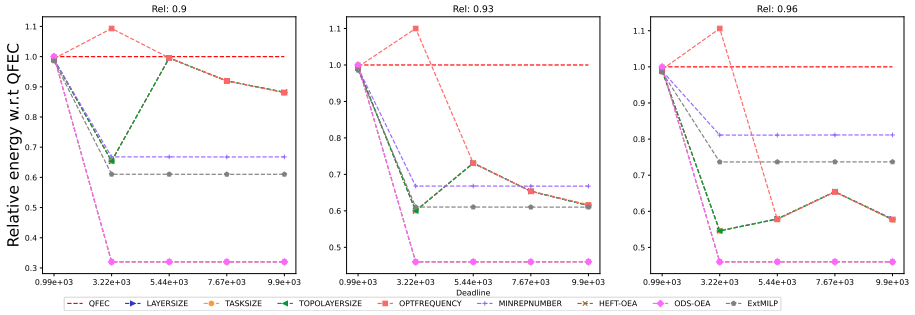

Figure 1489: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

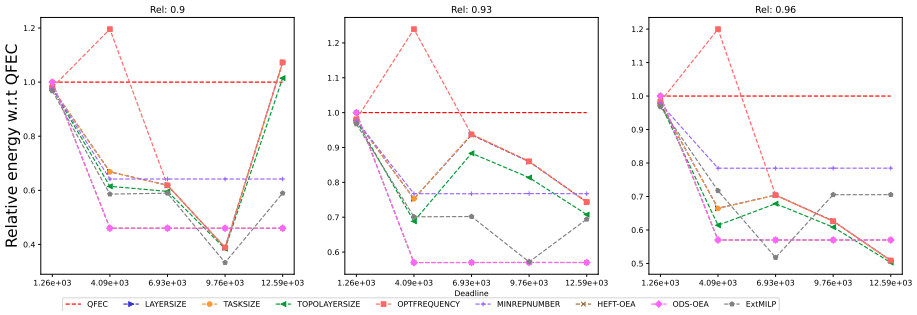

Figure 1490: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

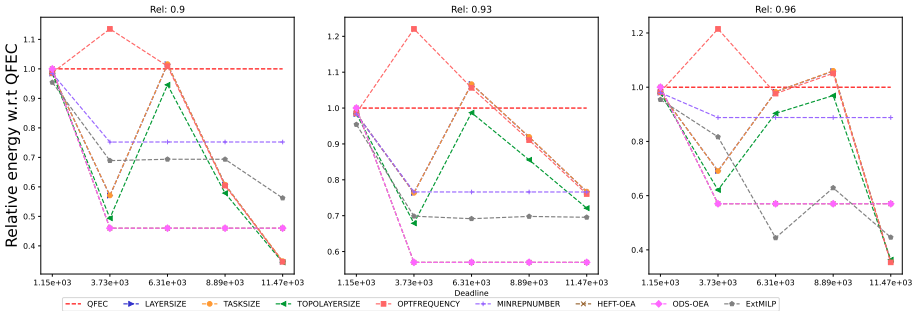

Figure 1491: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

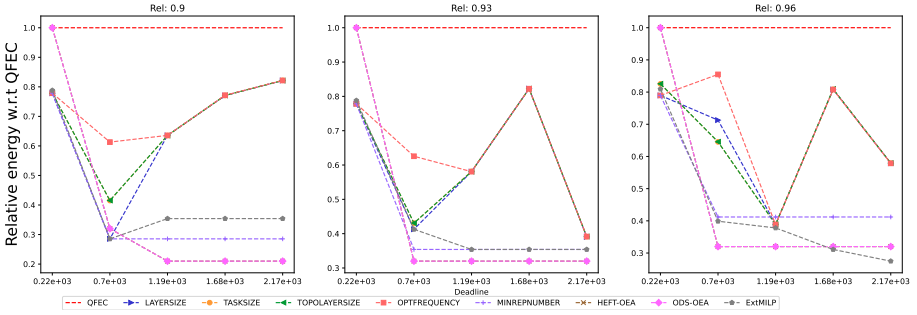

Figure 1492: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

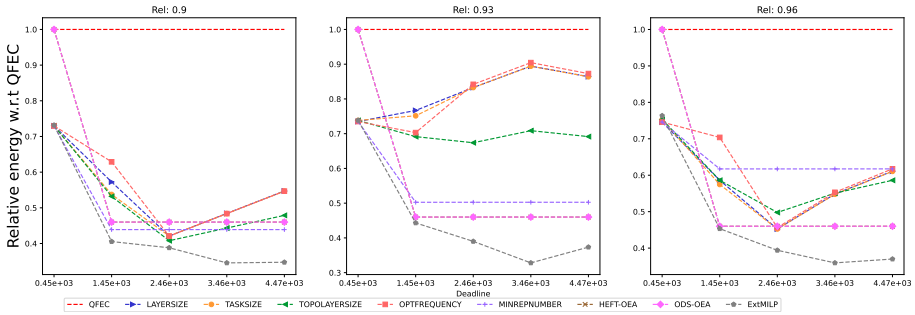

Figure 1493: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

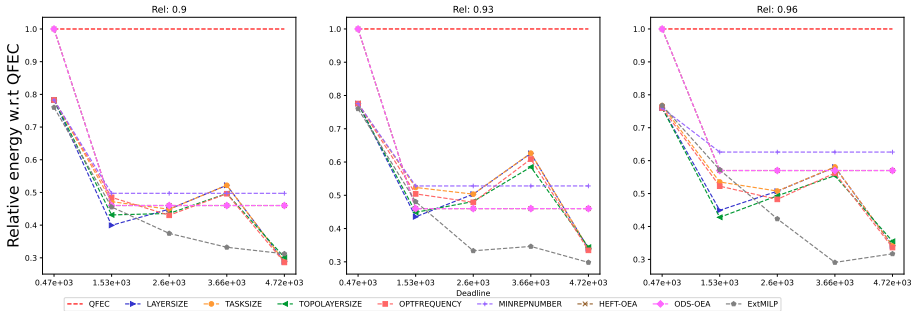

Figure 1494: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

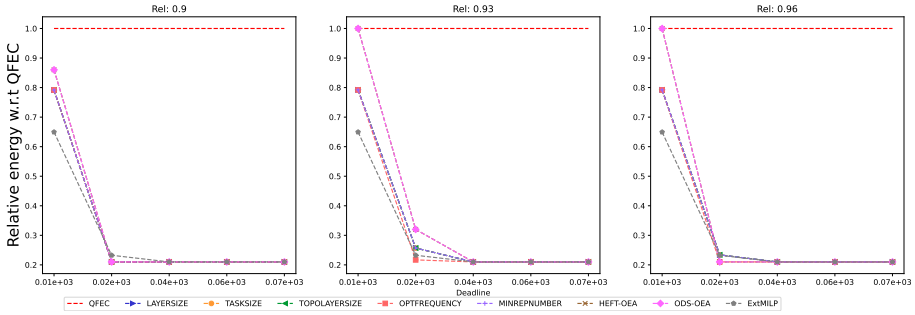

Figure 1495: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

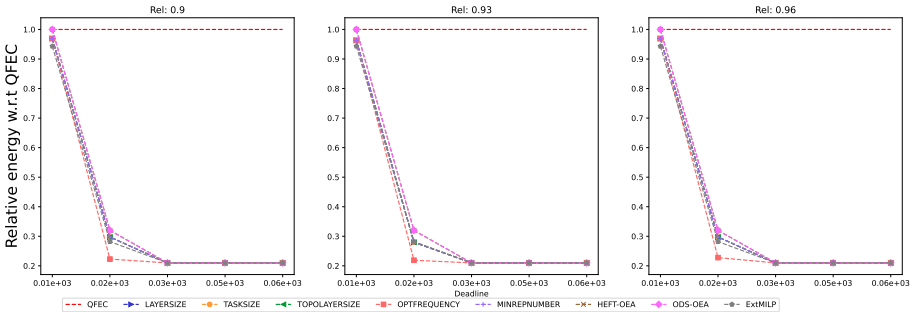

Figure 1496: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

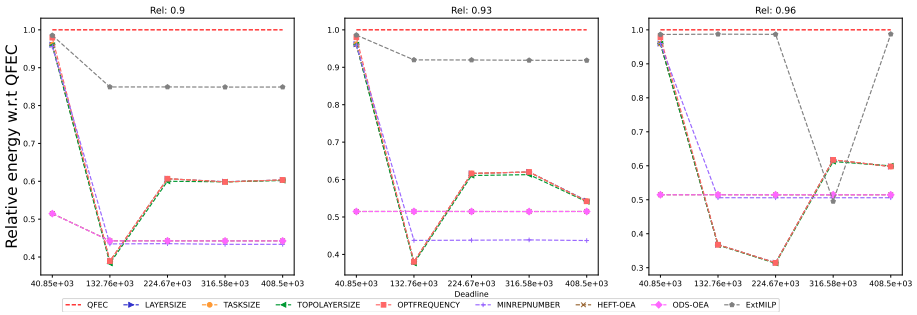

Figure 1497: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

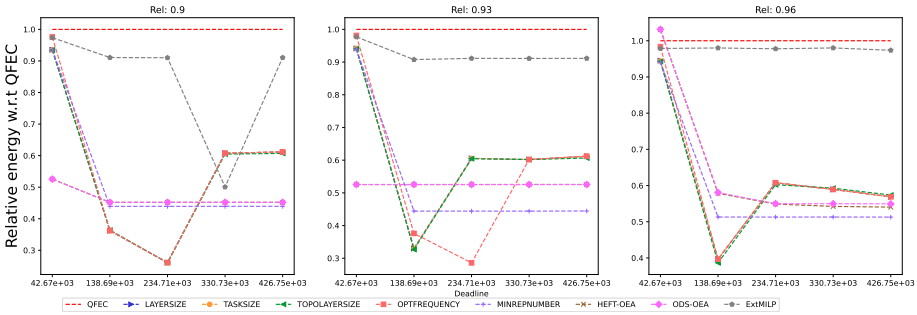

Figure 1498: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

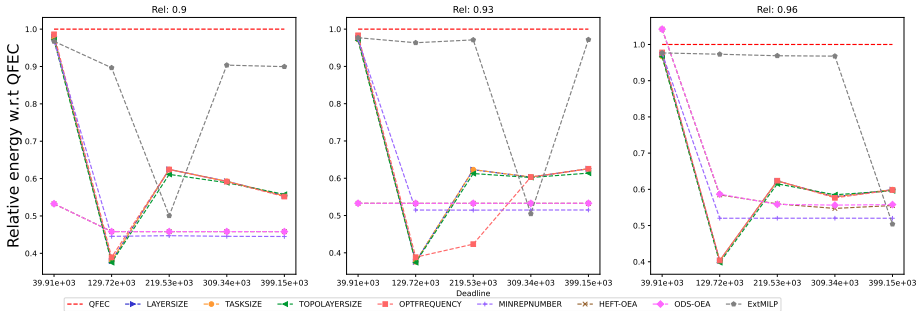

Figure 1499: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

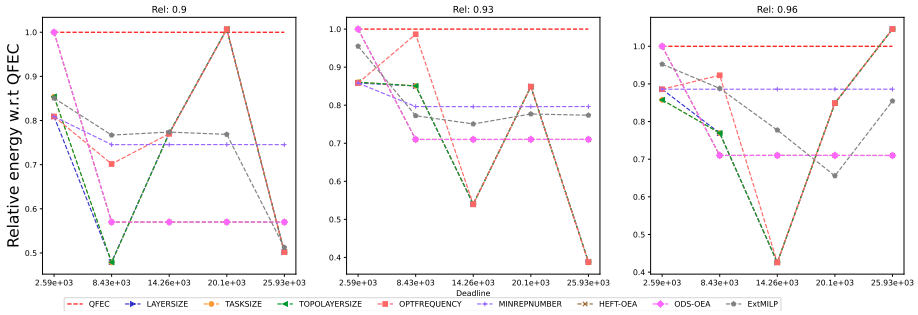

Figure 1500: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

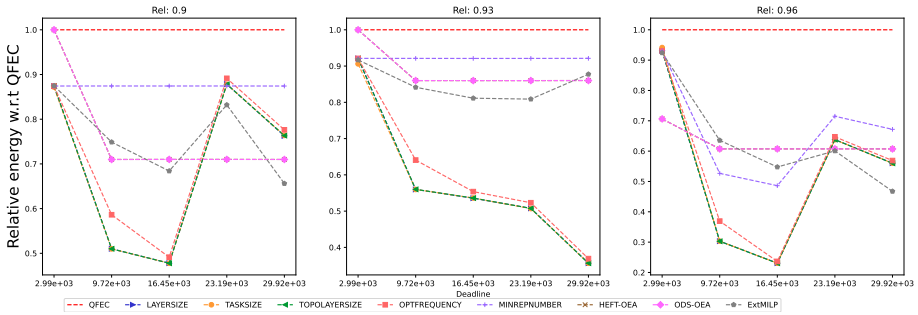

Figure 1501: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

F.9.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

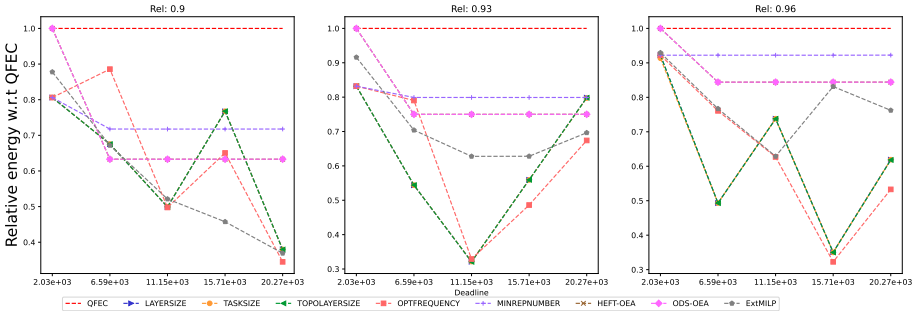

Figure 1502: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

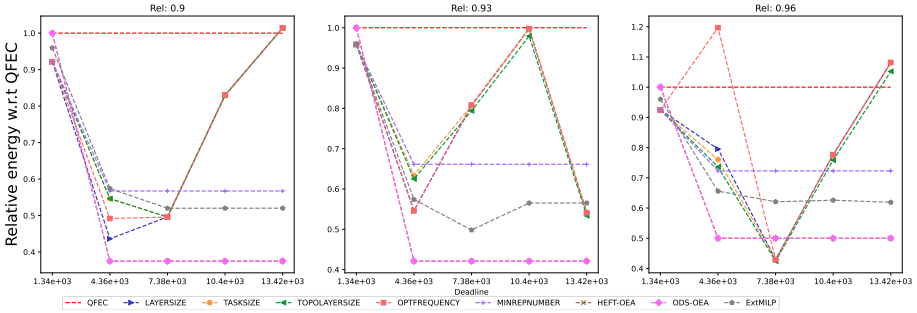

Figure 1503: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

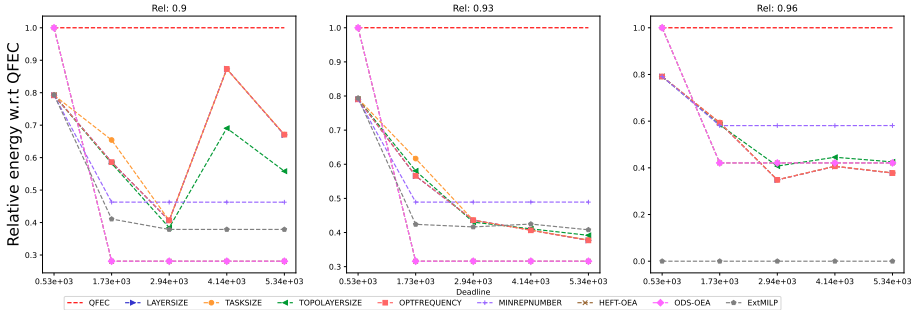

Figure 1504: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

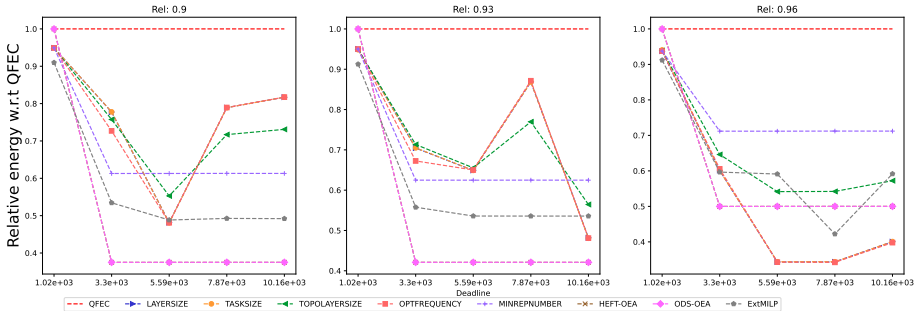

Figure 1505: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

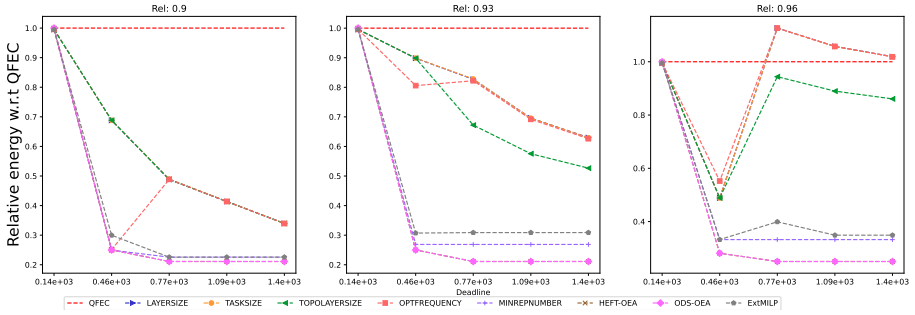

Figure 1506: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

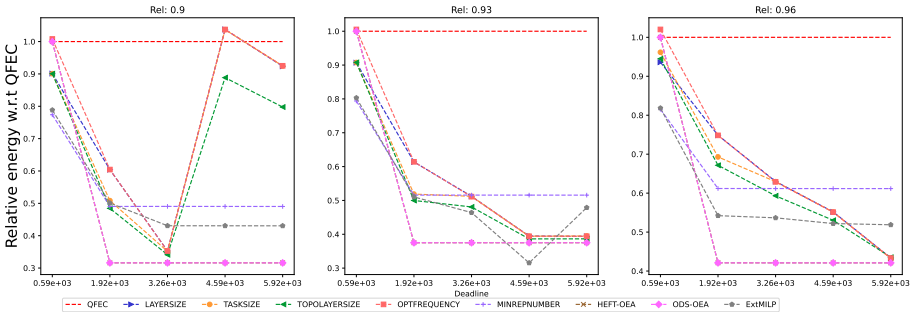

Figure 1507: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

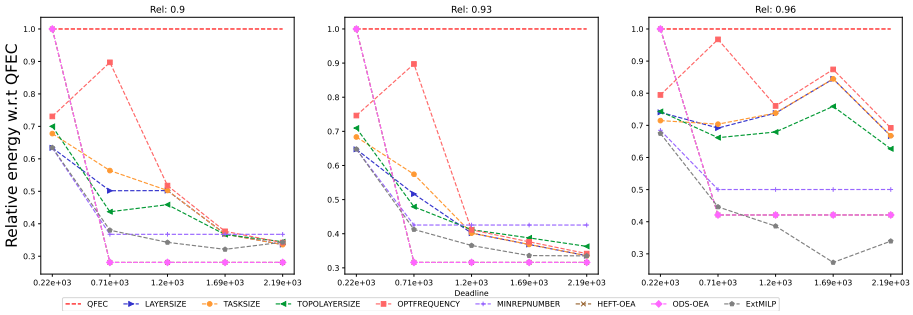

Figure 1508: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

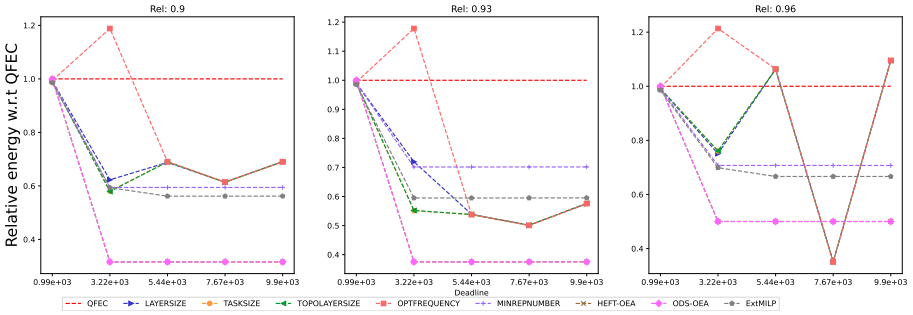

Figure 1509: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

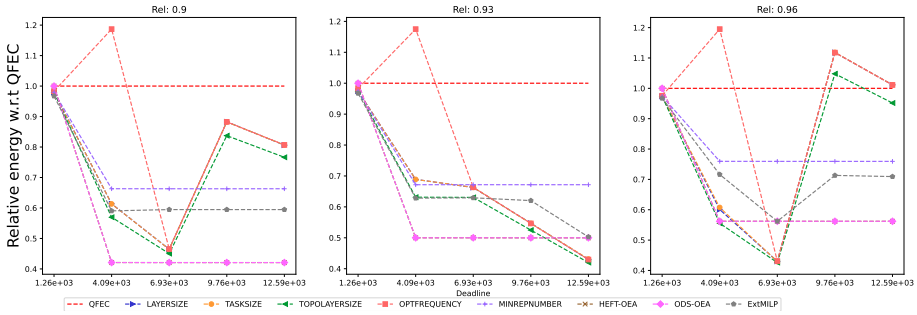

Figure 1510: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

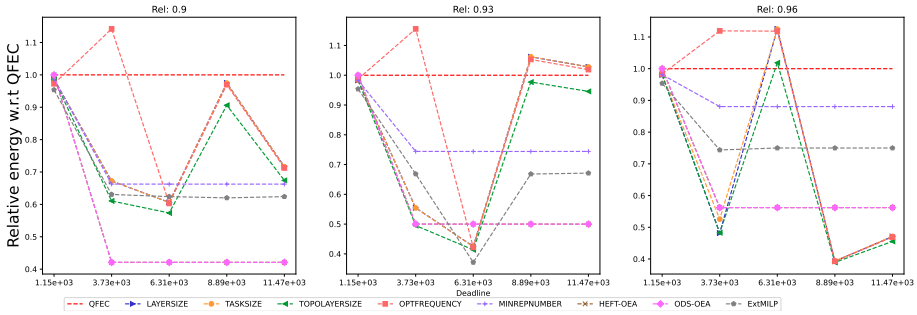

Figure 1511: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

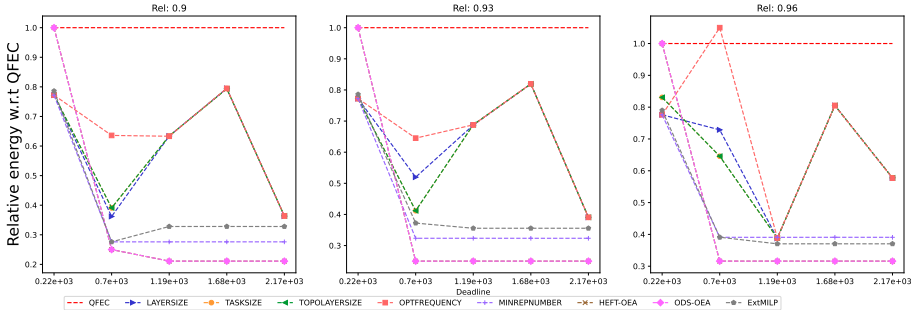

Figure 1512: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

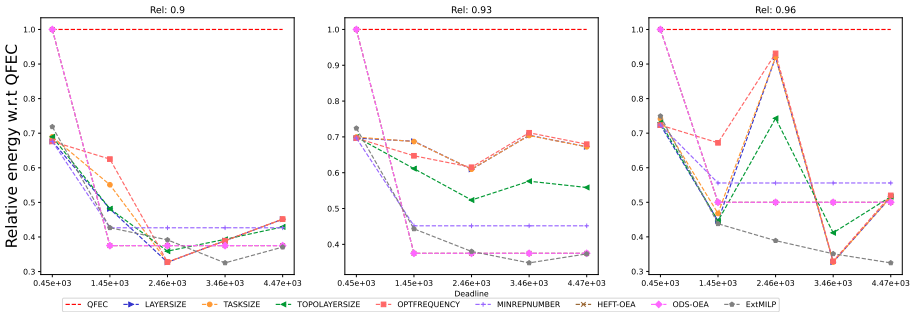

Figure 1513: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

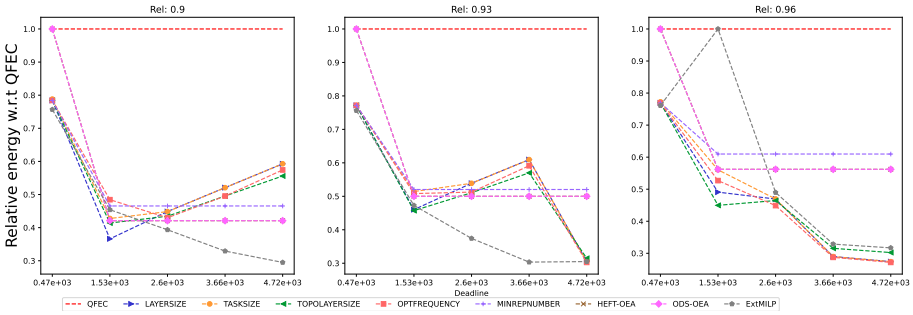

Figure 1514: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

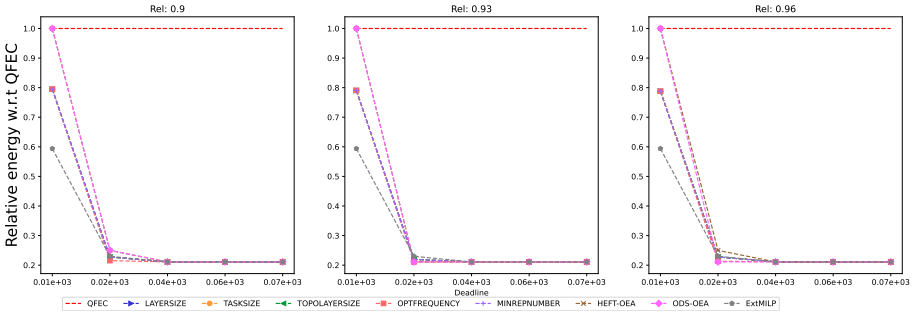

Figure 1515: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

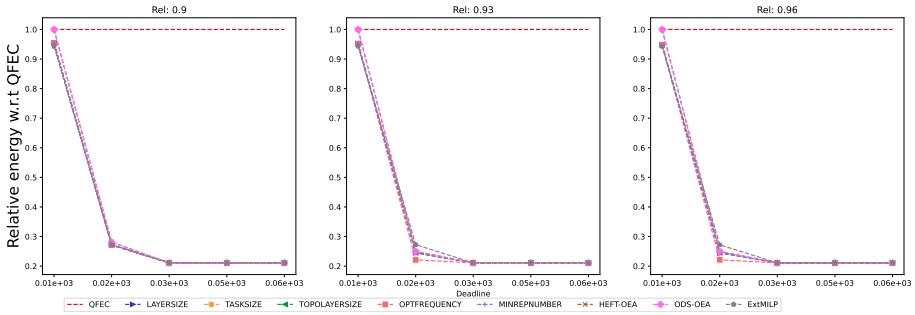

Figure 1516: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

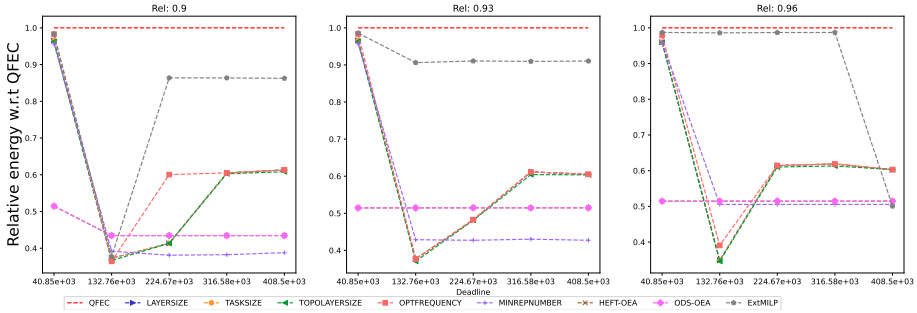

Figure 1517: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

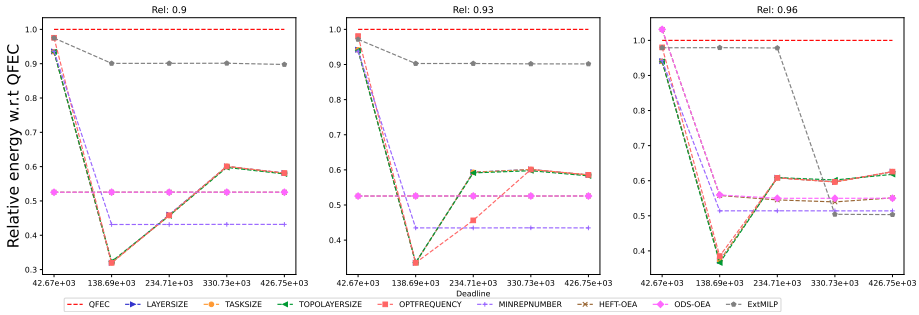

Figure 1518: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

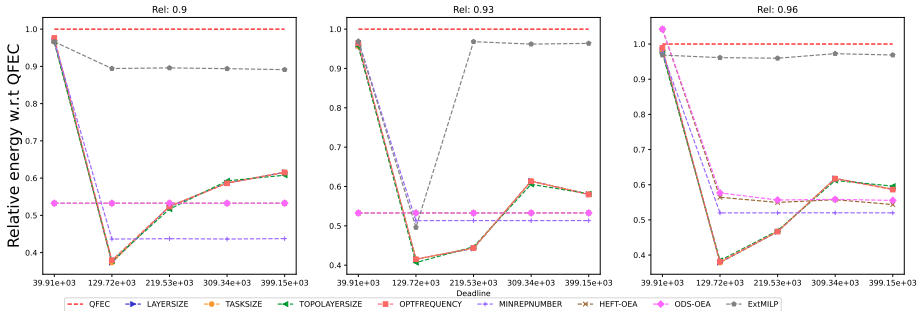

Figure 1519: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

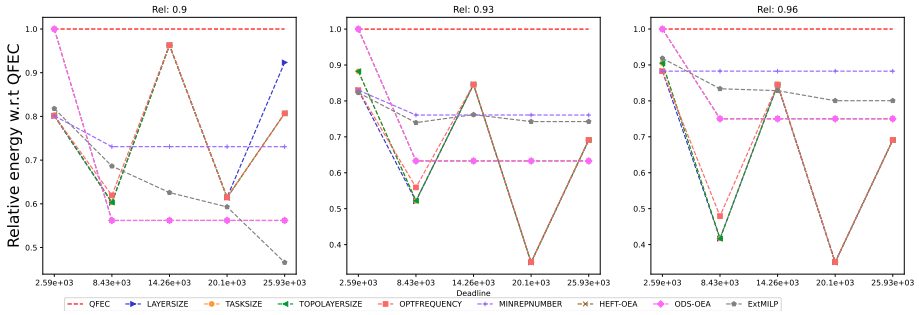

Figure 1520: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

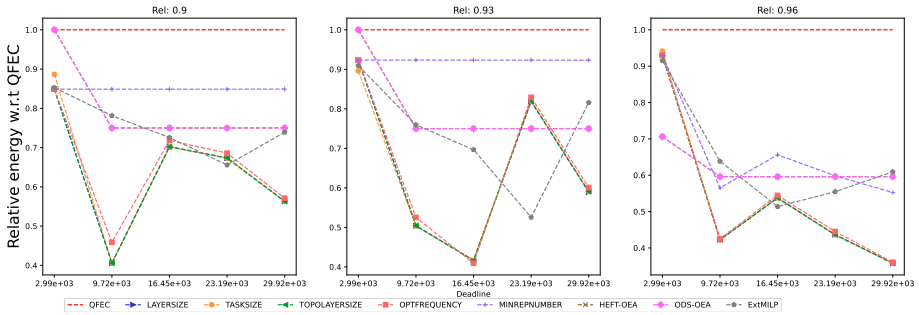

Figure 1521: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G Study of ExtMILP when actual execution times are drawn from a uniform distribution

G.1 $BC/WC = 0.1$

G.1.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

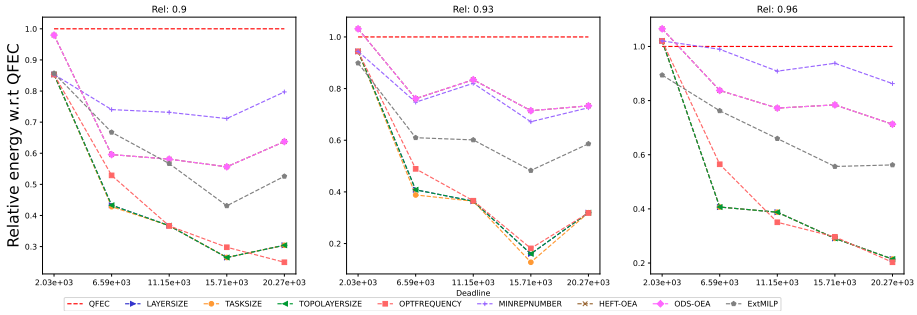

Figure 1522: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

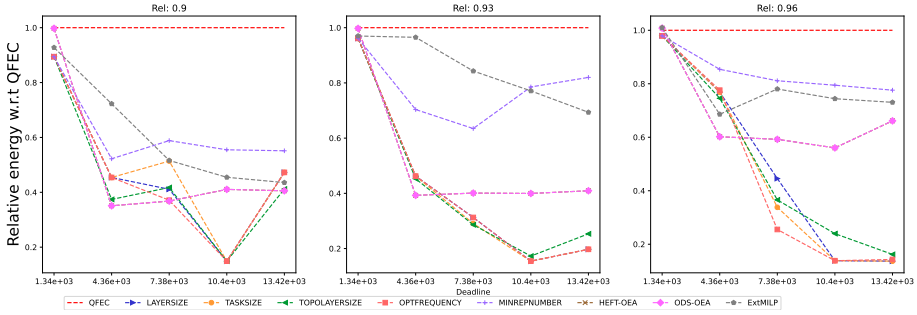

Figure 1523: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

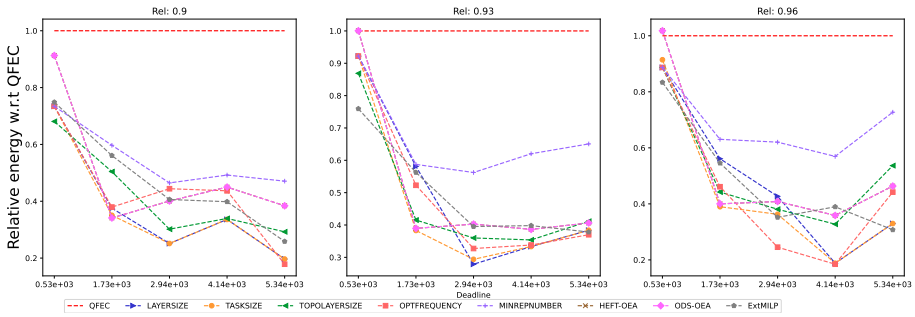

Figure 1524: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

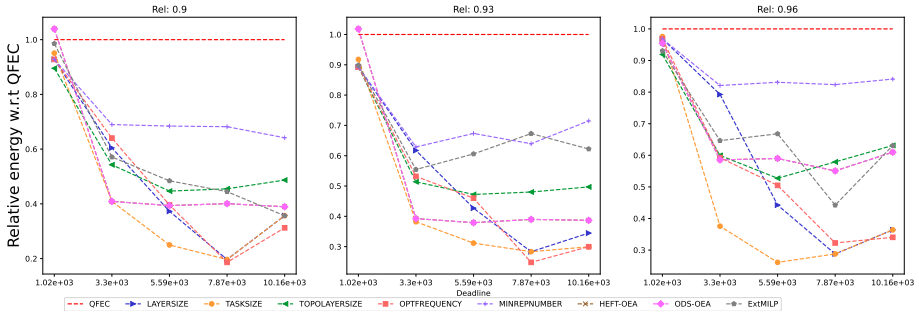

Figure 1525: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

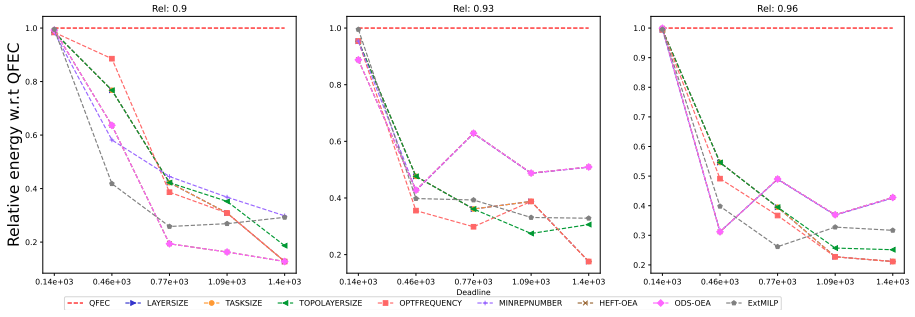

Figure 1526: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

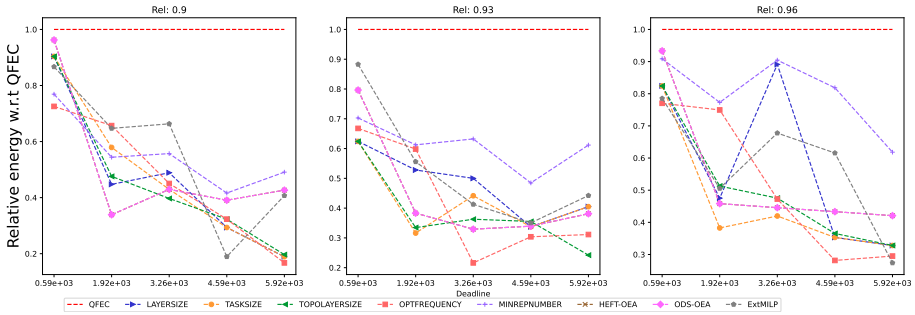

Figure 1527: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

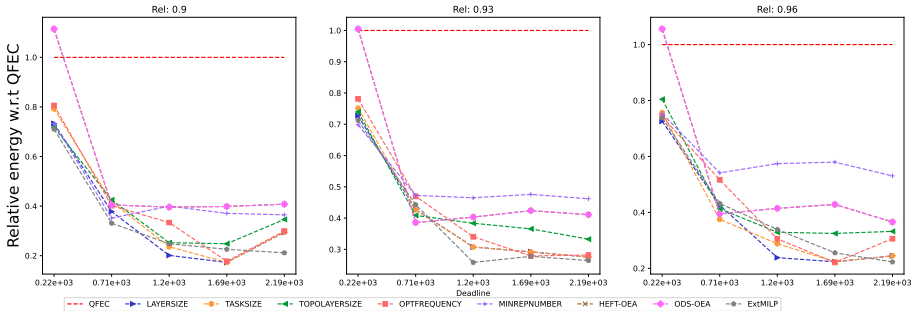

Figure 1528: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

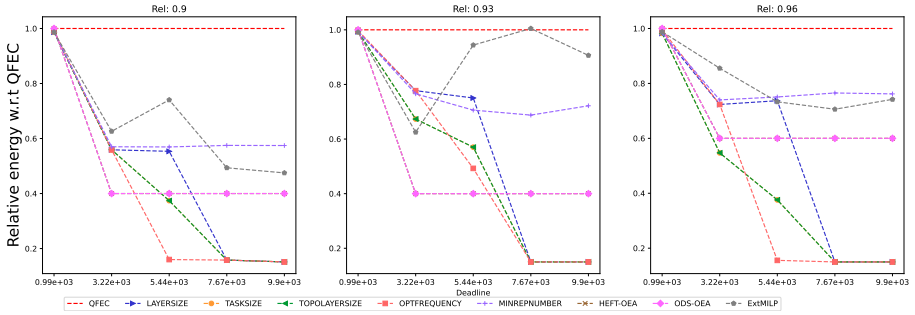

Figure 1529: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

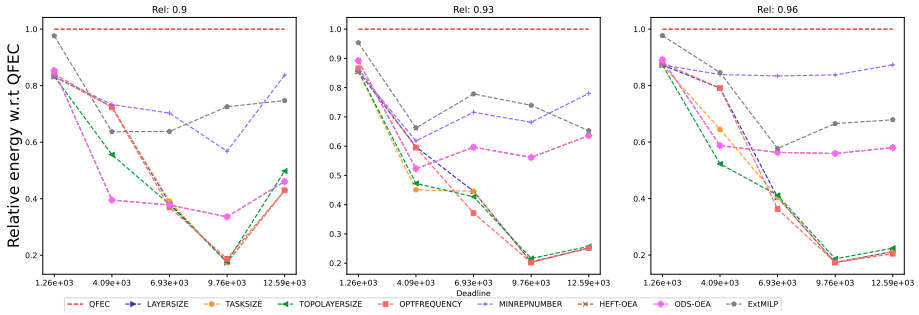

Figure 1530: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

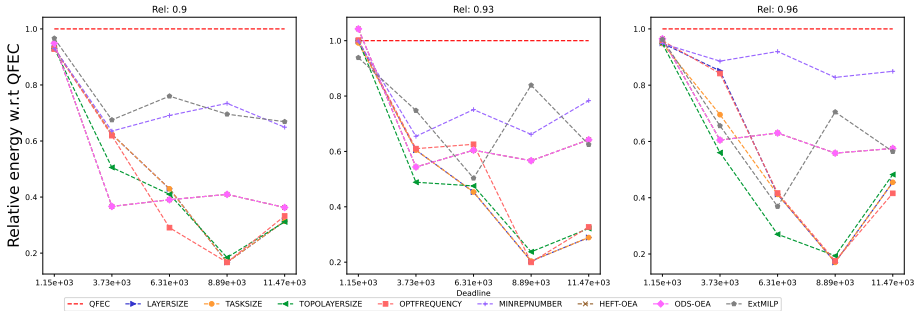

Figure 1531: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

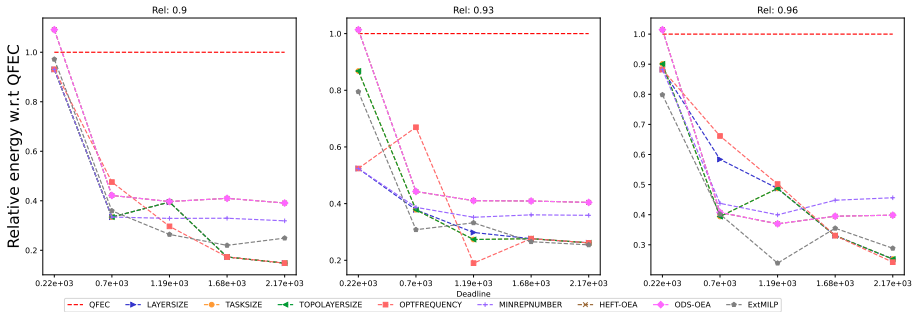

Figure 1532: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

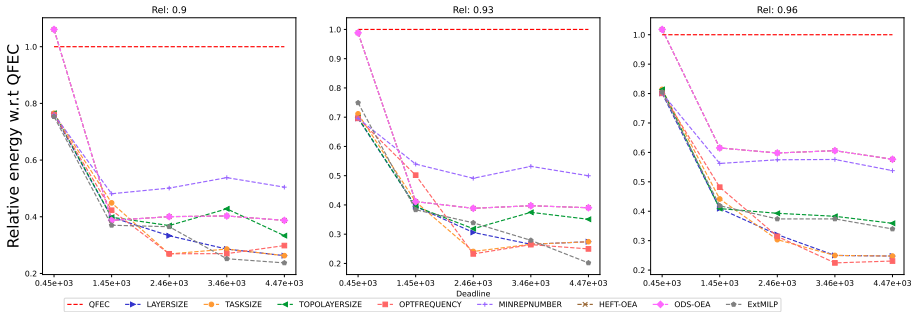

Figure 1533: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

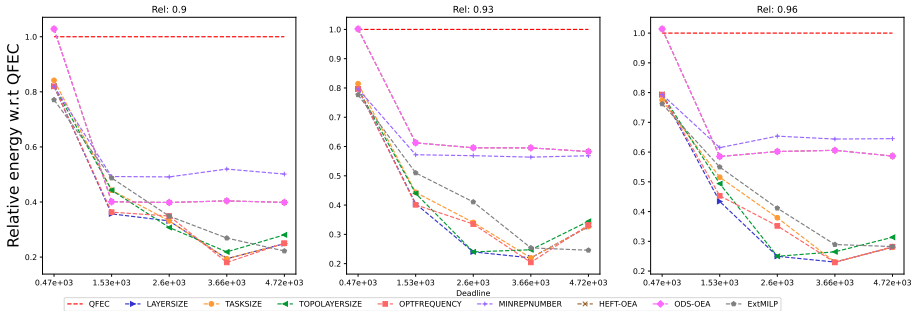

Figure 1534: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

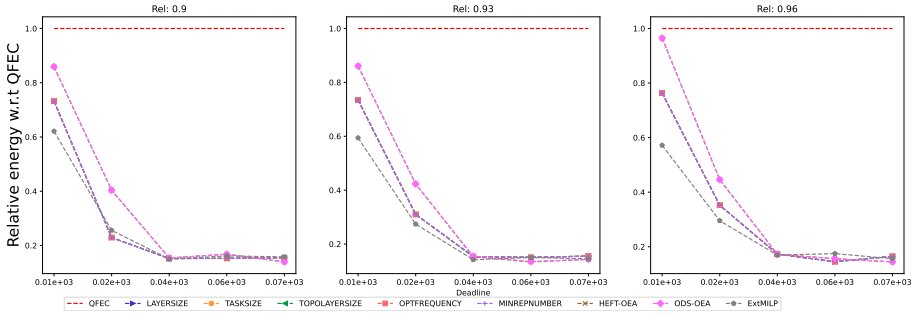

Figure 1535: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

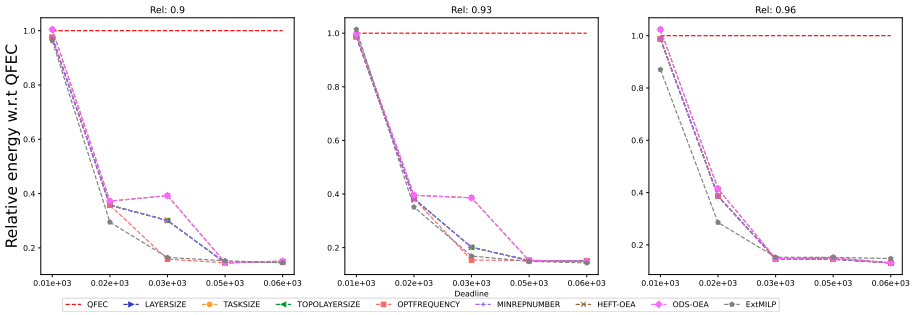

Figure 1536: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

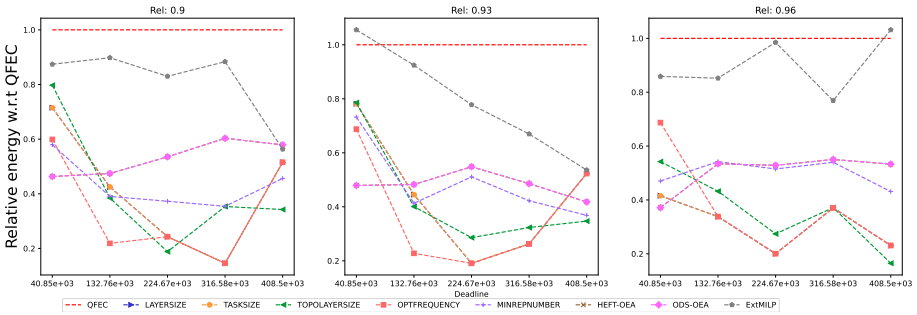

Figure 1537: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

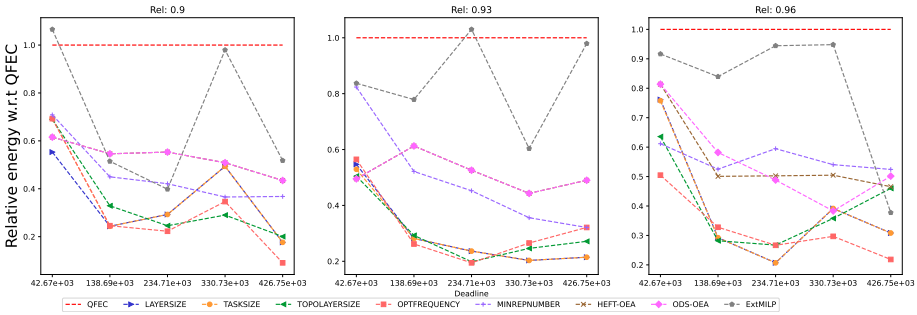

Figure 1538: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

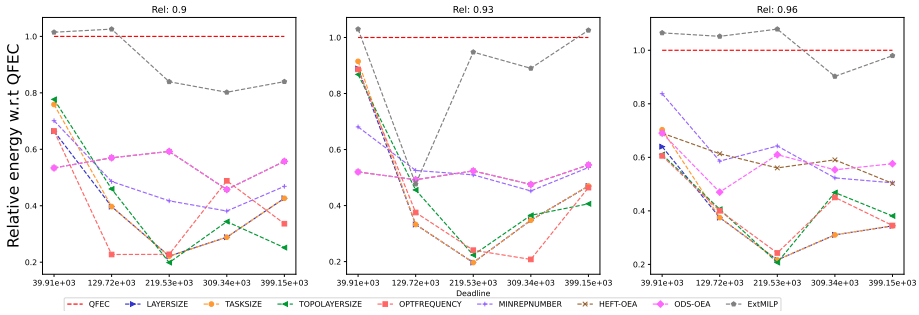

Figure 1539: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

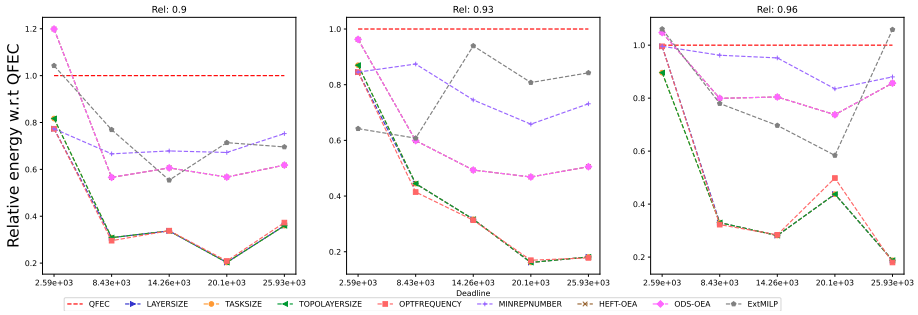

Figure 1540: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

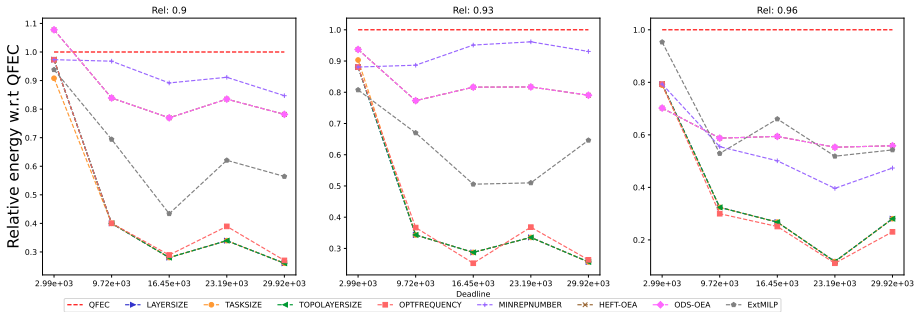

Figure 1541: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.1.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

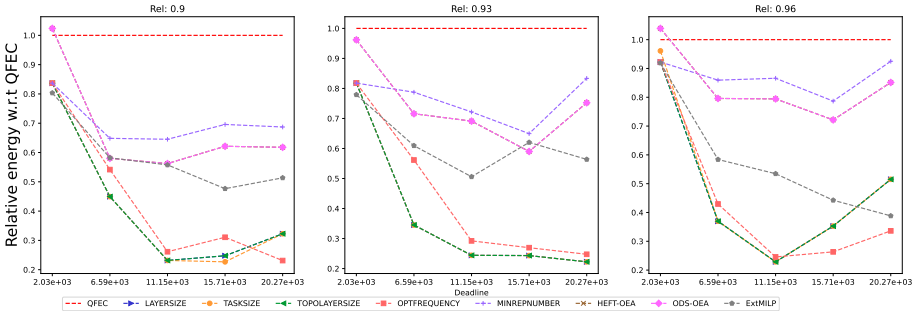

Figure 1542: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

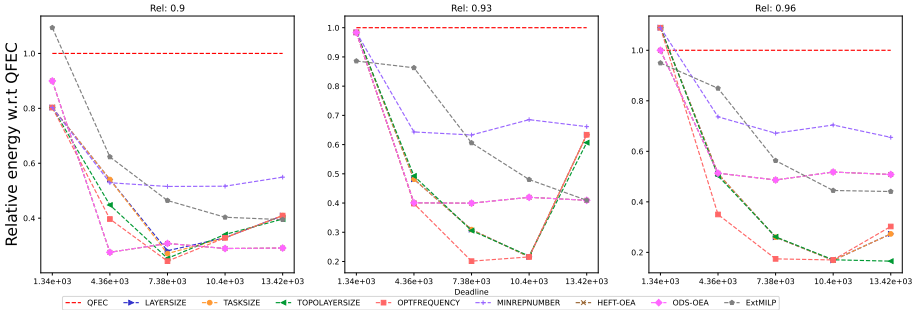

Figure 1543: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

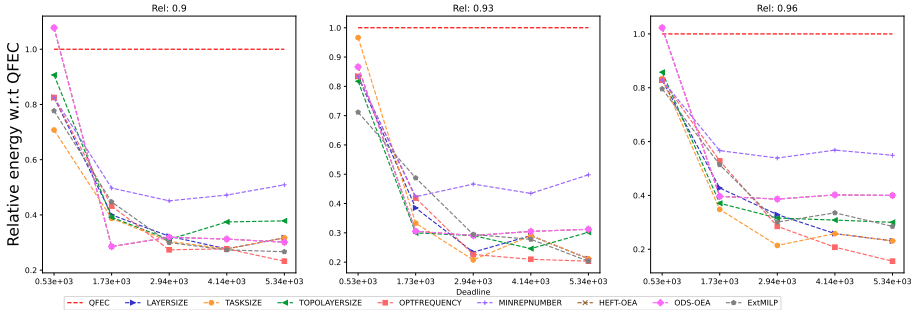

Figure 1544: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

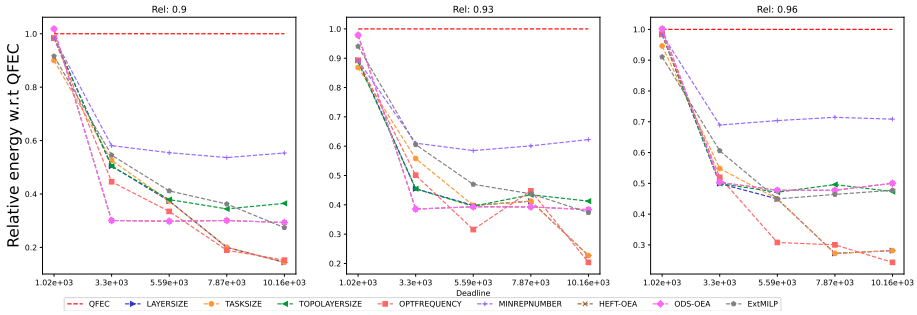

Figure 1545: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

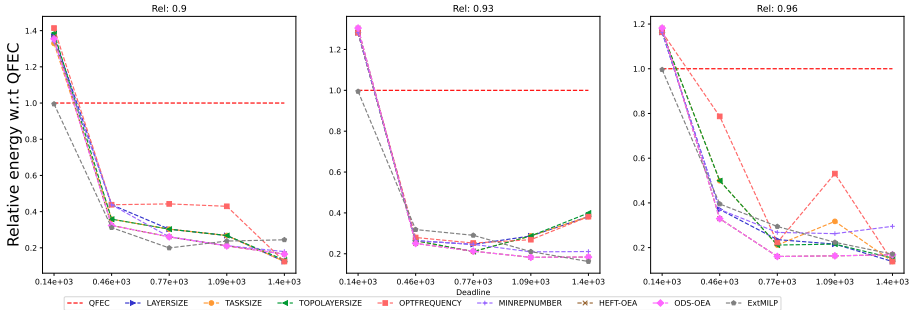

Figure 1546: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

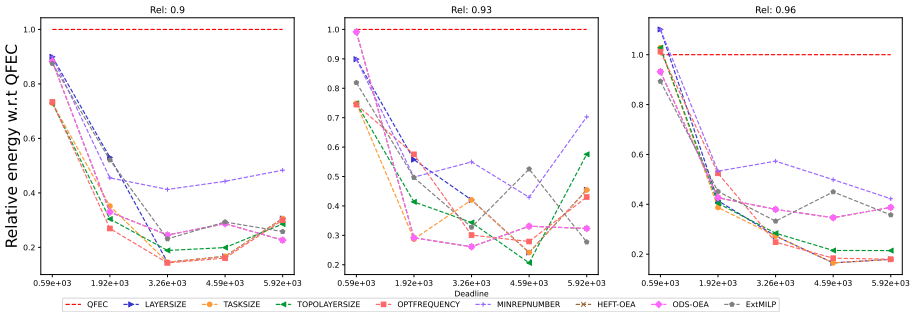

Figure 1547: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

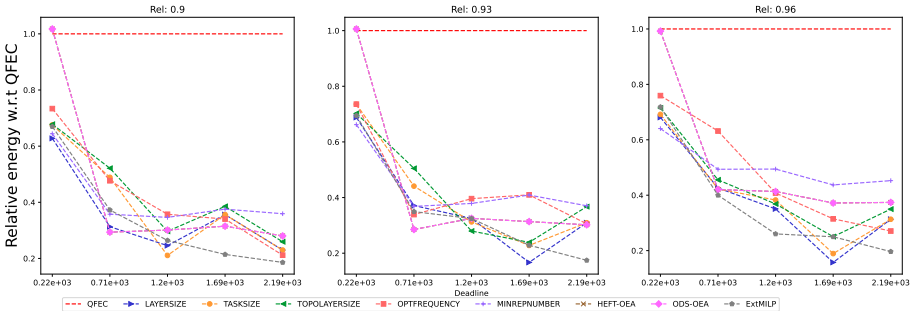

Figure 1548: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

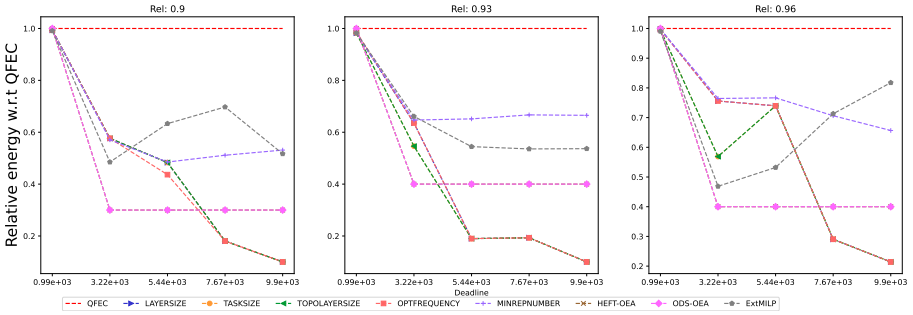

Figure 1549: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

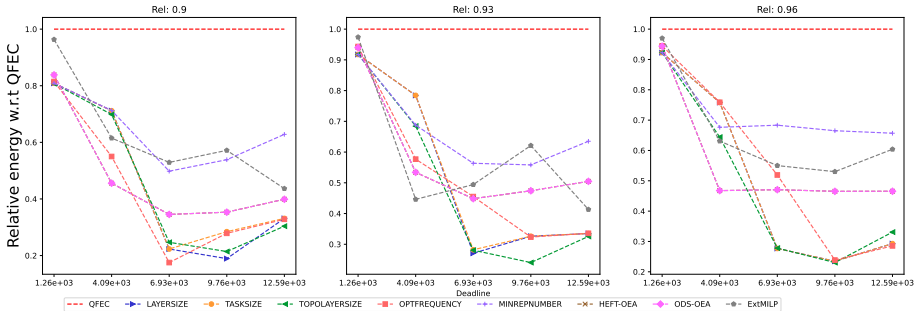

Figure 1550: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

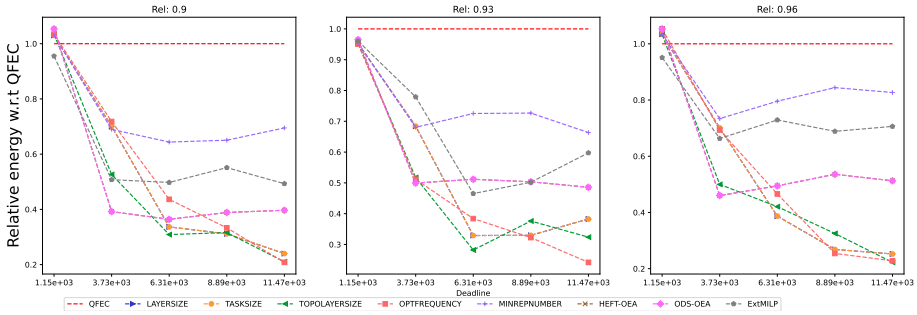

Figure 1551: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

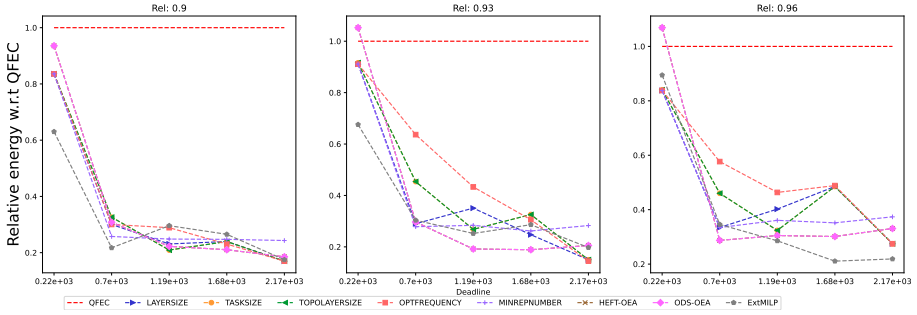

Figure 1552: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

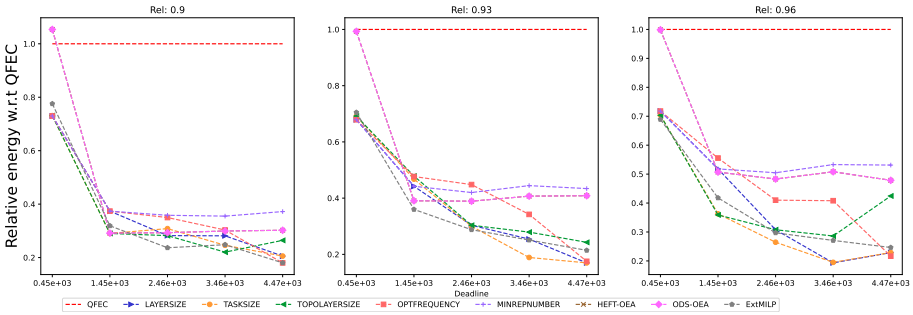

Figure 1553: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

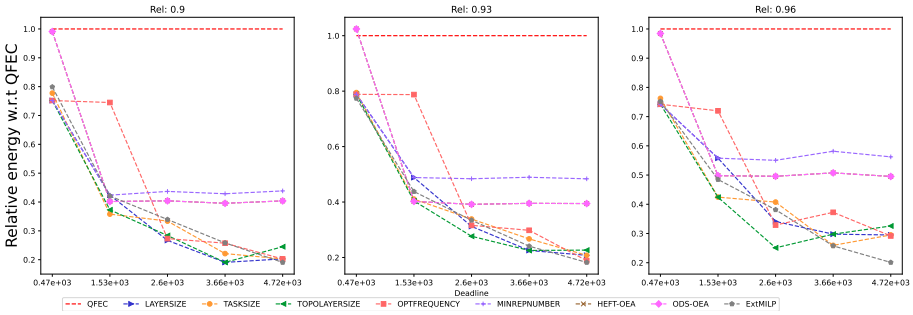

Figure 1554: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

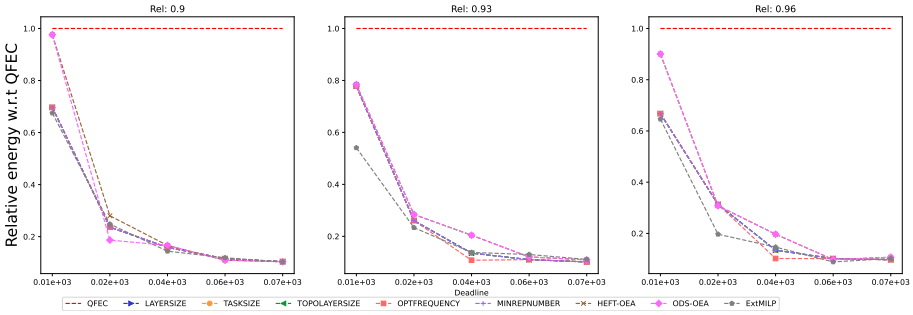

Figure 1555: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

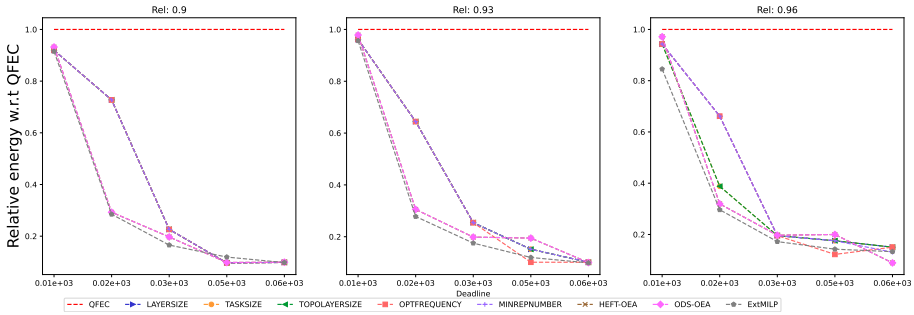

Figure 1556: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

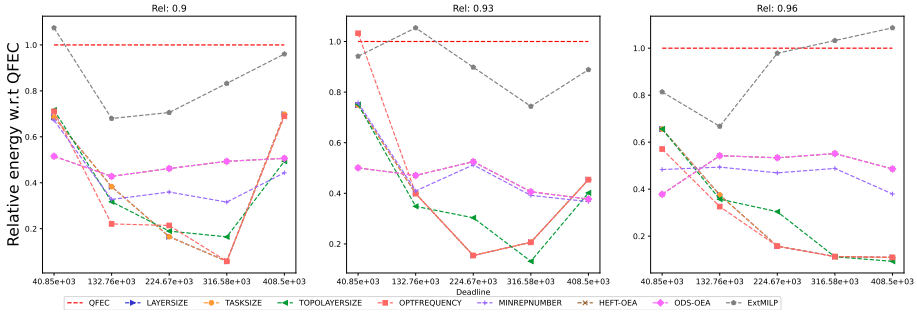

Figure 1557: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

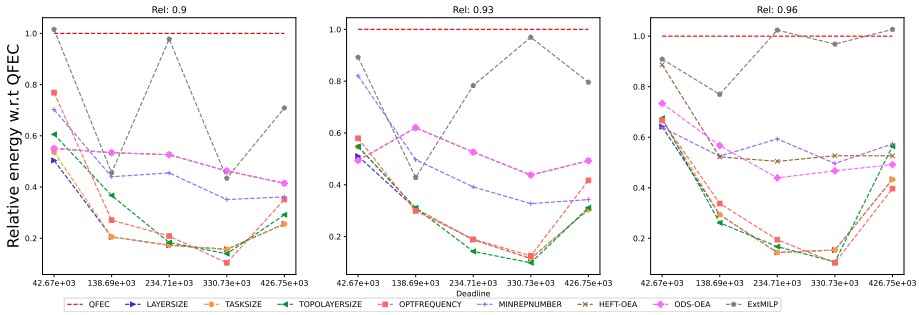

Figure 1558: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

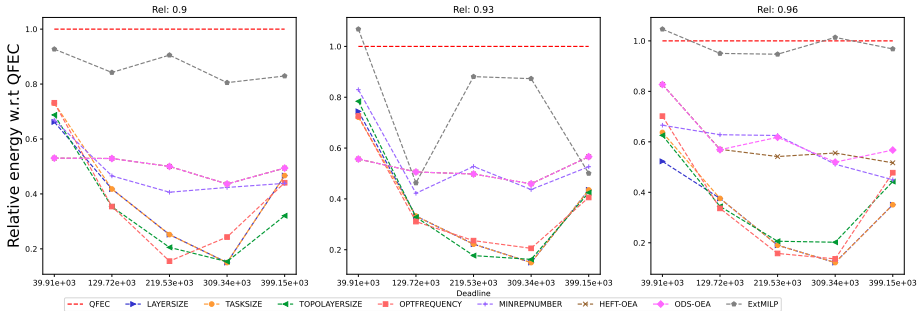

Figure 1559: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

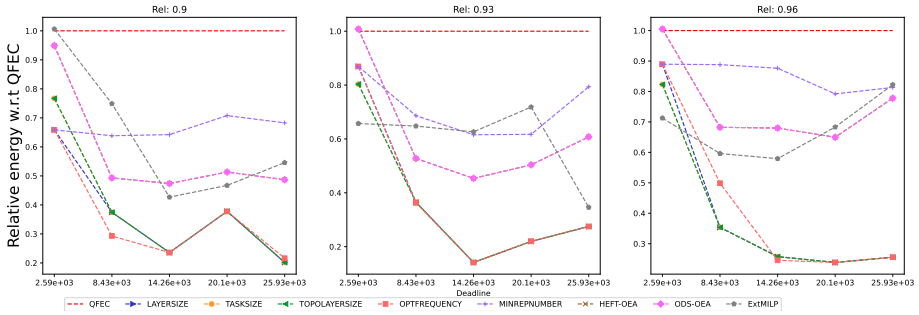

Figure 1560: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

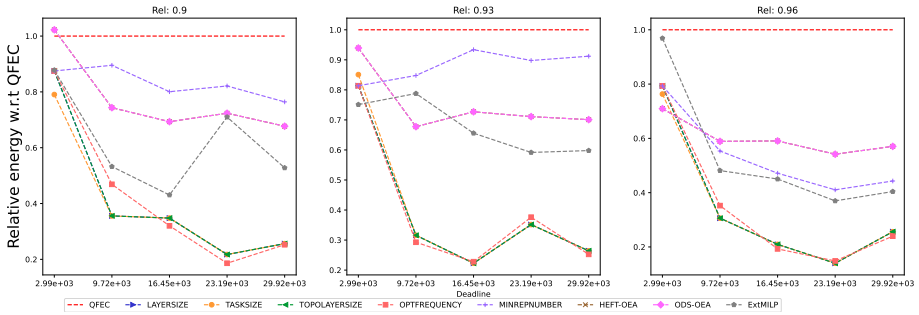

Figure 1561: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.1.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

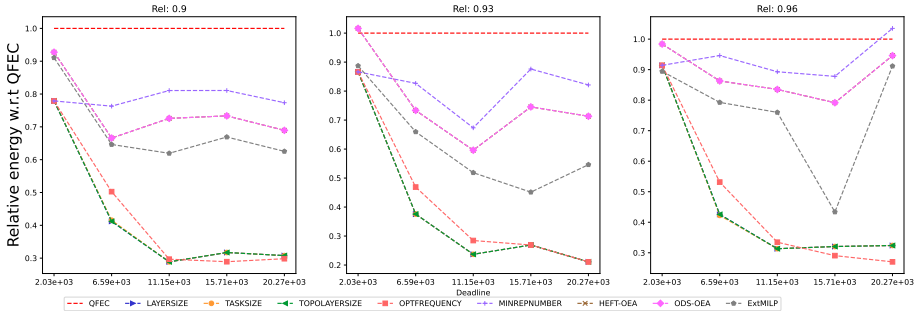

Figure 1562: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

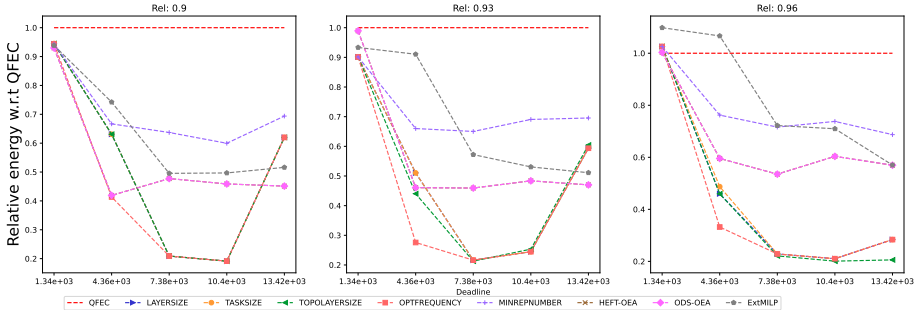

Figure 1563: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

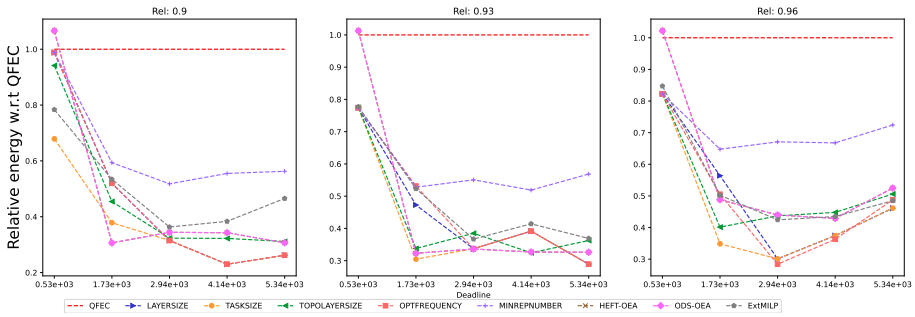

Figure 1564: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

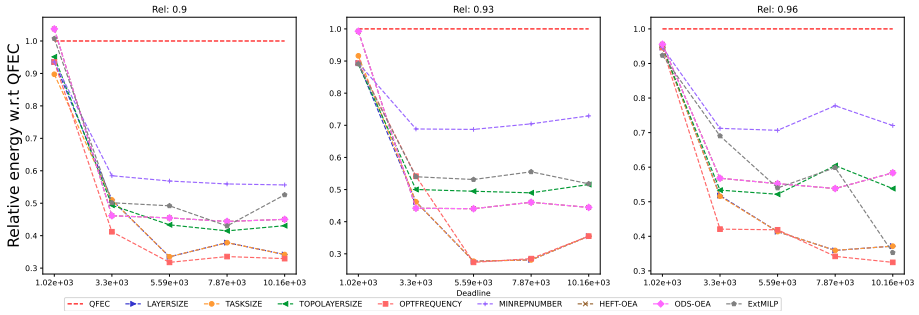

Figure 1565: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

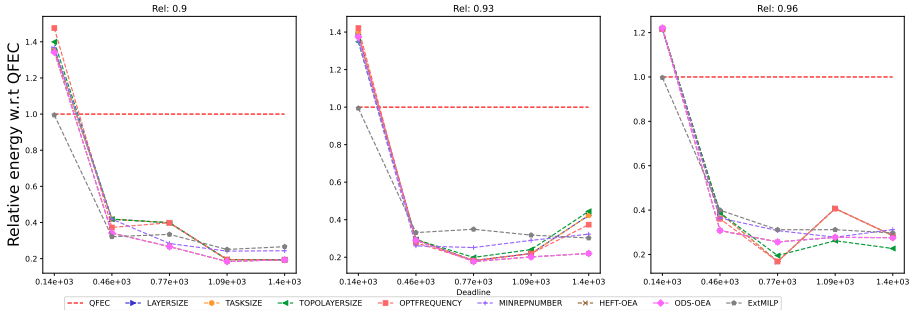

Figure 1566: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

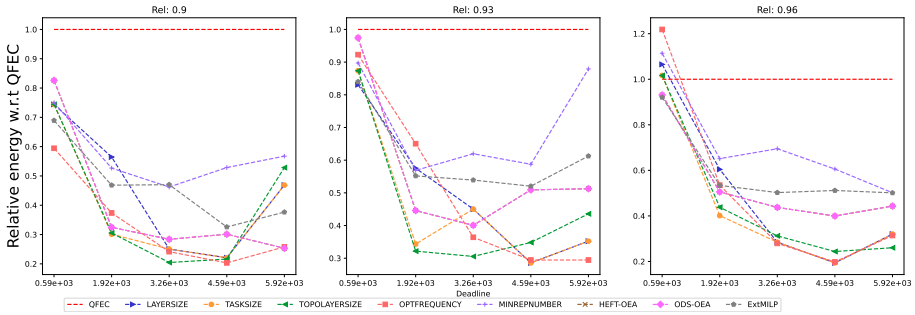

Figure 1567: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

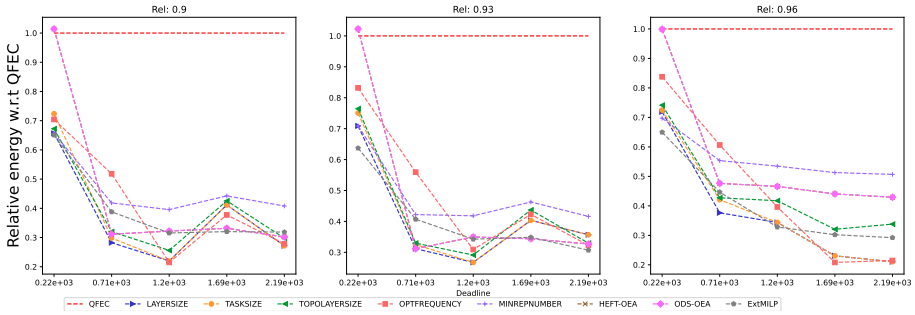

Figure 1568: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

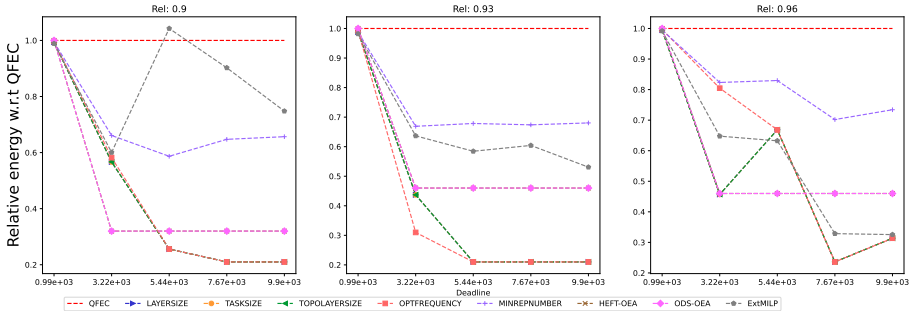

Figure 1569: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

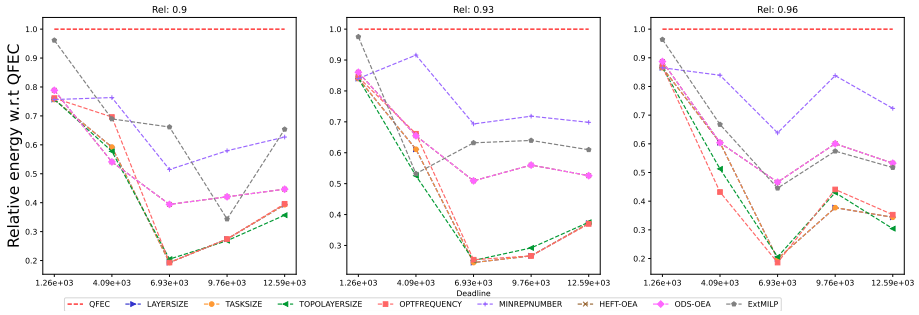

Figure 1570: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

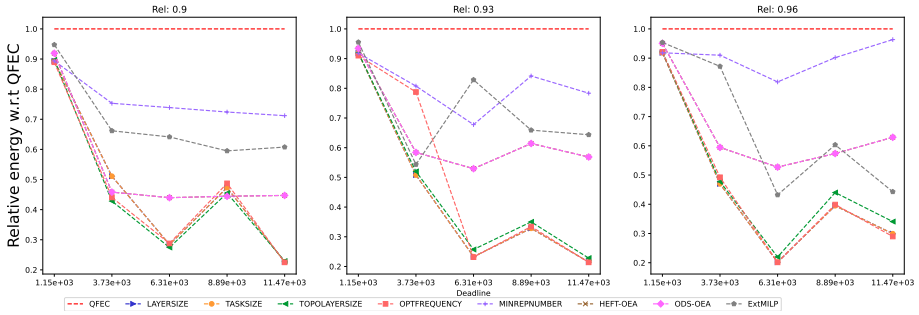

Figure 1571: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

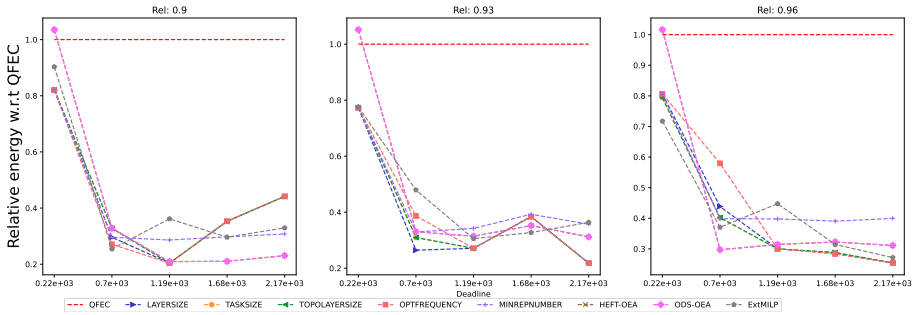

Figure 1572: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

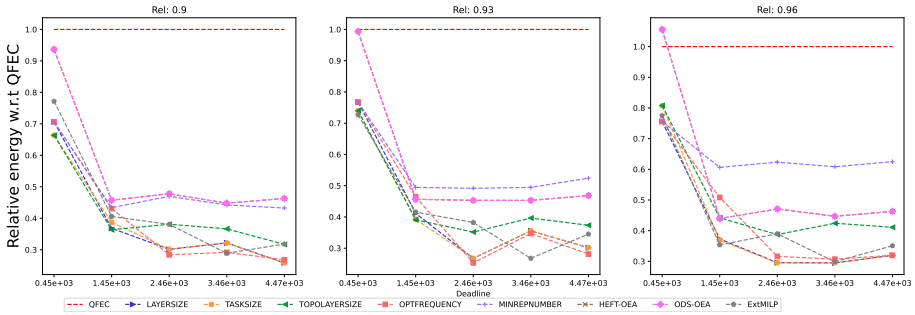

Figure 1573: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

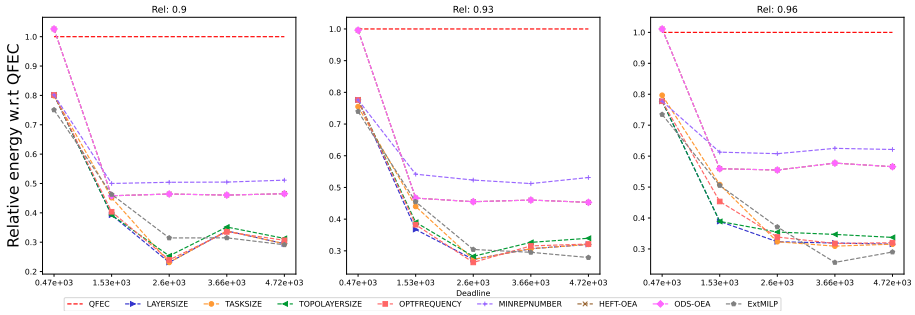

Figure 1574: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

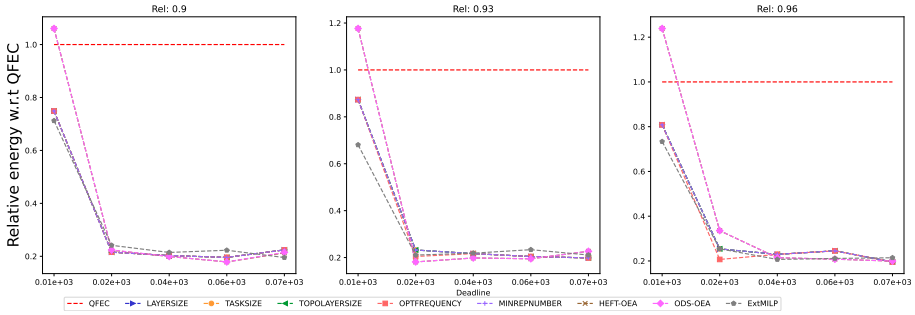

Figure 1575: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

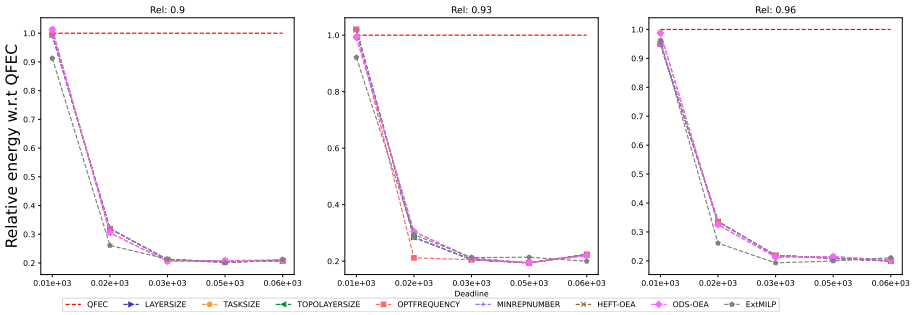

Figure 1576: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

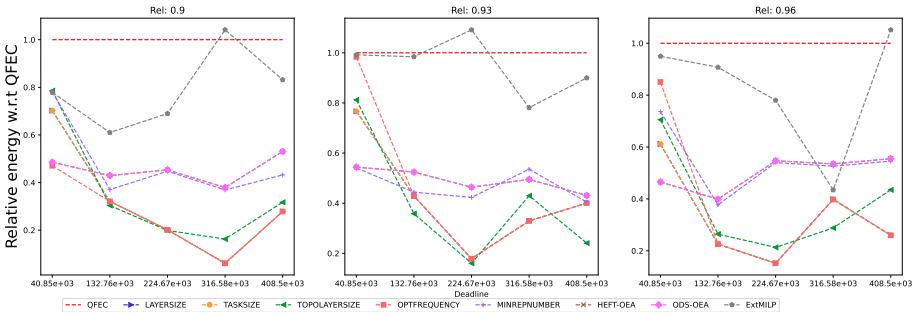

Figure 1577: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

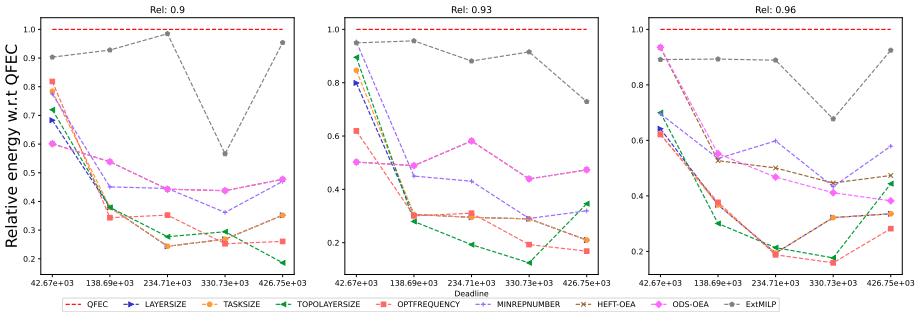

Figure 1578: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

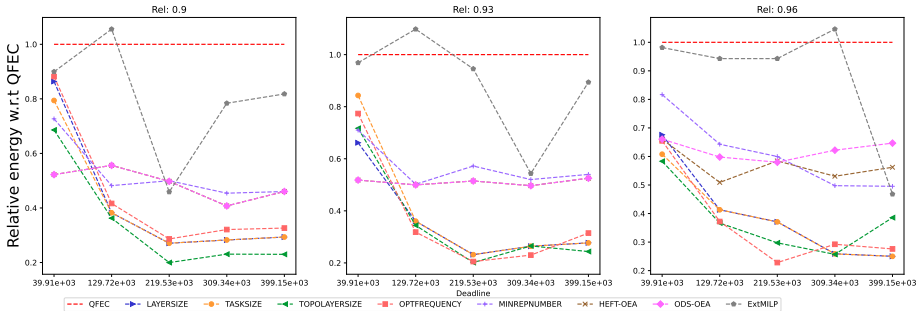

Figure 1579: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

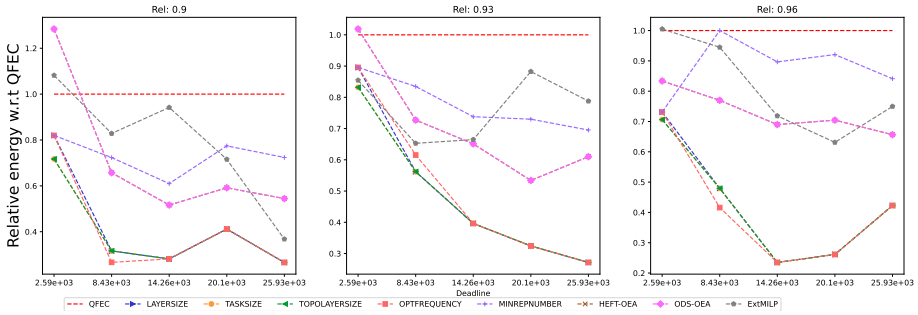

Figure 1580: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

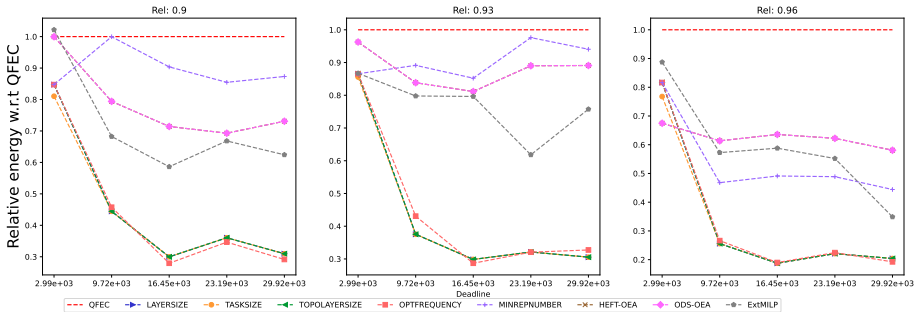

Figure 1581: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.1.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

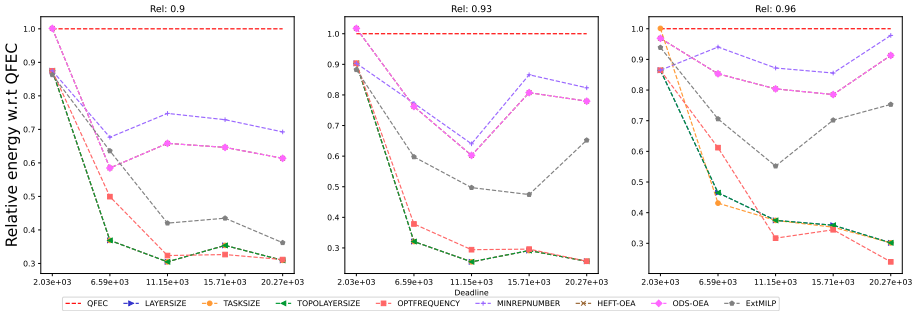

Figure 1582: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

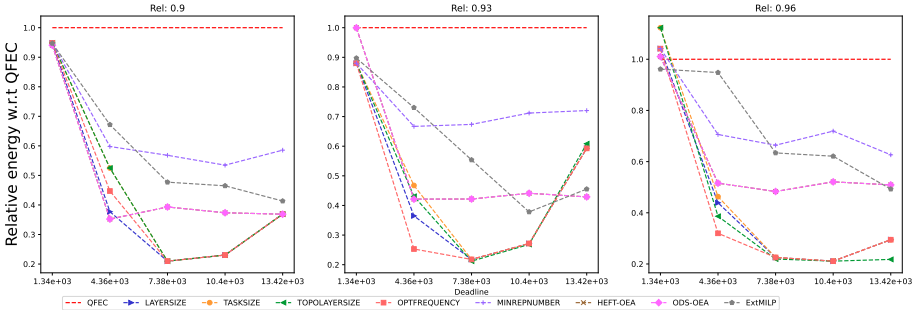

Figure 1583: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

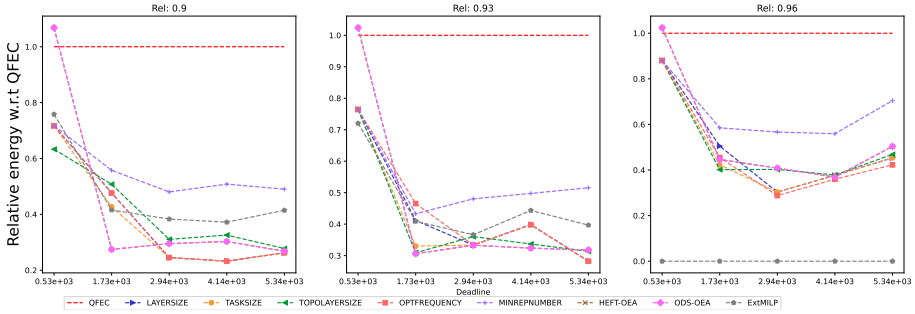

Figure 1584: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

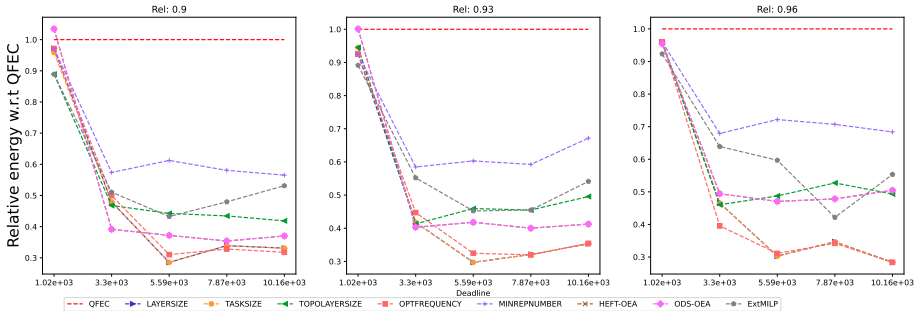

Figure 1585: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

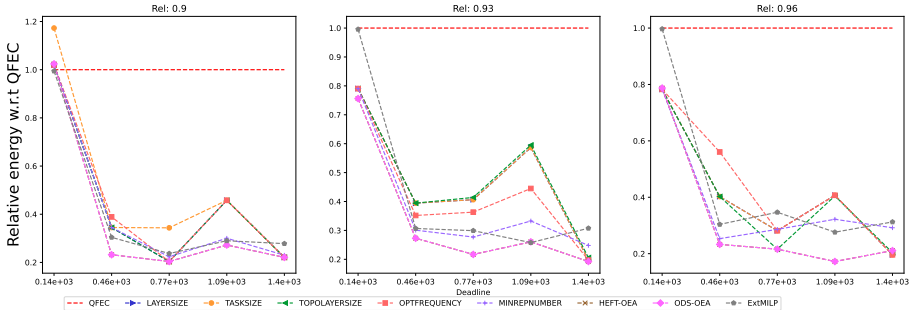

Figure 1586: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

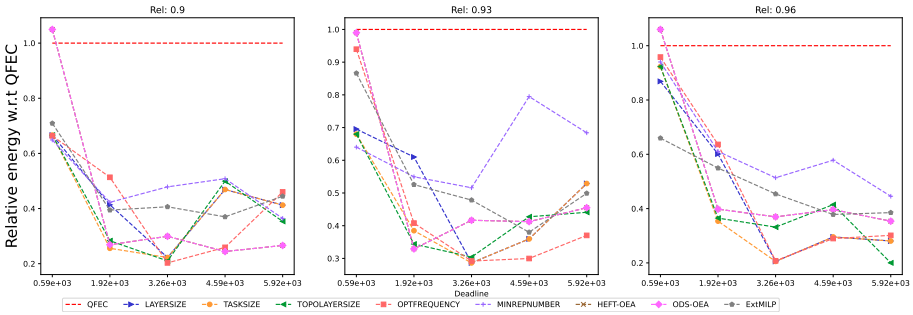

Figure 1587: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

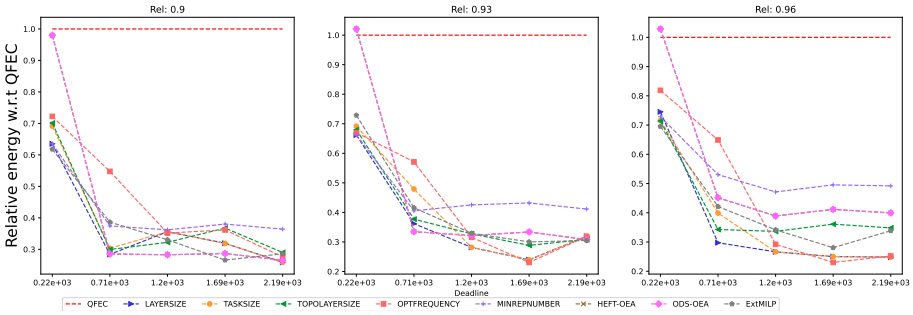

Figure 1588: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

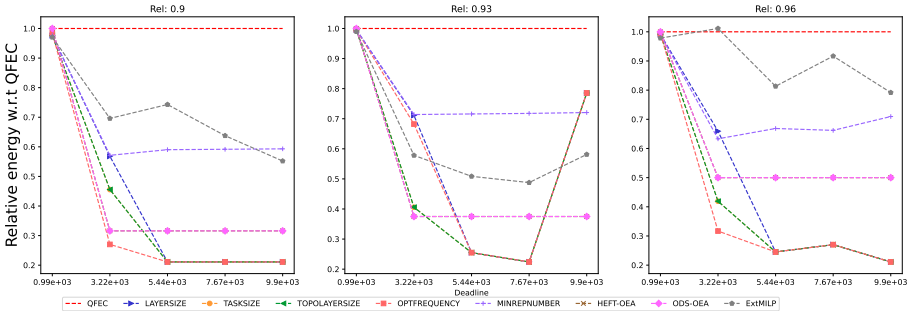

Figure 1589: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

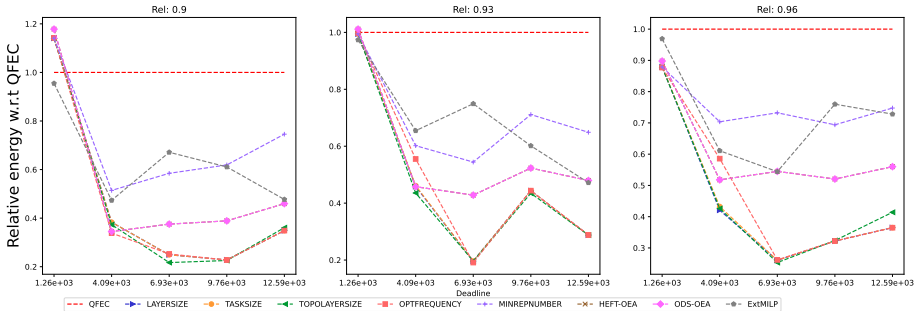

Figure 1590: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

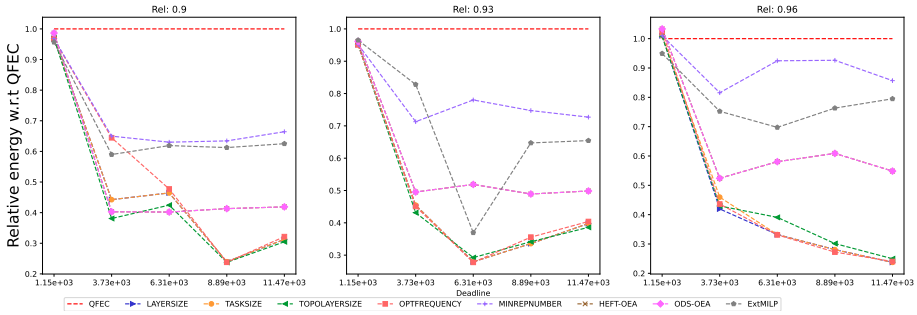

Figure 1591: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

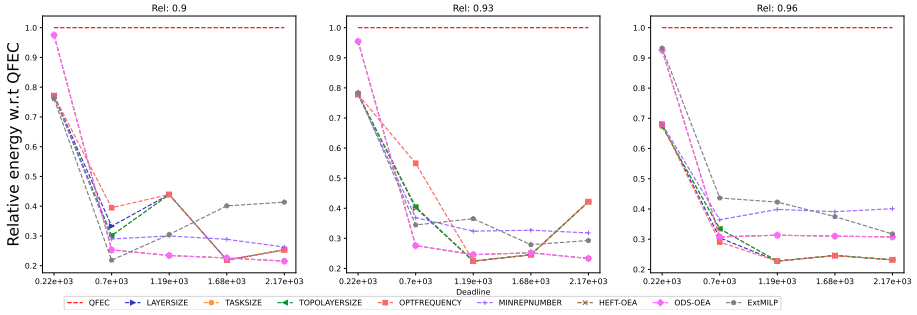

Figure 1592: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

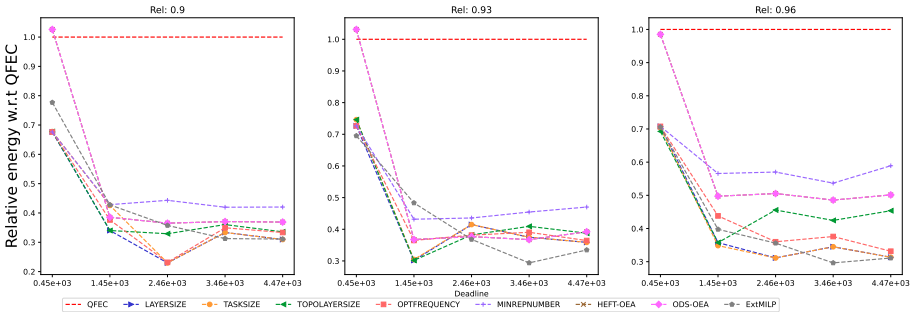

Figure 1593: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

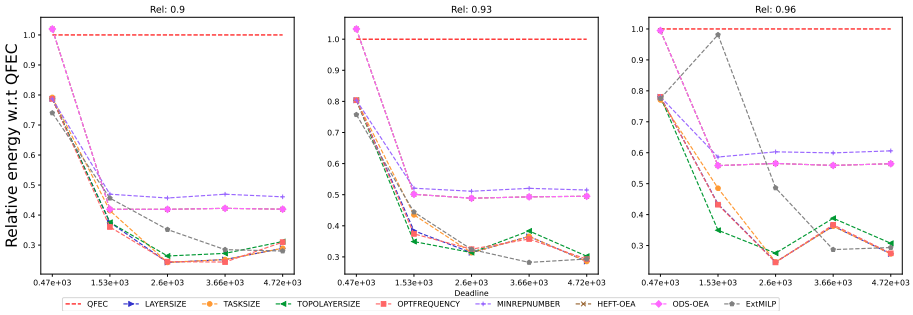

Figure 1594: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

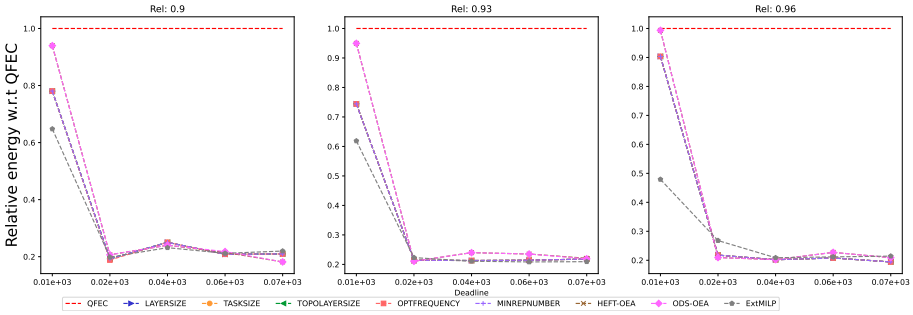

Figure 1595: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

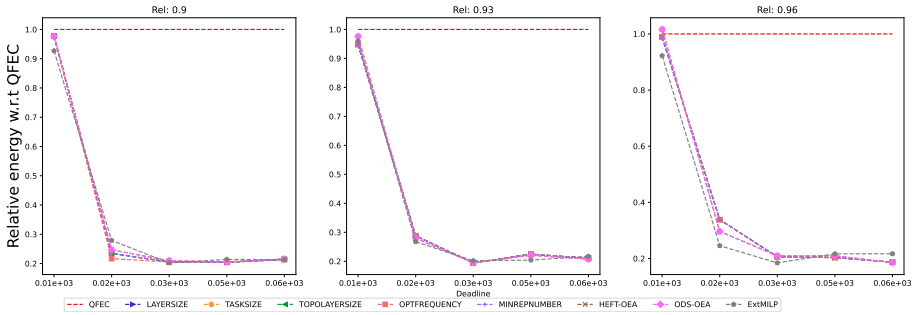

Figure 1596: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

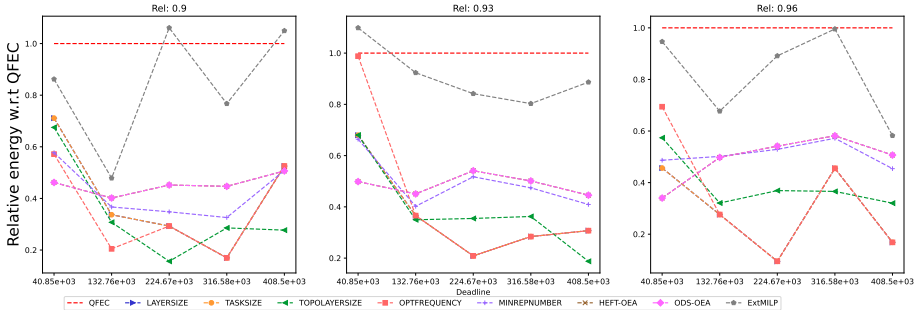

Figure 1597: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

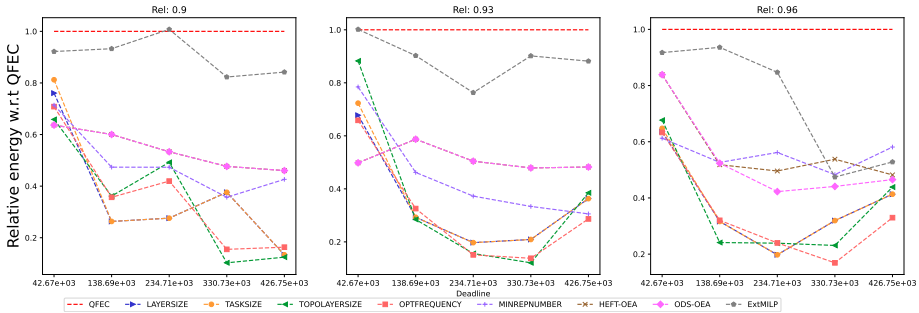

Figure 1598: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

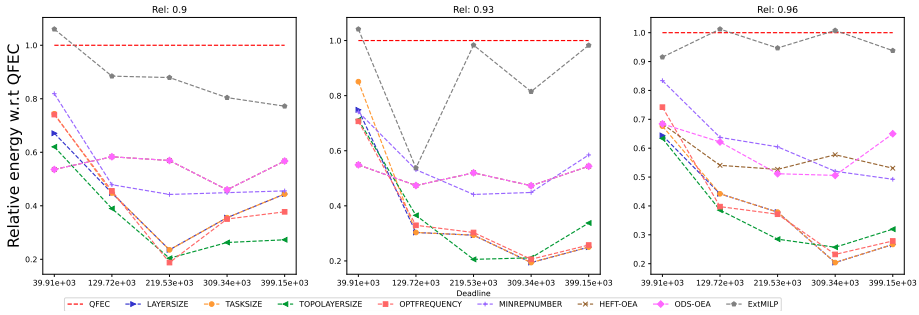

Figure 1599: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

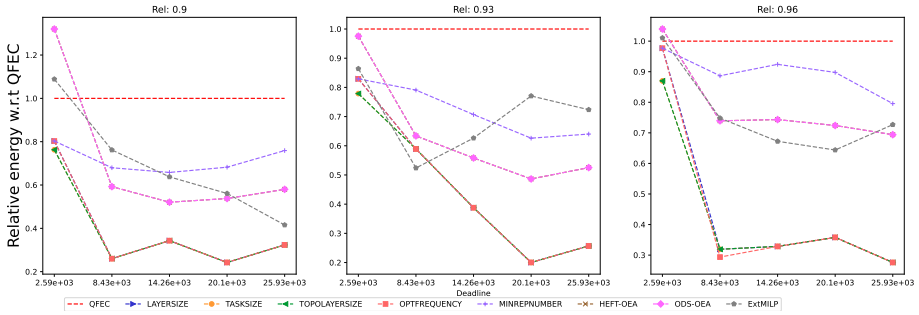

Figure 1600: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

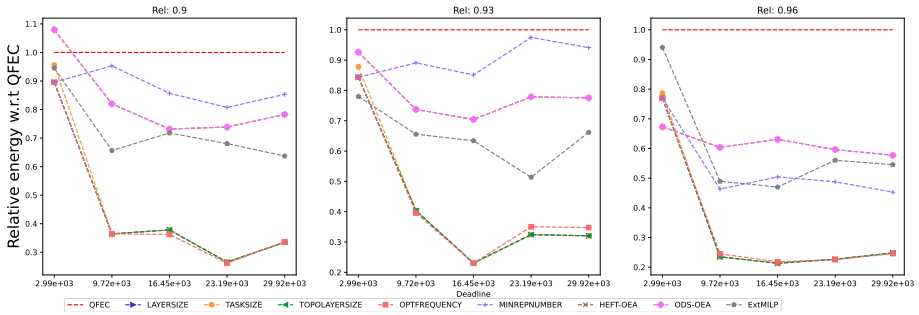

Figure 1601: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.2 $BC/WC = 0.2$

G.2.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

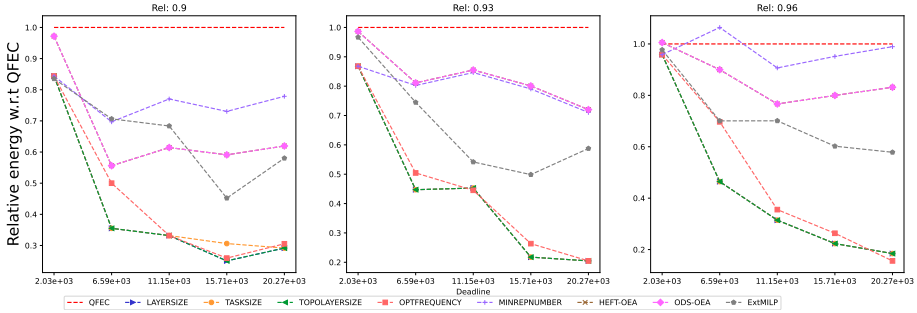

Figure 1602: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

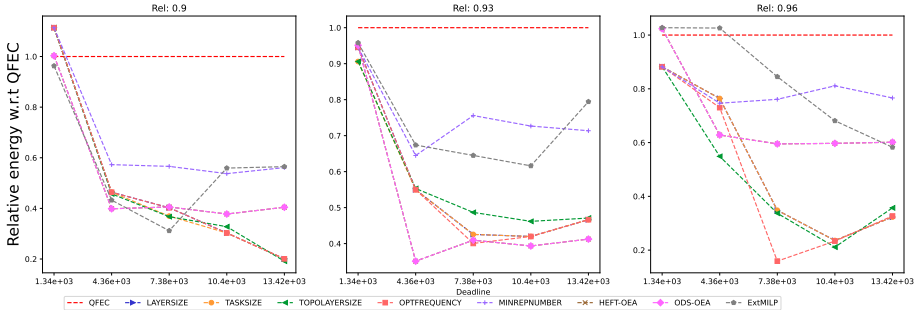

Figure 1603: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

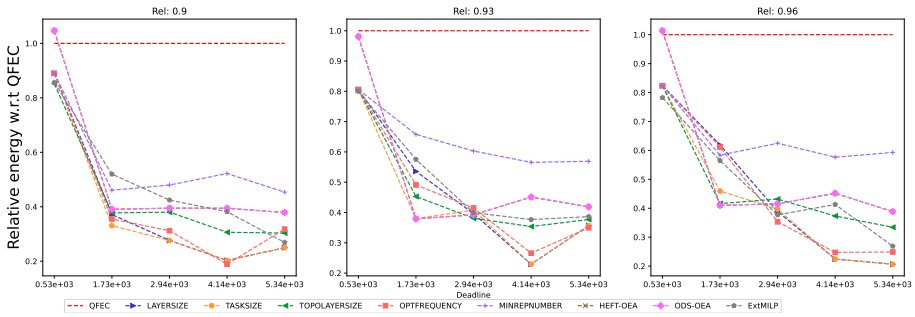

Figure 1604: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

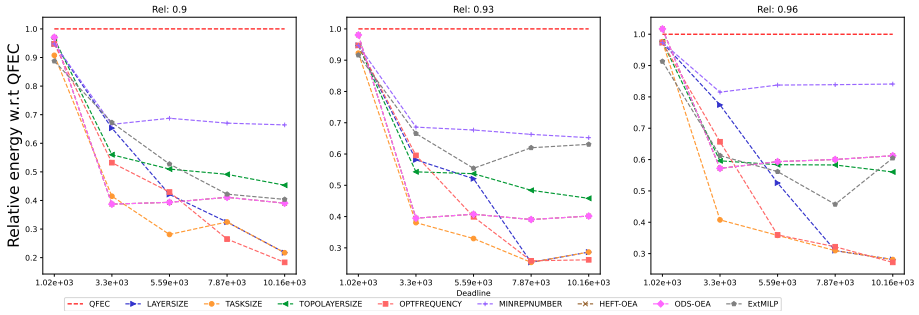

Figure 1605: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

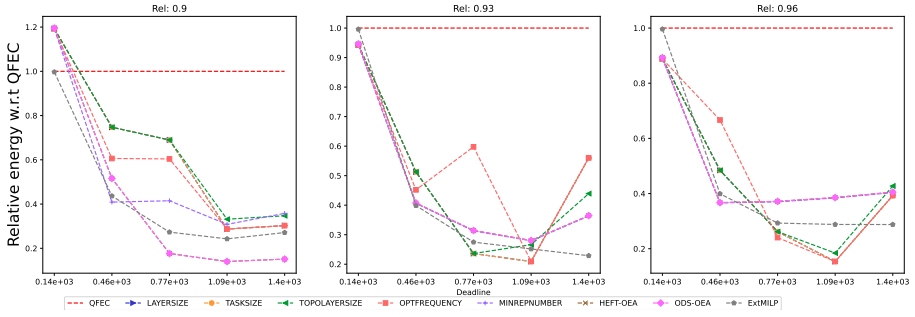

Figure 1606: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

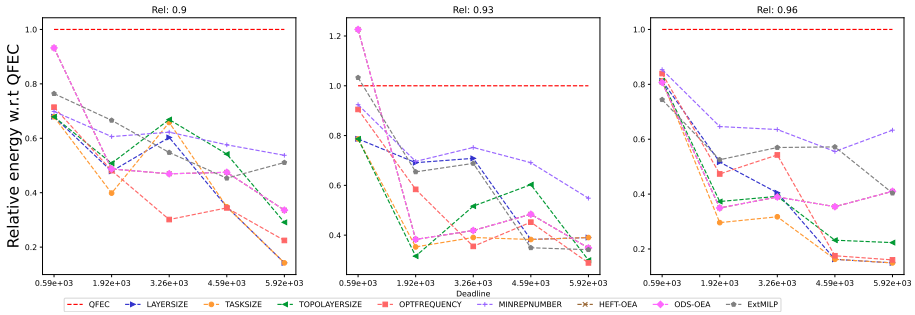

Figure 1607: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

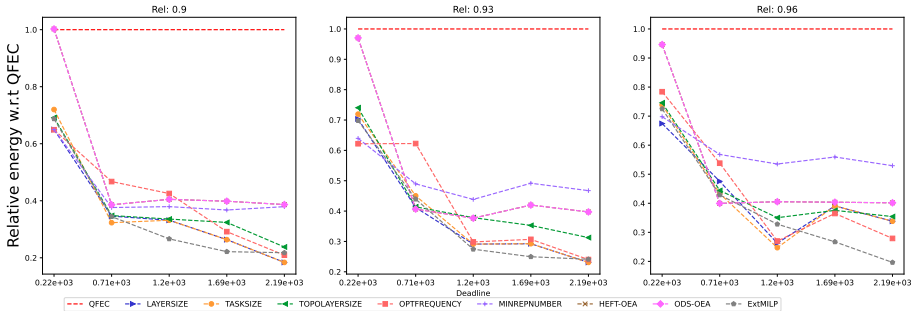

Figure 1608: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

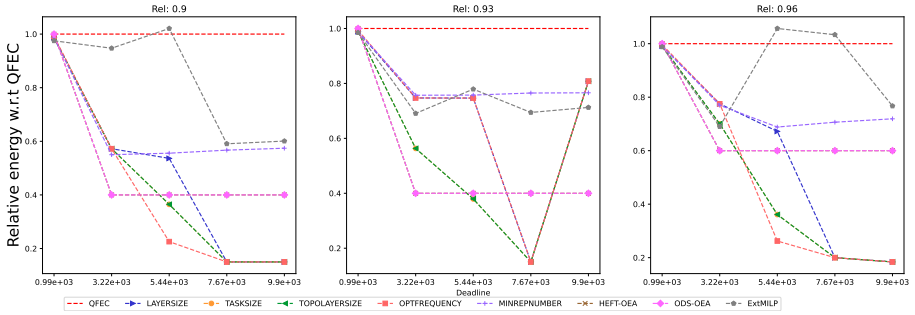

Figure 1609: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

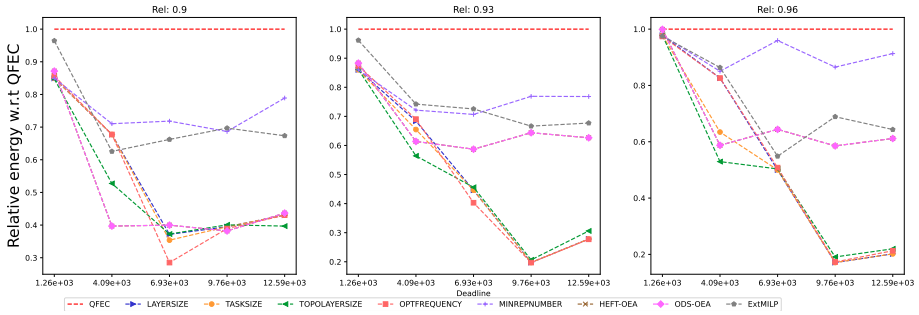

Figure 1610: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

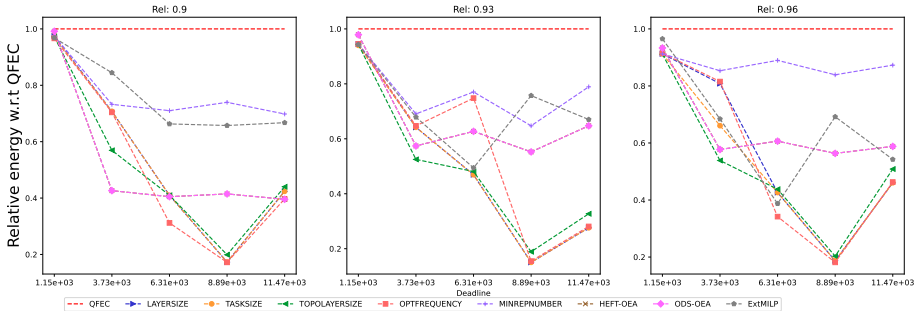

Figure 1611: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

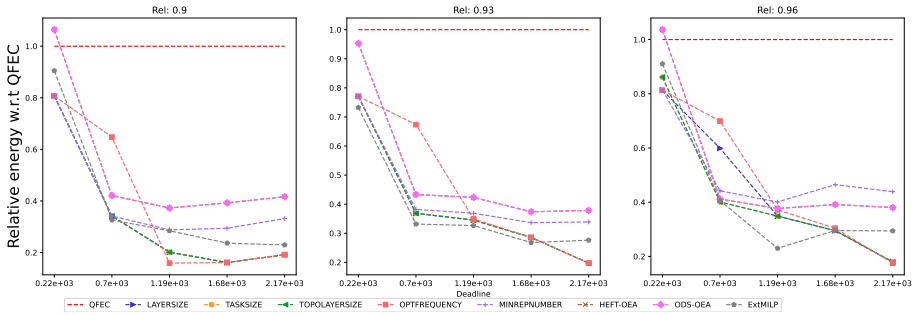

Figure 1612: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

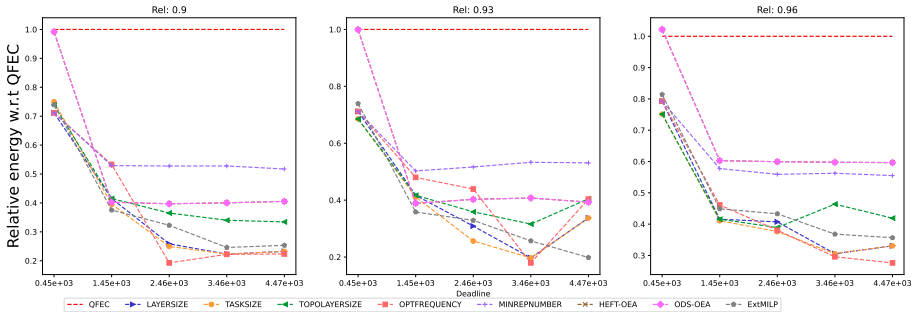

Figure 1613: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

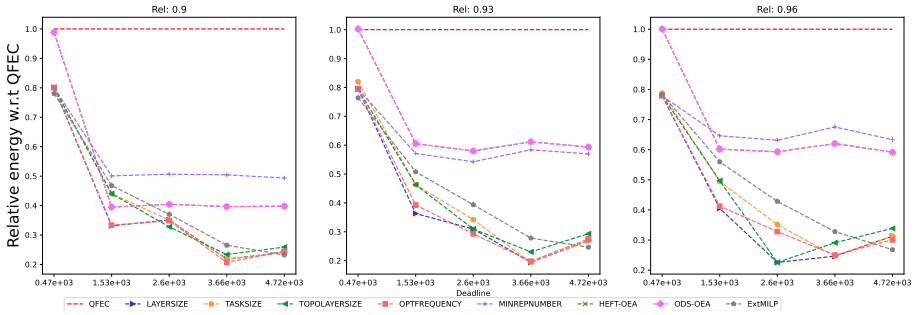

Figure 1614: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

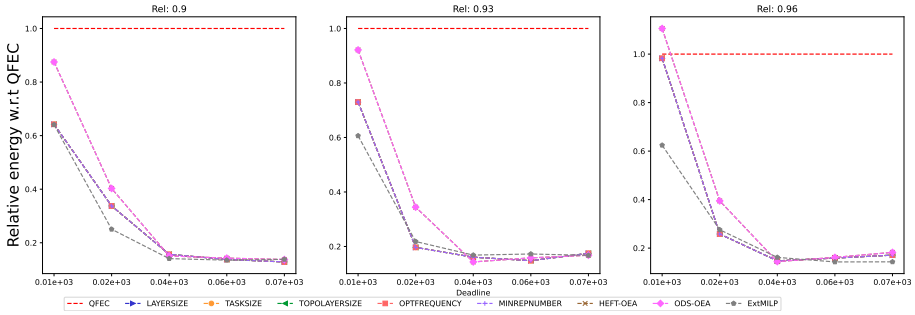

Figure 1615: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

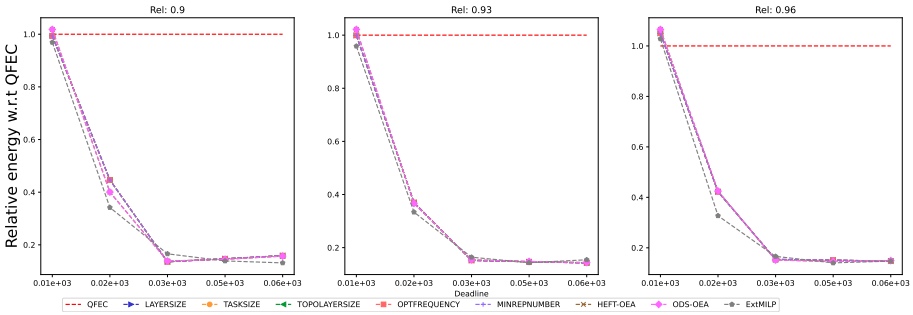

Figure 1616: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

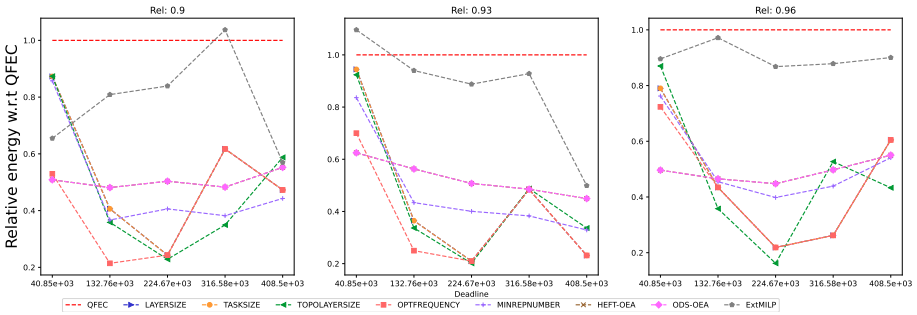

Figure 1617: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

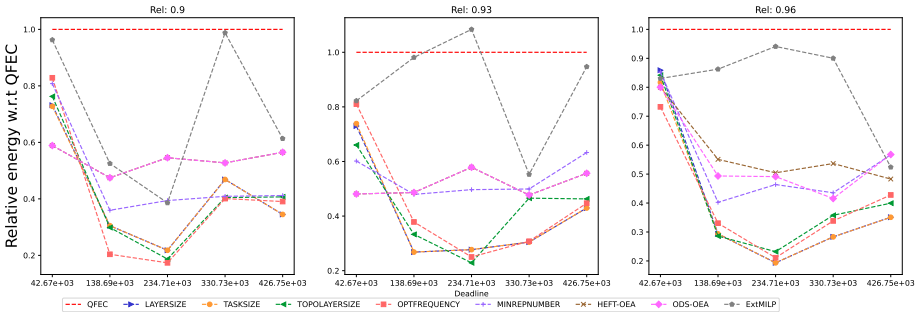

Figure 1618: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

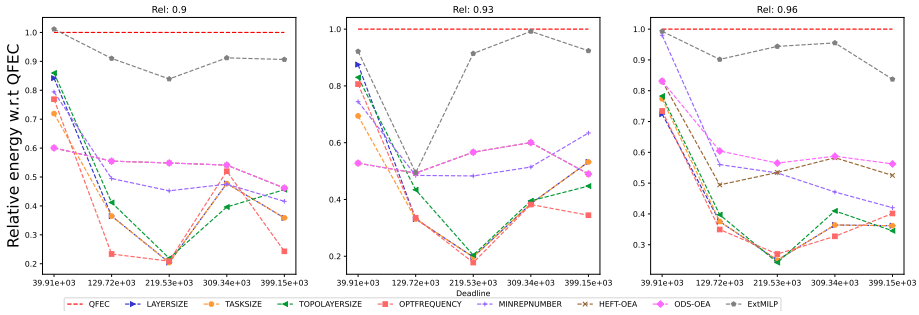

Figure 1619: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

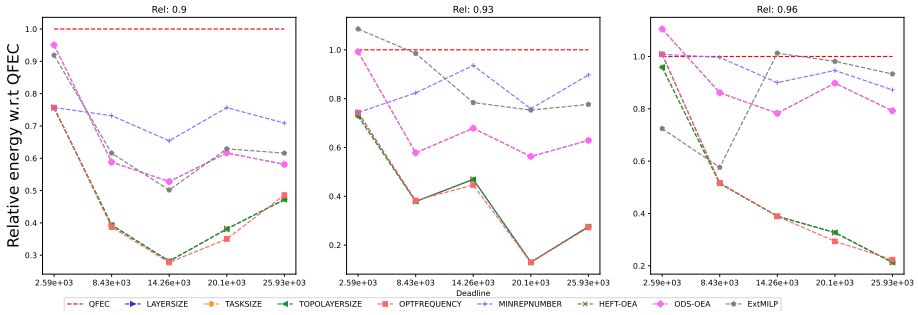

Figure 1620: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

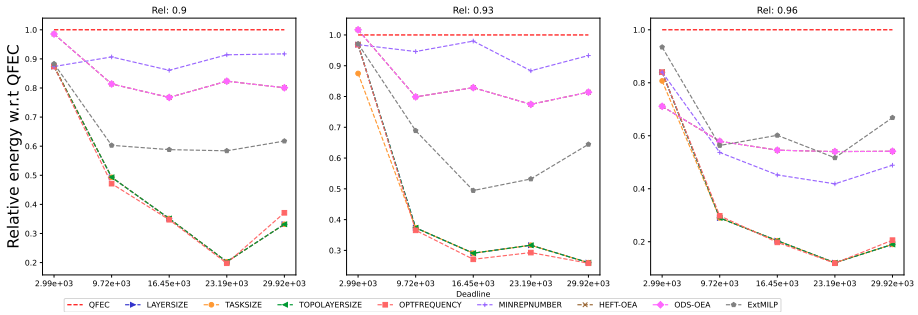

Figure 1621: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.2.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

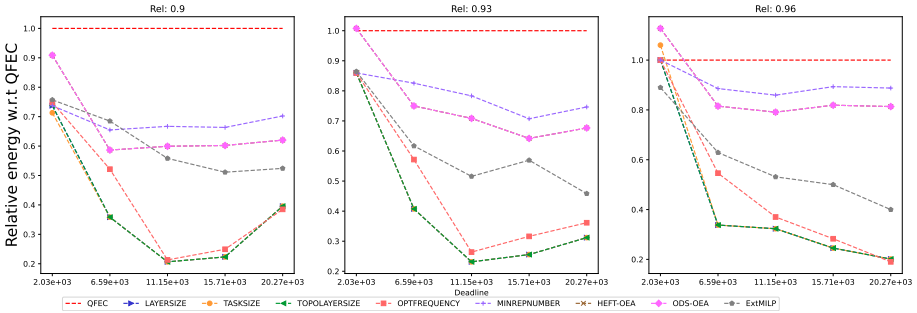

Figure 1622: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

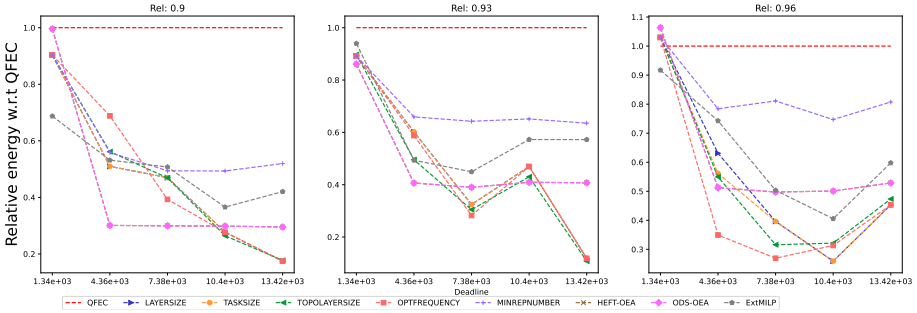

Figure 1623: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

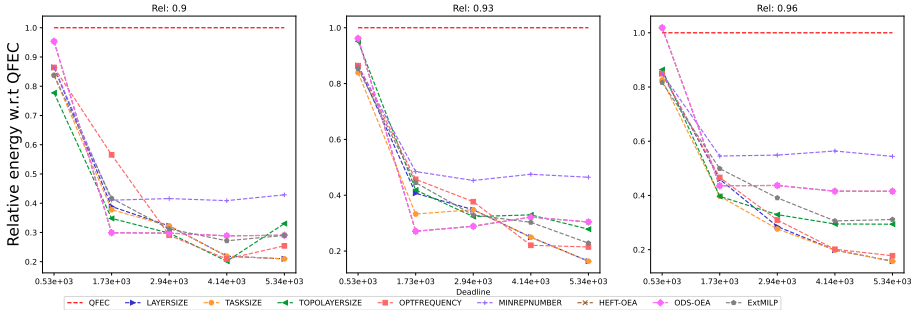

Figure 1624: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

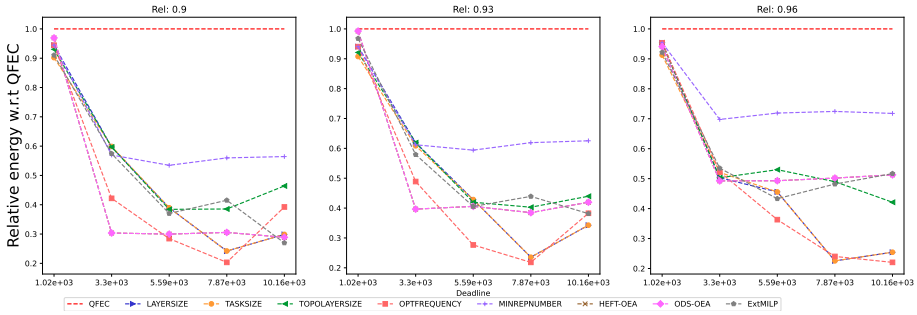

Figure 1625: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

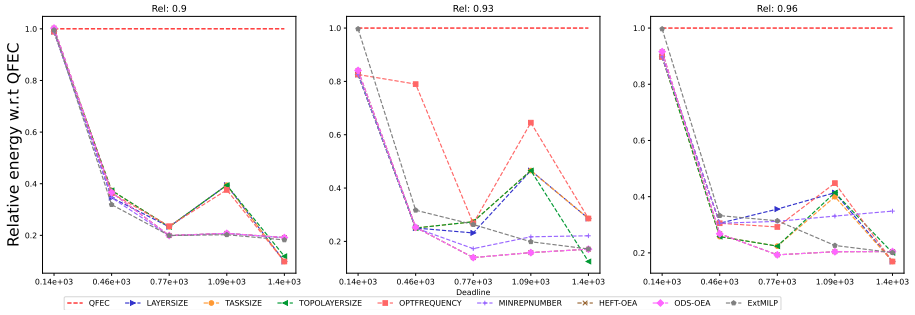

Figure 1626: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

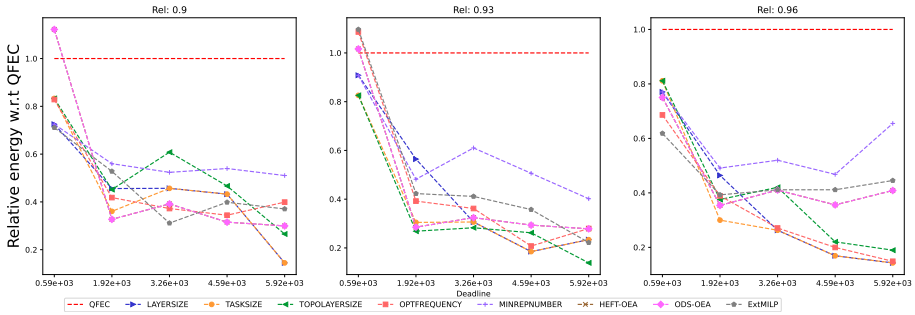

Figure 1627: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

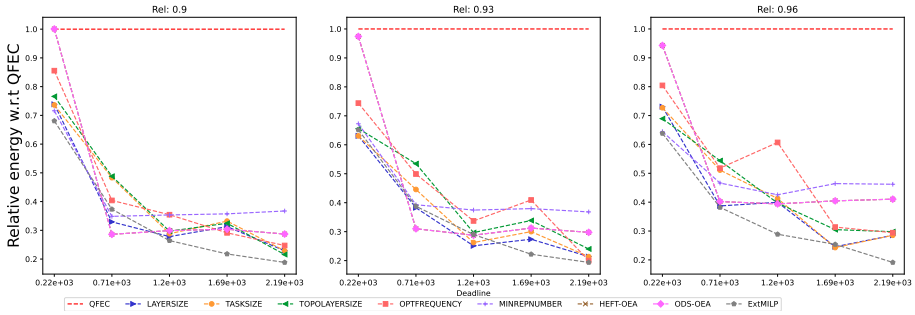

Figure 1628: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

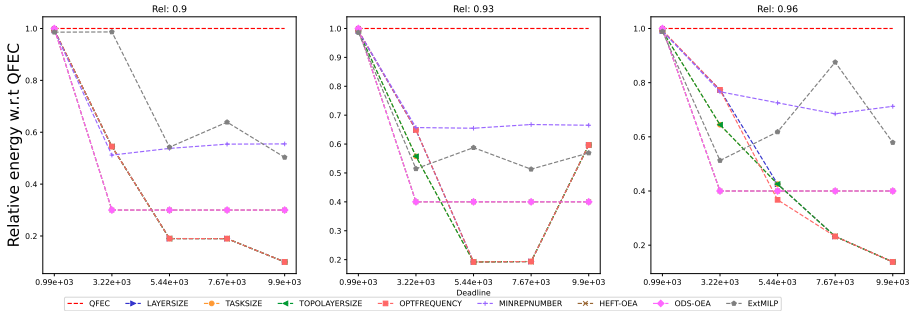

Figure 1629: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

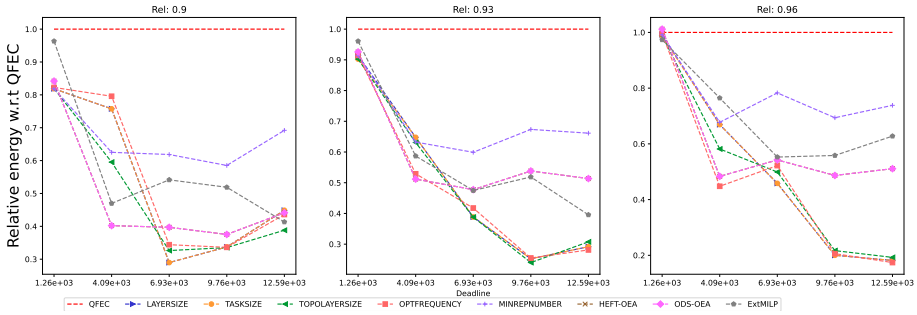

Figure 1630: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

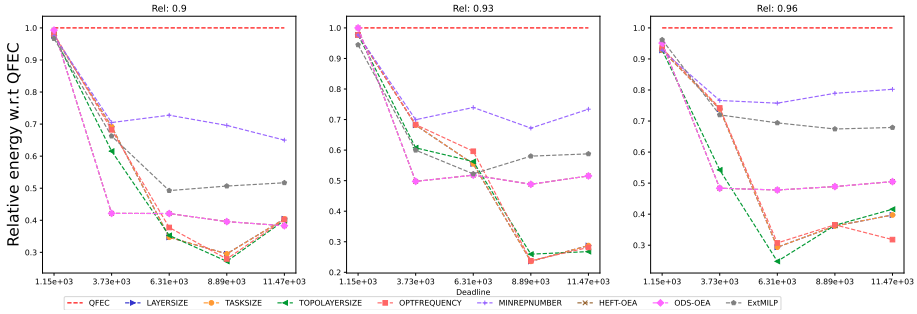

Figure 1631: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

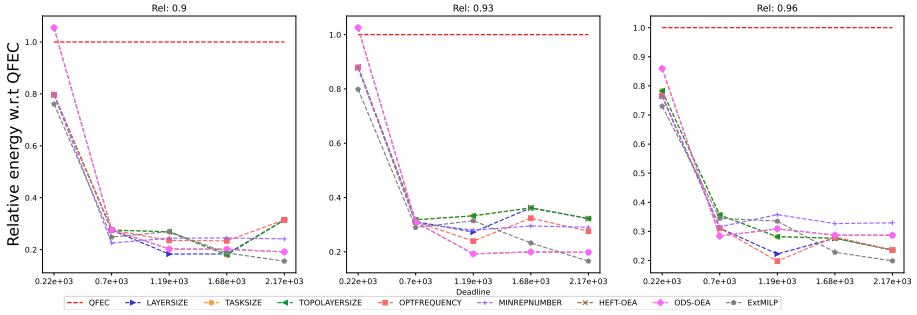

Figure 1632: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

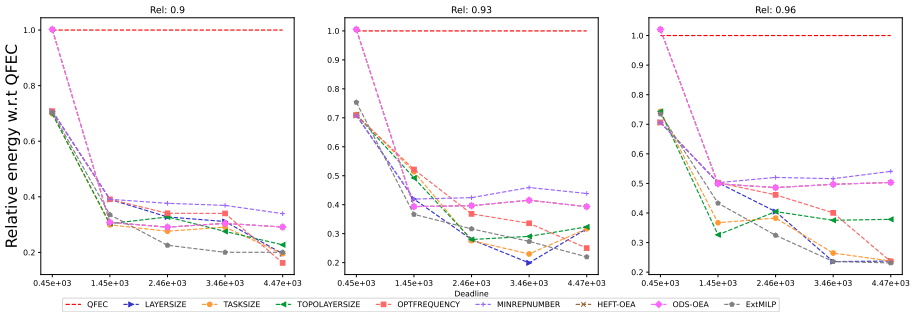

Figure 1633: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

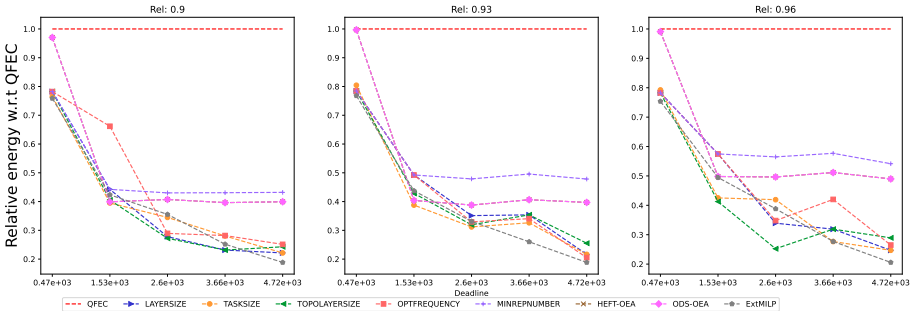

Figure 1634: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

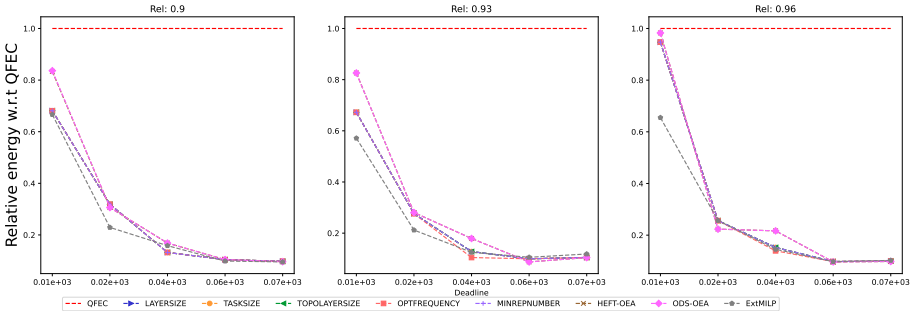

Figure 1635: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

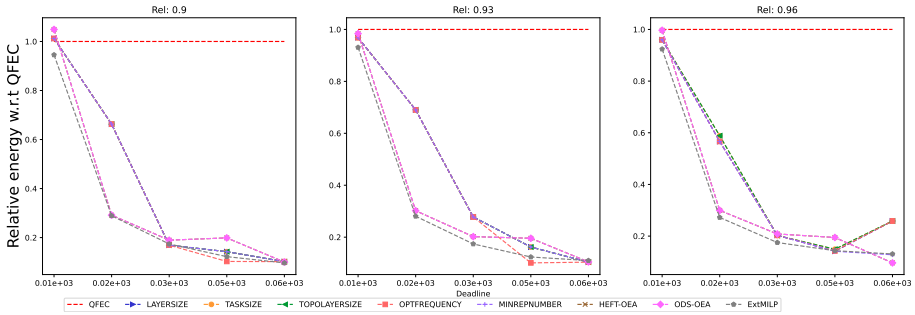

Figure 1636: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

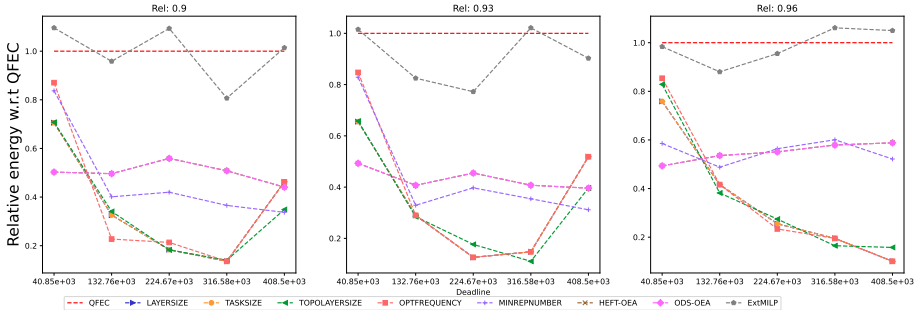

Figure 1637: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

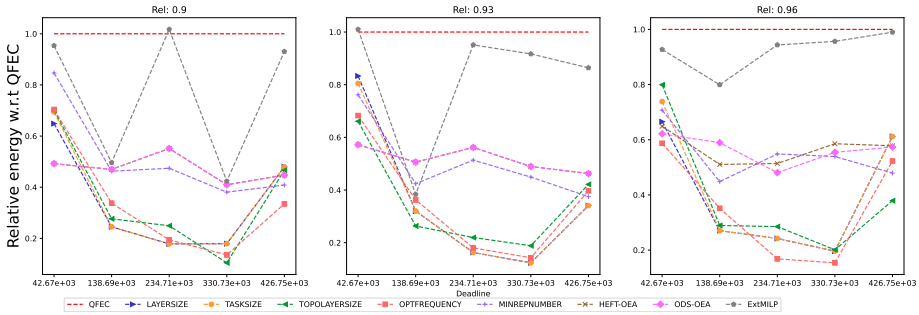

Figure 1638: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

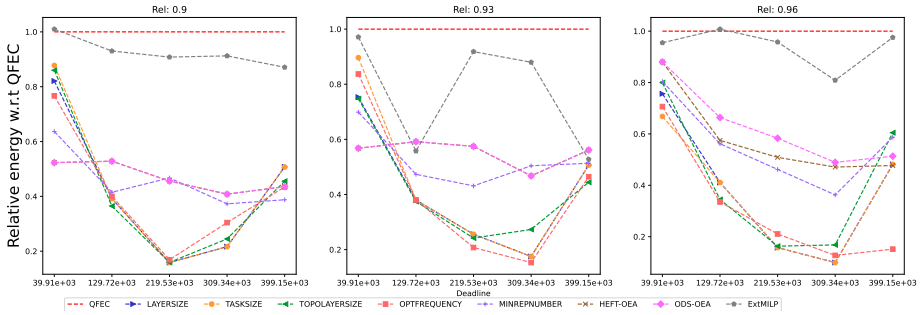

Figure 1639: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

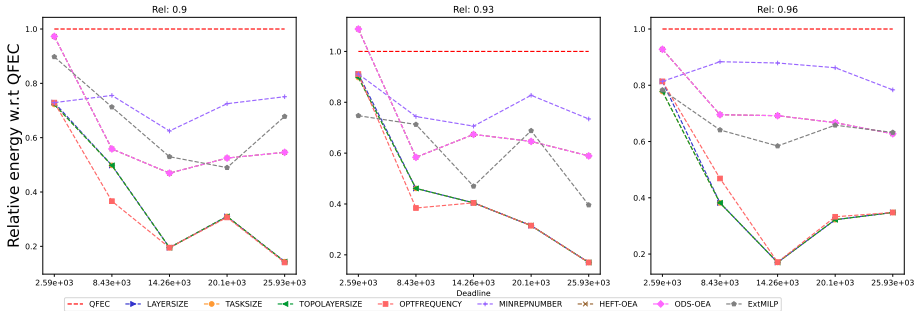

Figure 1640: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

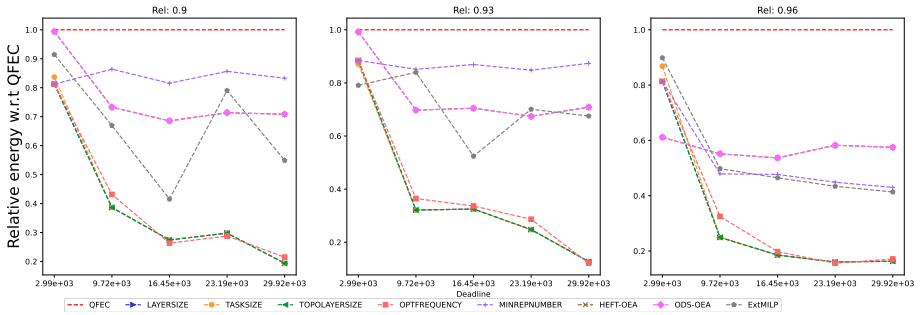

Figure 1641: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.2.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

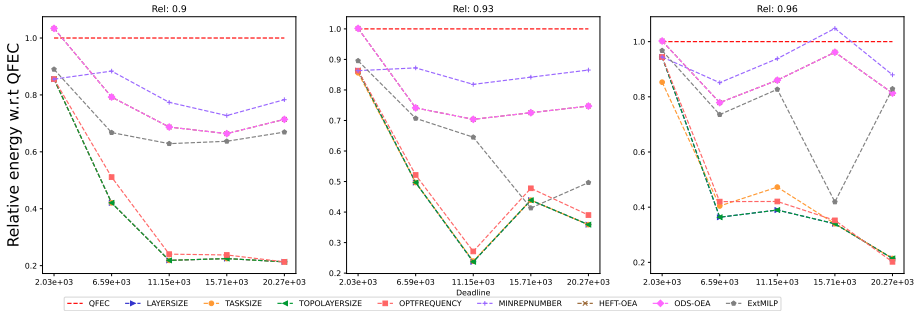

Figure 1642: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

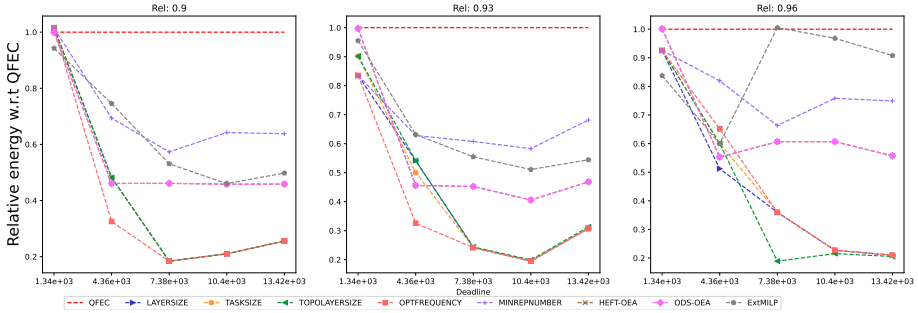

Figure 1643: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

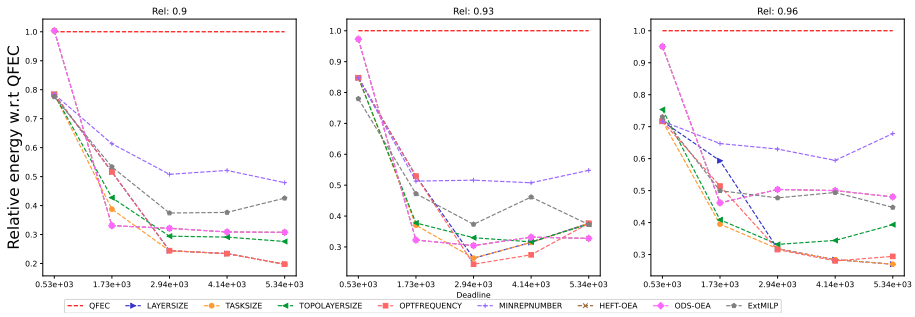

Figure 1644: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

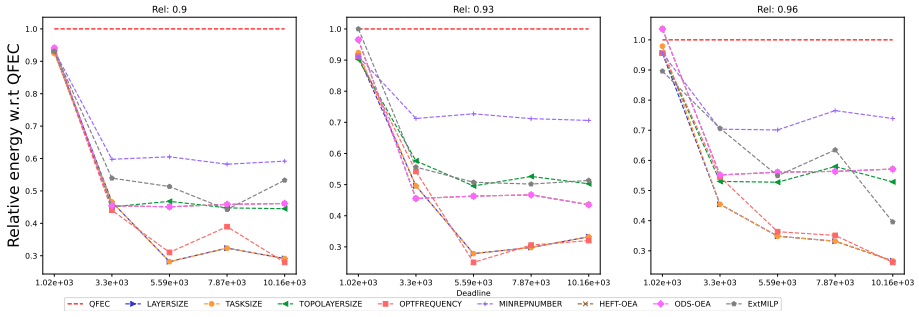

Figure 1645: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

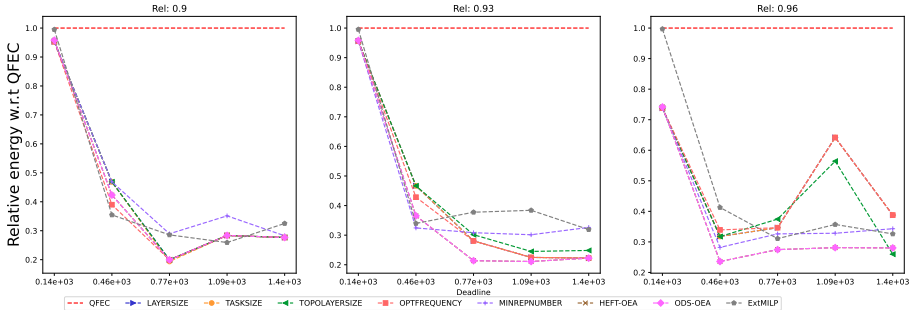

Figure 1646: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

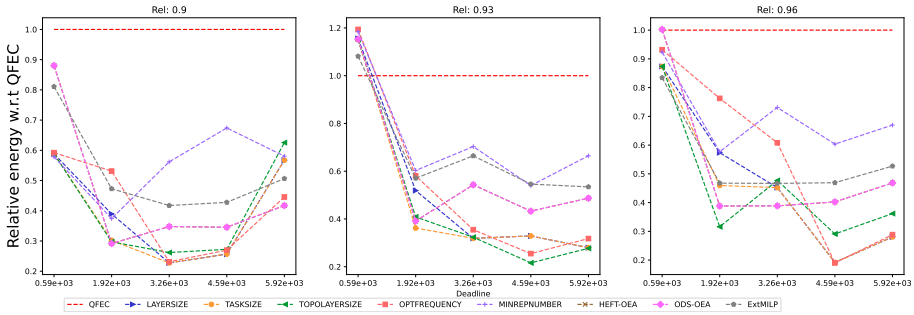

Figure 1647: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

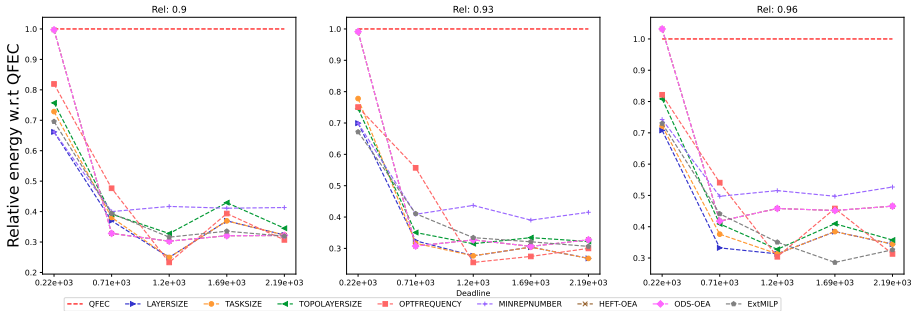

Figure 1648: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

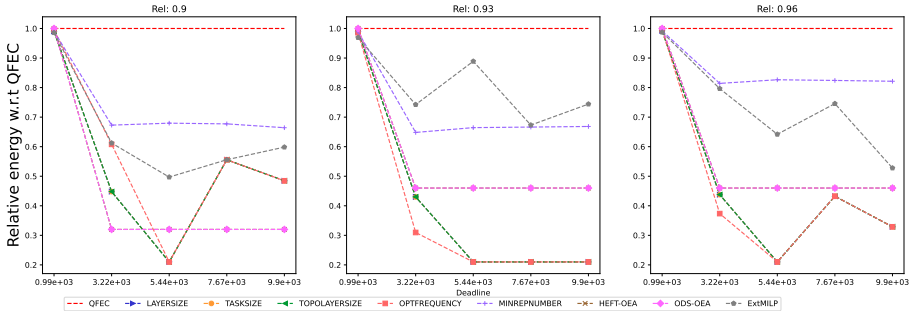

Figure 1649: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

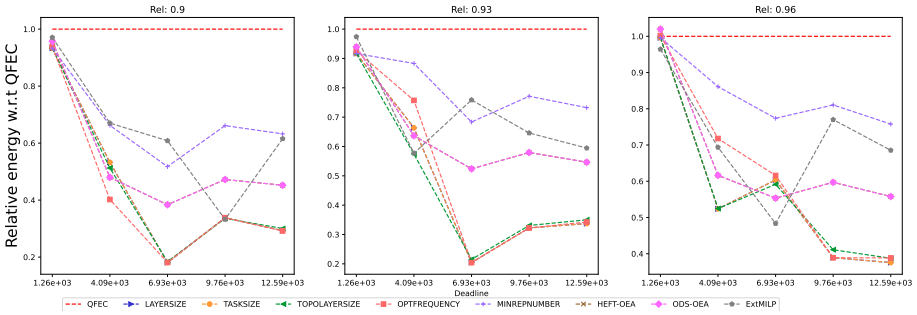

Figure 1650: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

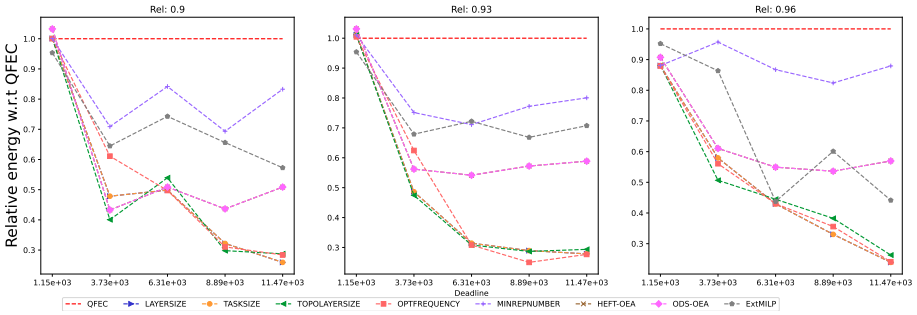

Figure 1651: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

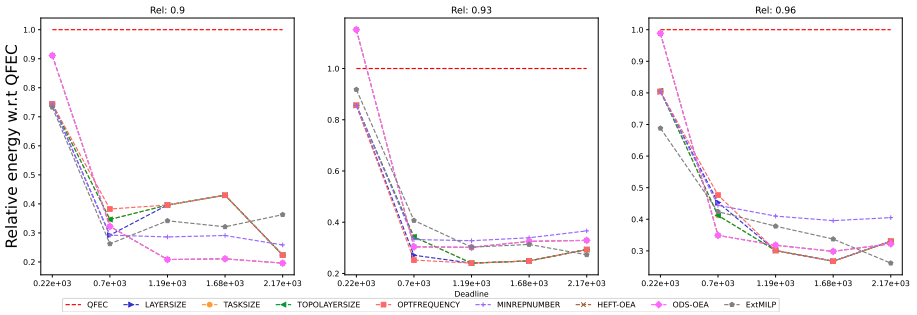

Figure 1652: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

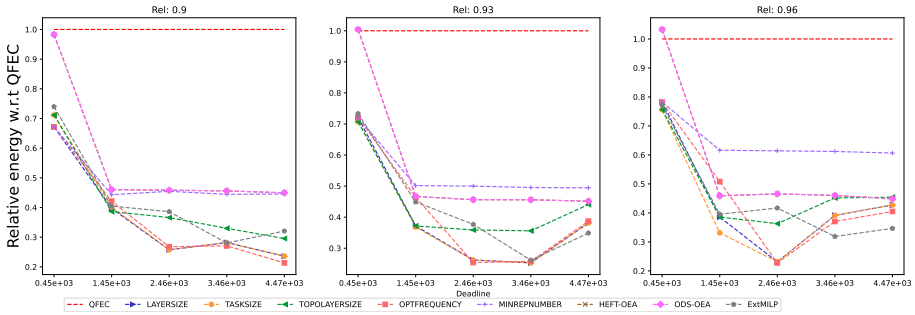

Figure 1653: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

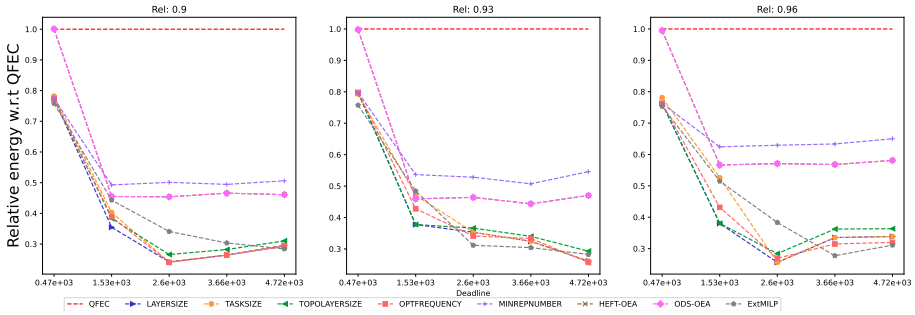

Figure 1654: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

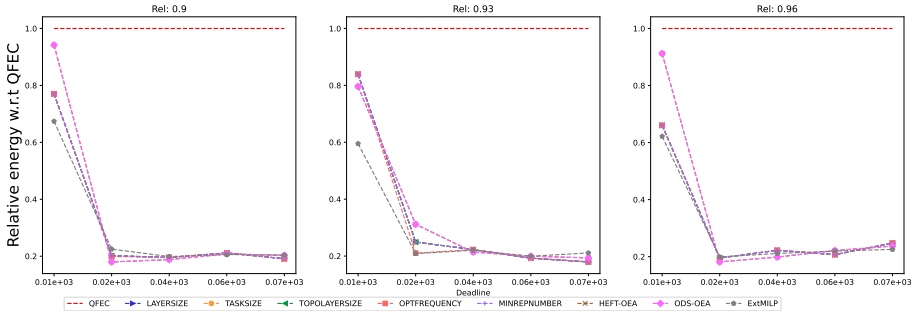

Figure 1655: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

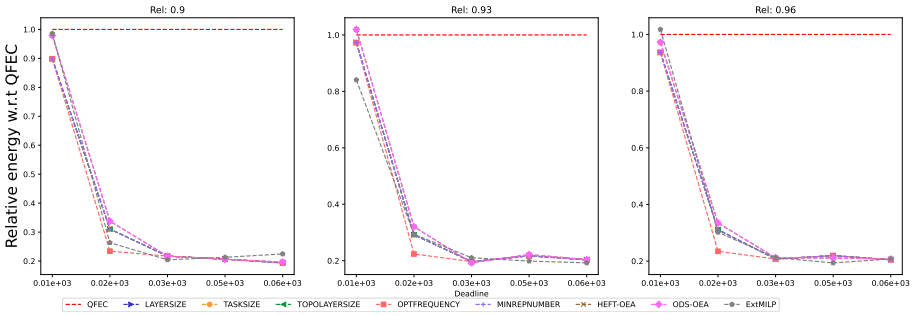

Figure 1656: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

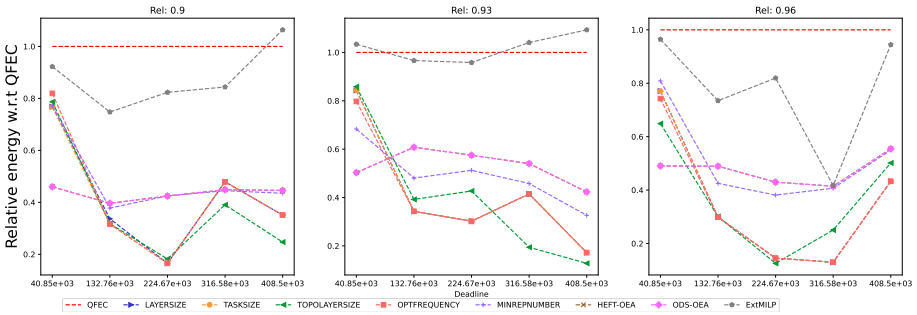

Figure 1657: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

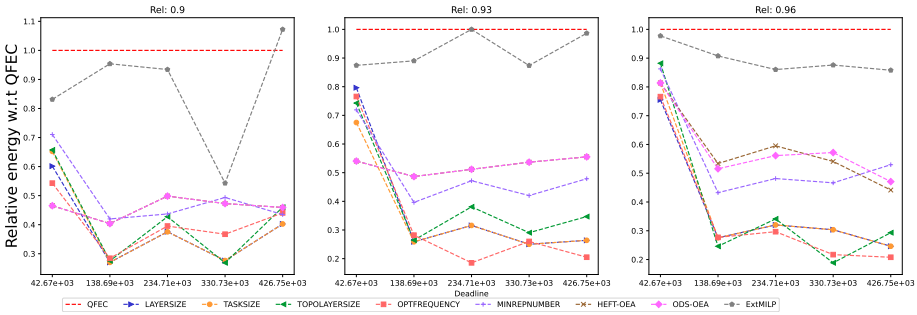

Figure 1658: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

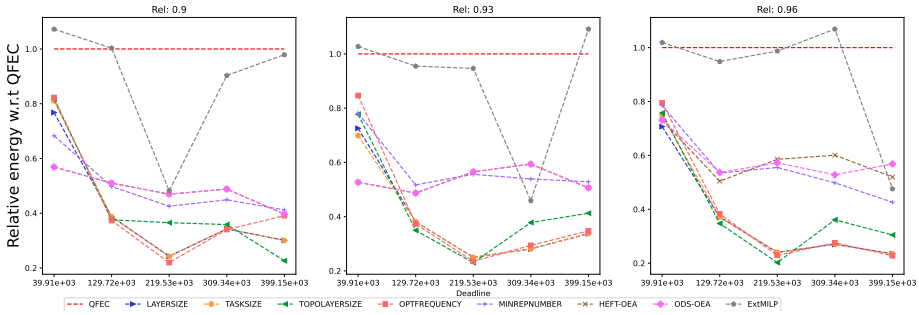

Figure 1659: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

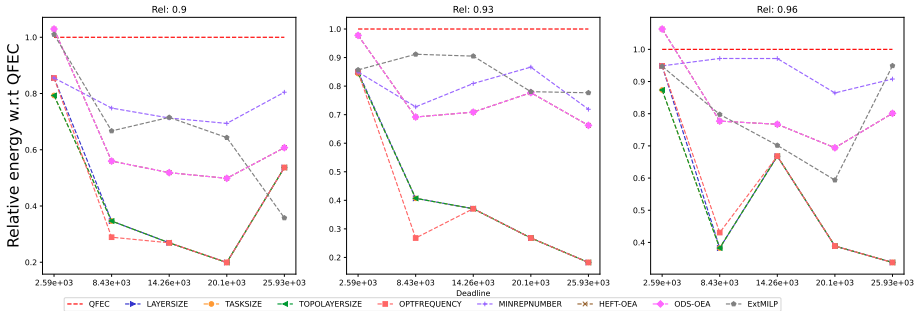

Figure 1660: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

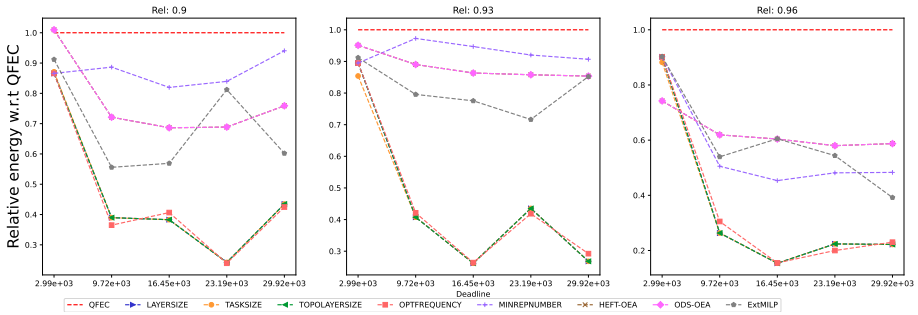

Figure 1661: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.2.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

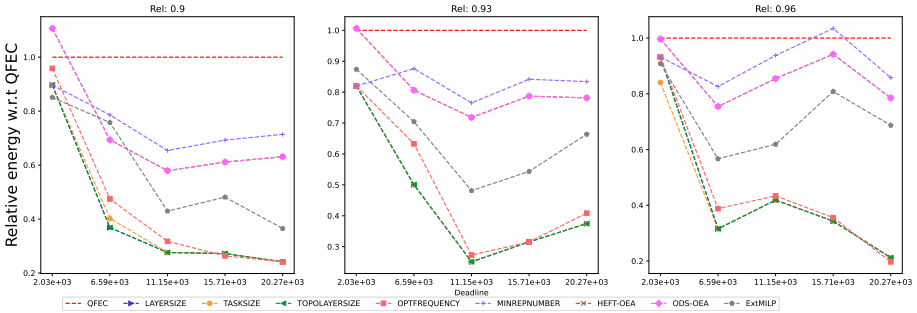

Figure 1662: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

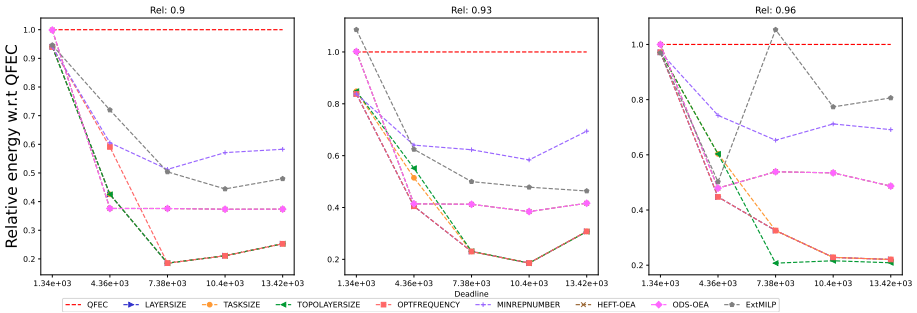

Figure 1663: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

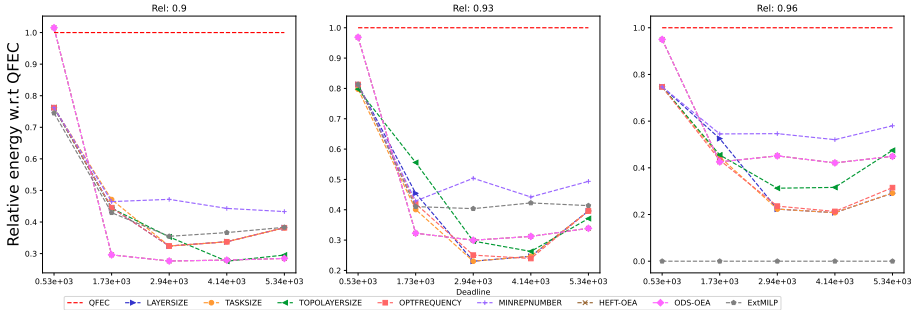

Figure 1664: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

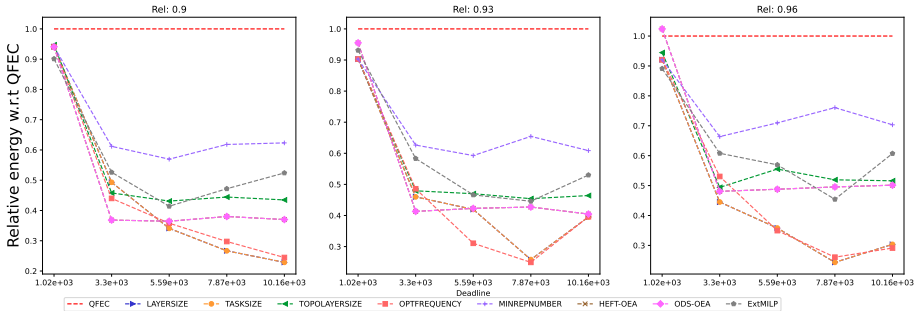

Figure 1665: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

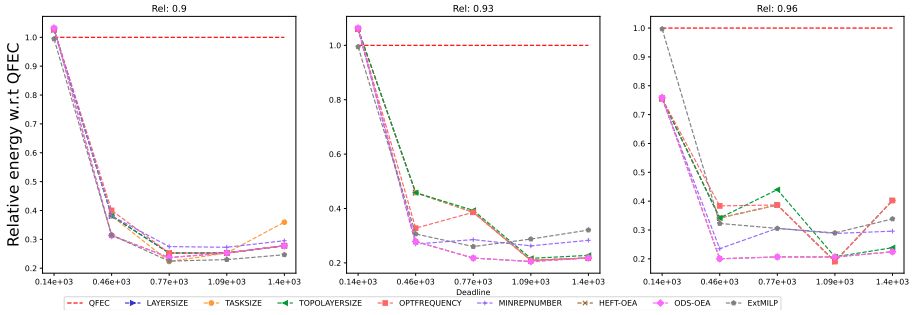

Figure 1666: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

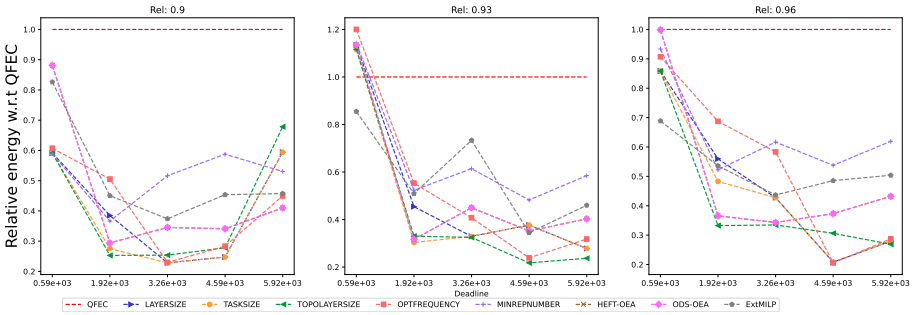

Figure 1667: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

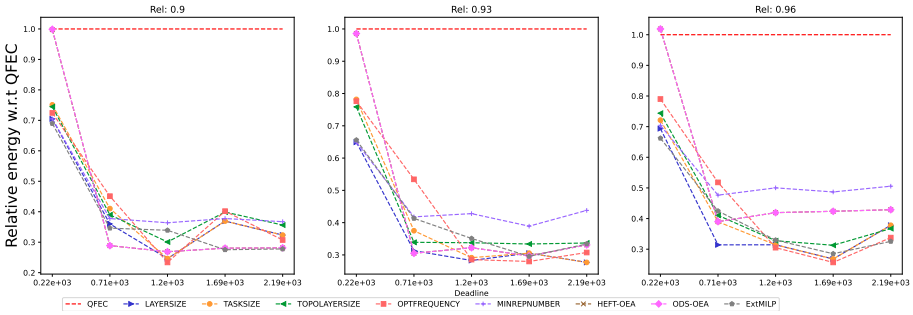

Figure 1668: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

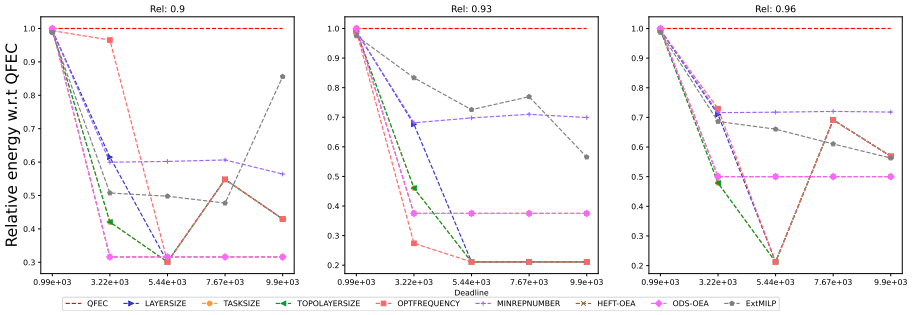

Figure 1669: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

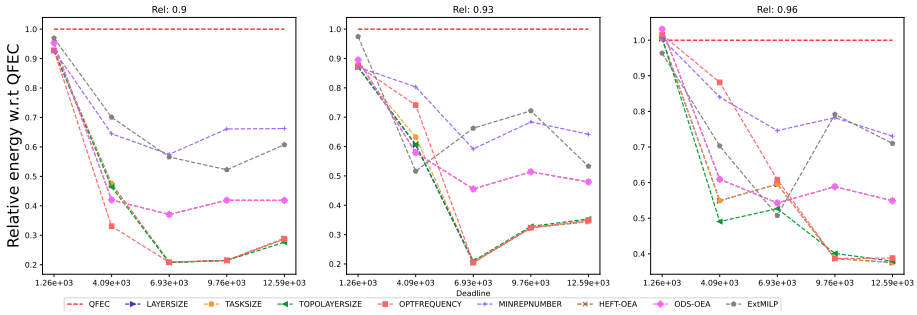

Figure 1670: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

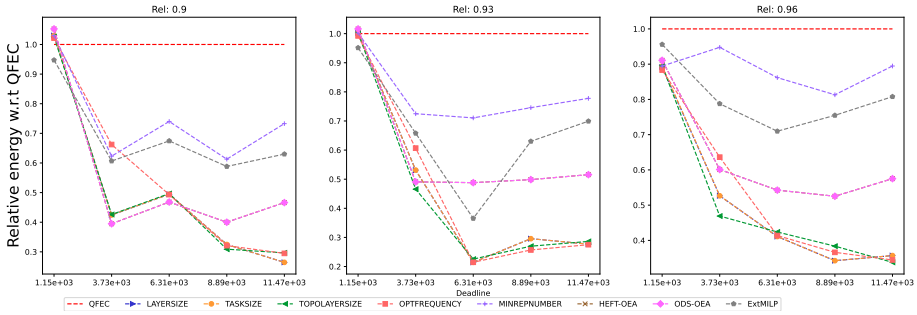

Figure 1671: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

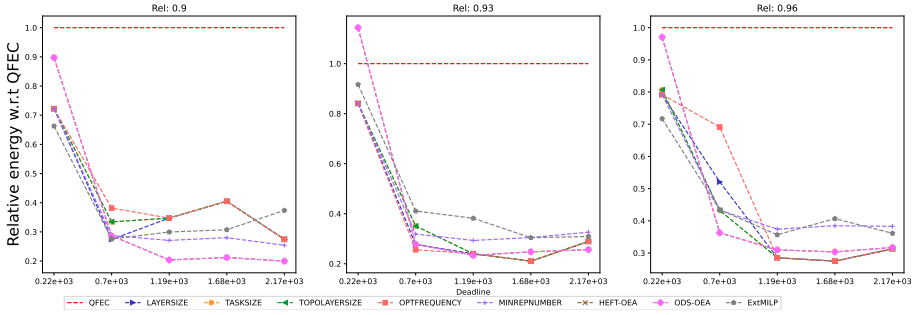

Figure 1672: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

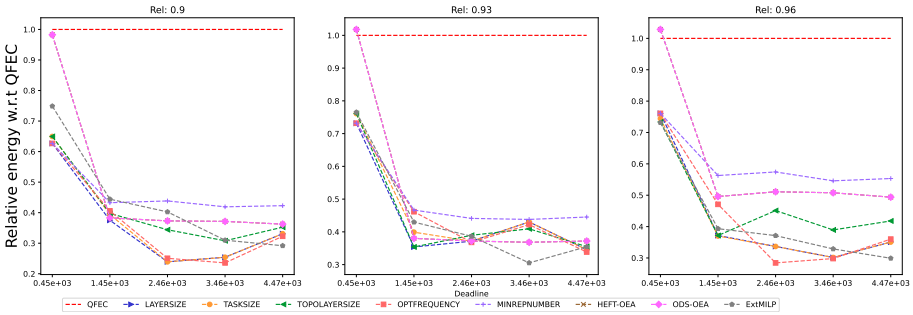

Figure 1673: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

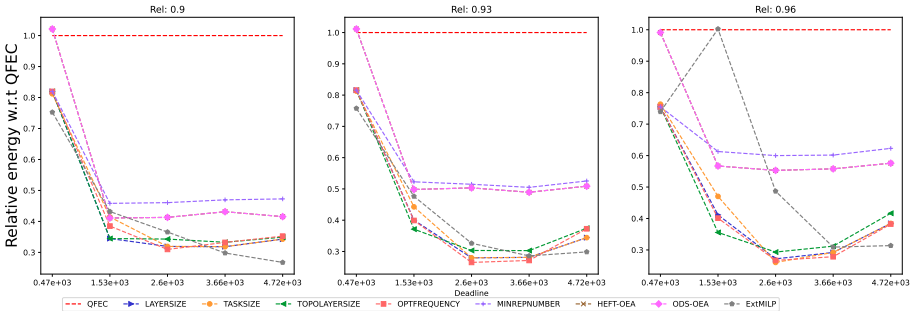

Figure 1674: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

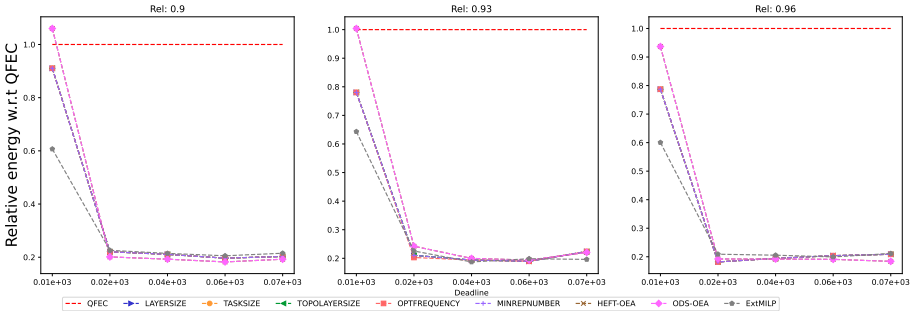

Figure 1675: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

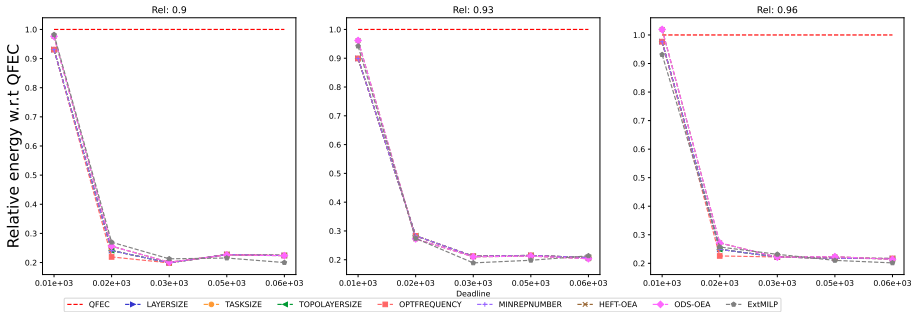

Figure 1676: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

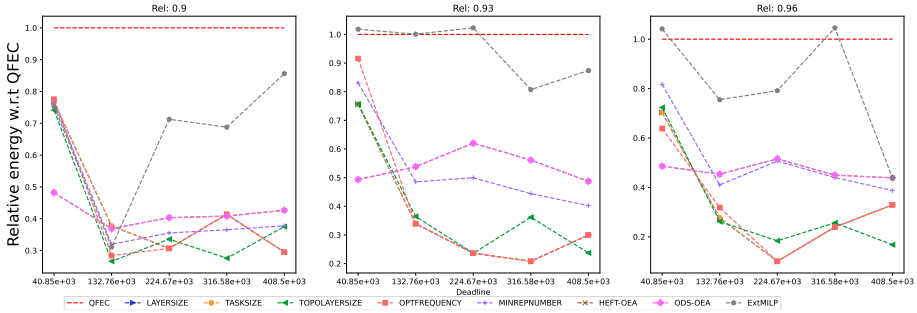

Figure 1677: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

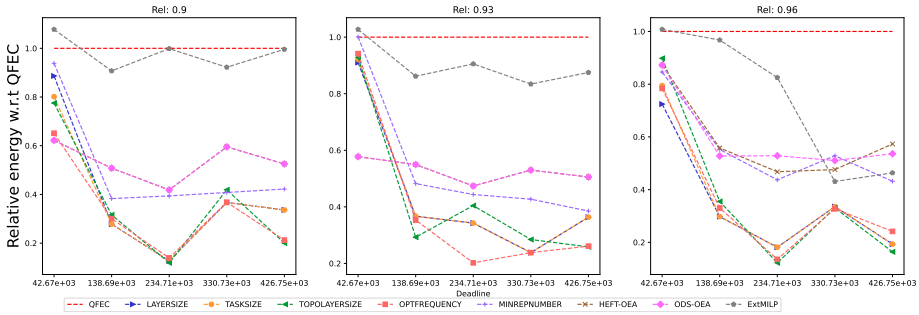

Figure 1678: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

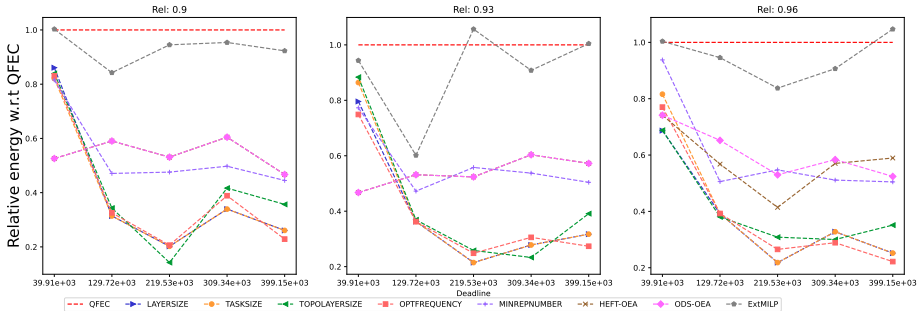

Figure 1679: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

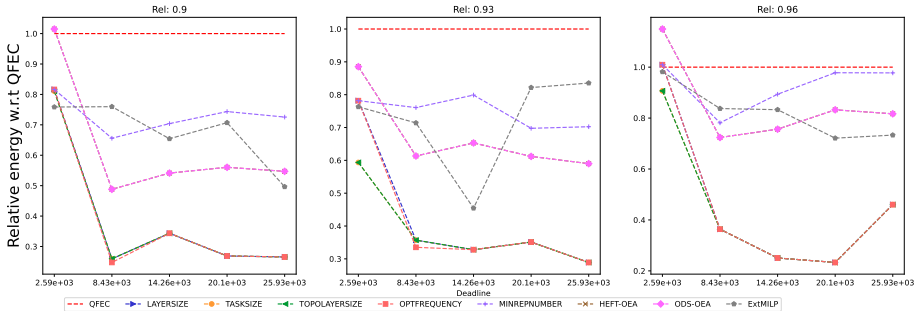

Figure 1680: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

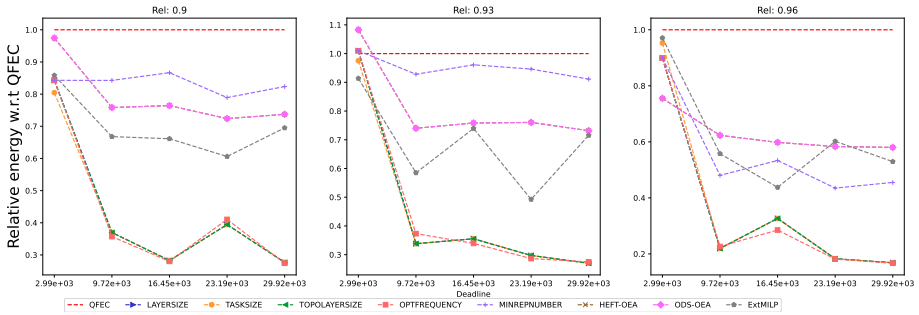

Figure 1681: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.3 $BC/WC = 0.3$

G.3.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

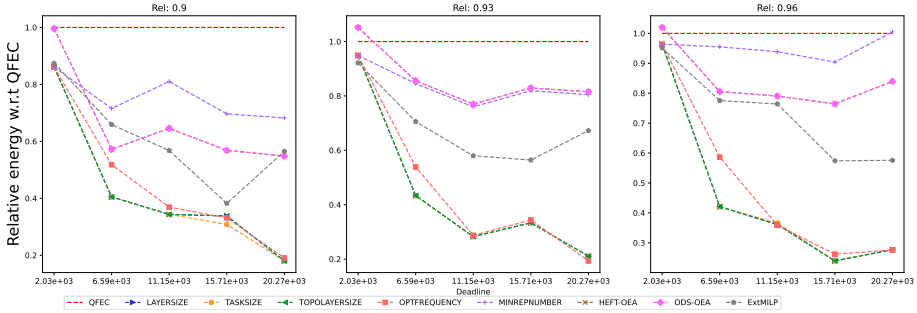

Figure 1682: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

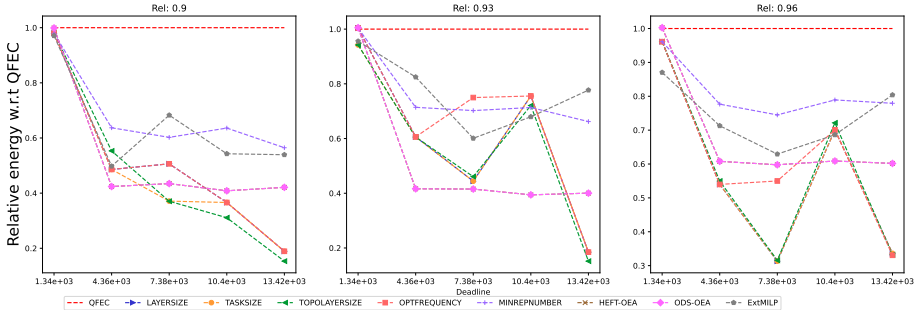

Figure 1683: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

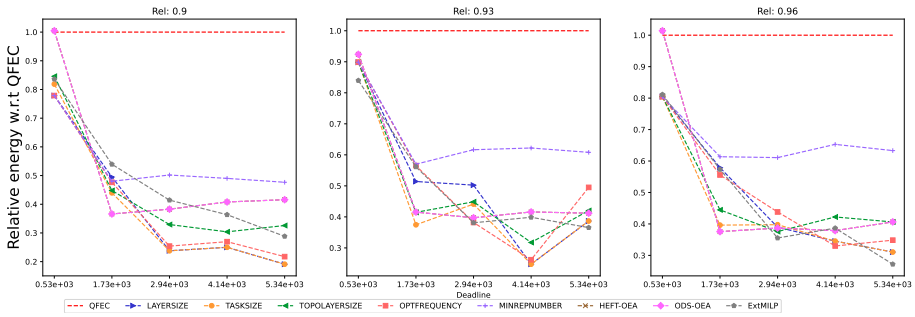

Figure 1684: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

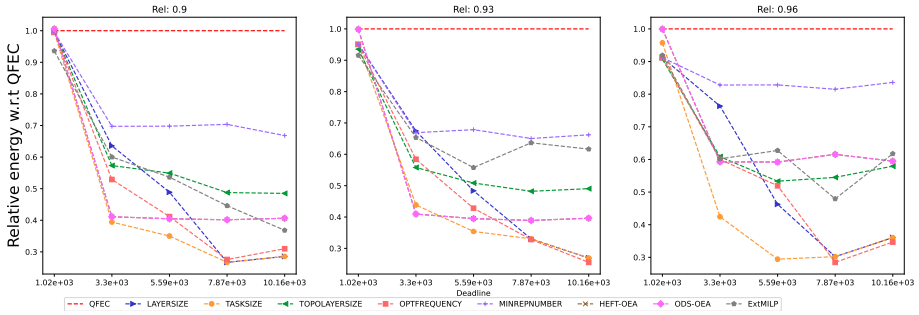

Figure 1685: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

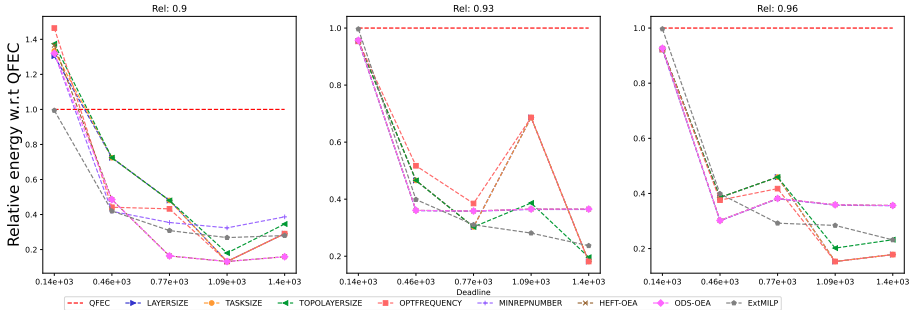

Figure 1686: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

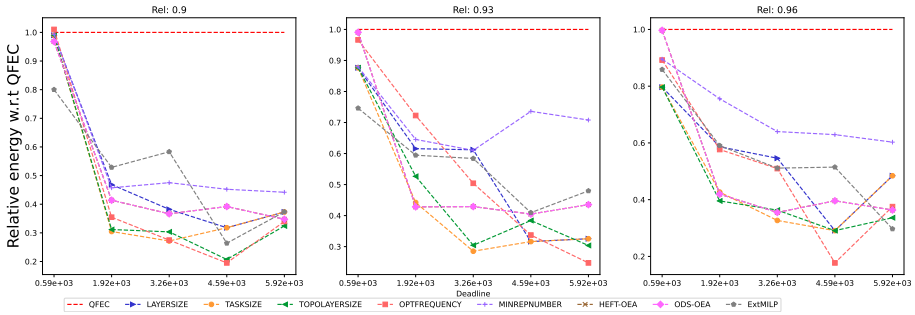

Figure 1687: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

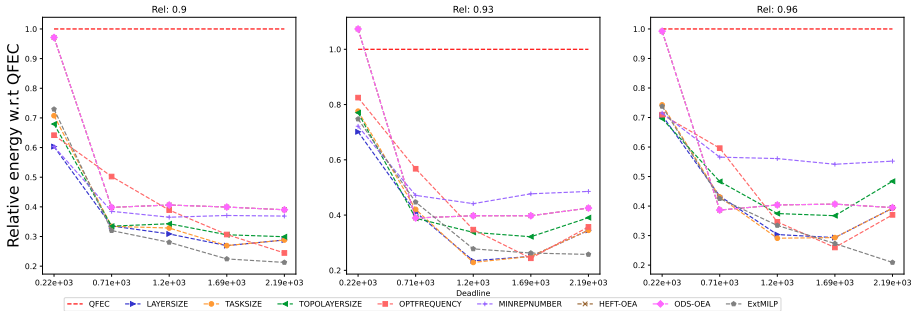

Figure 1688: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

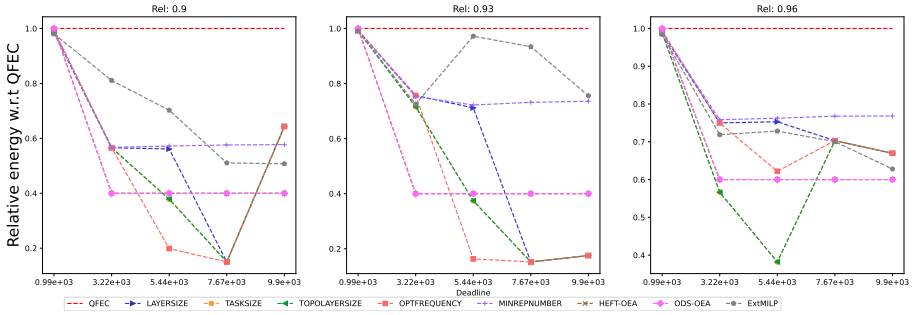

Figure 1689: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

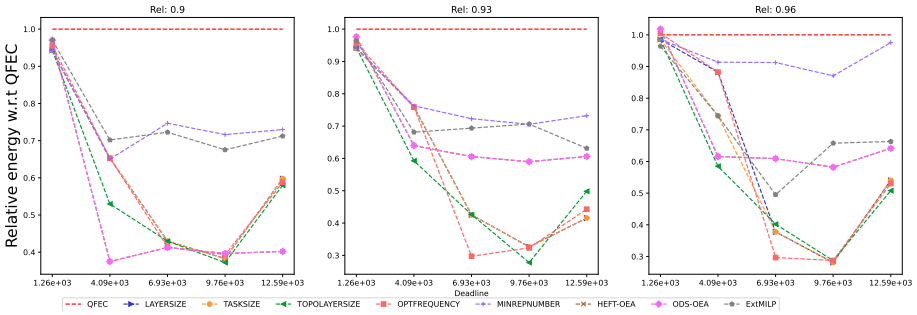

Figure 1690: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

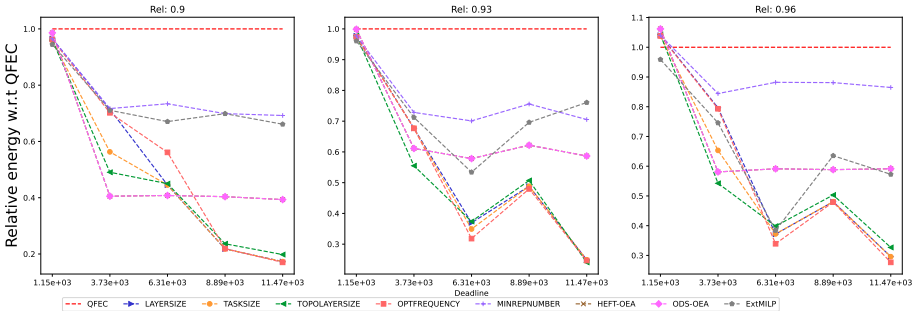

Figure 1691: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

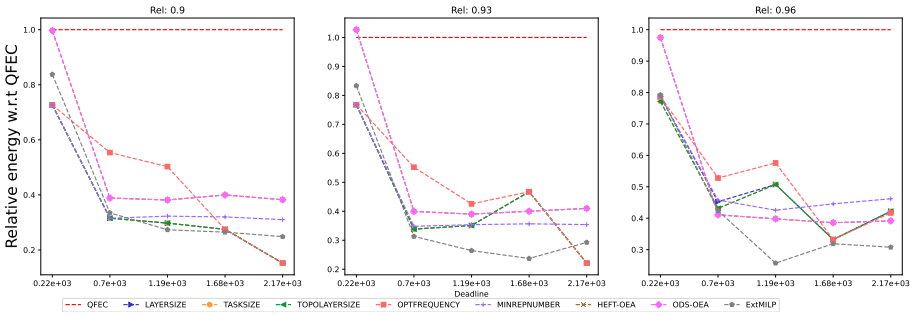

Figure 1692: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

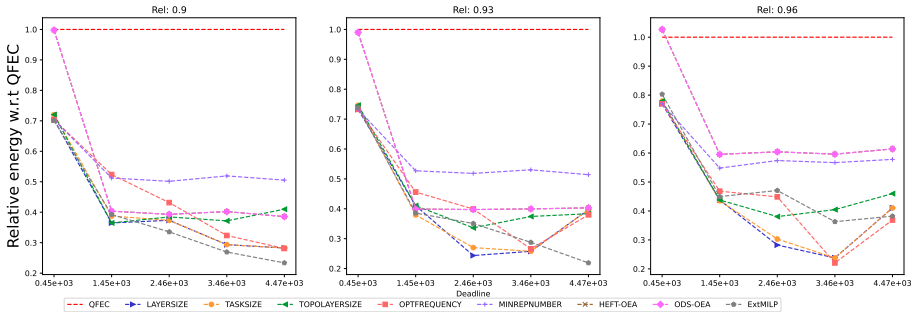

Figure 1693: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

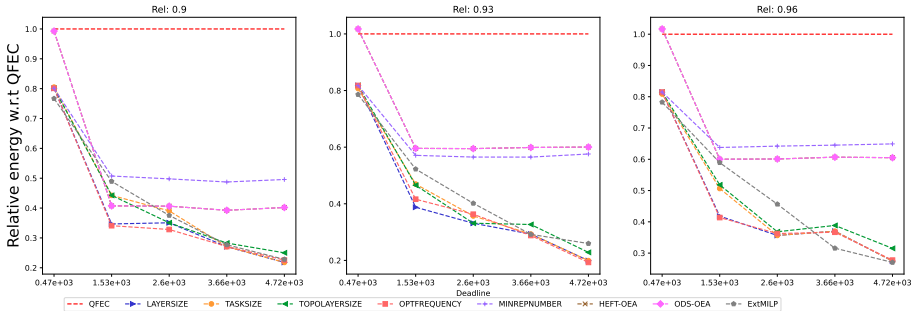

Figure 1694: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

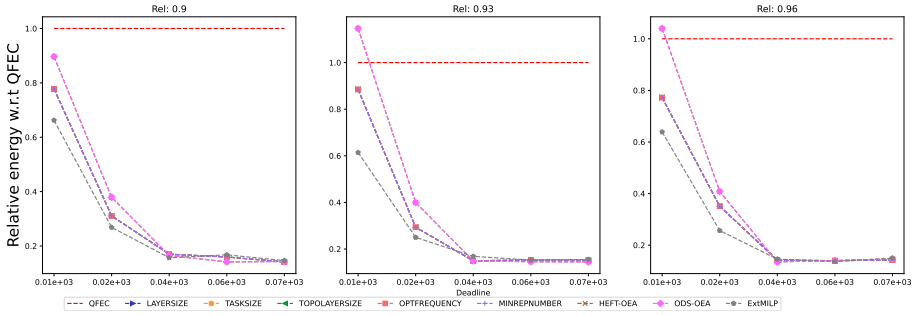

Figure 1695: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

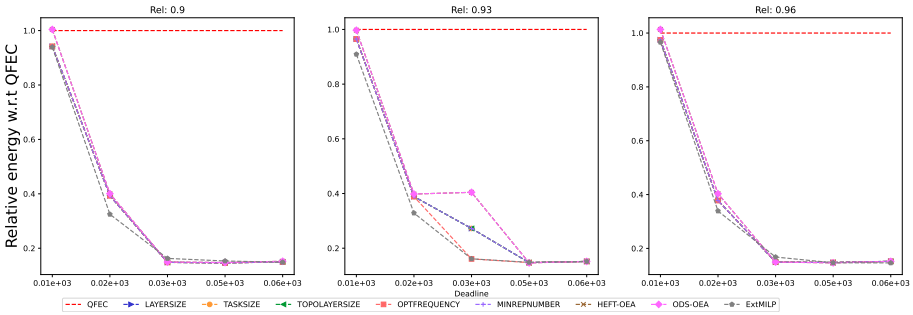

Figure 1696: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

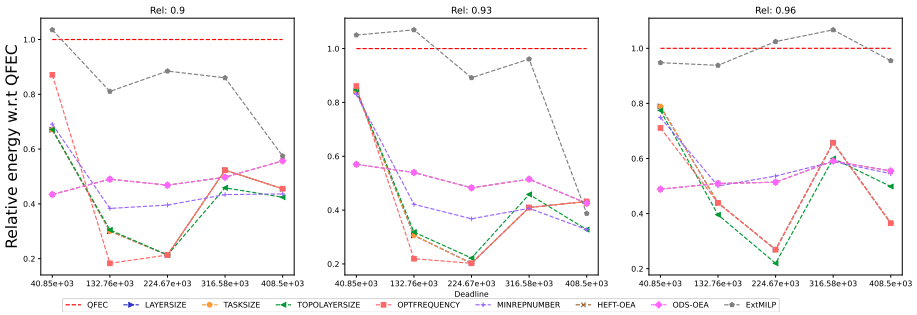

Figure 1697: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

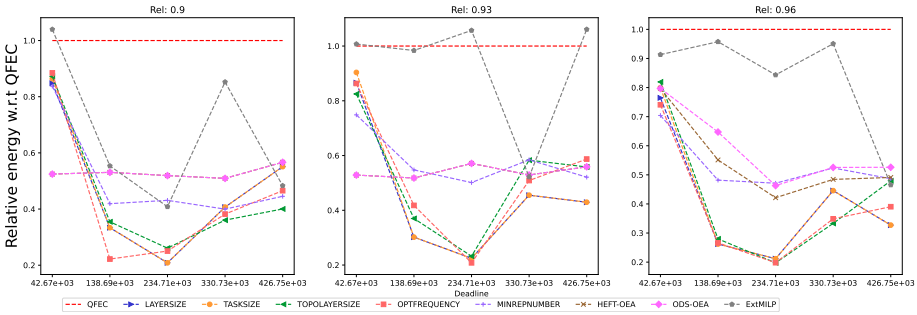

Figure 1698: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

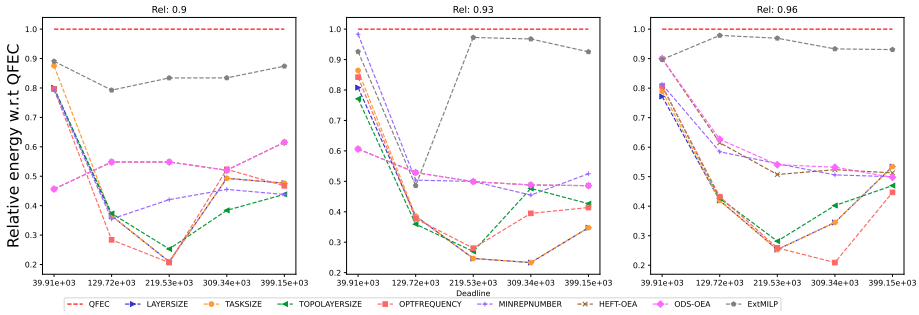

Figure 1699: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

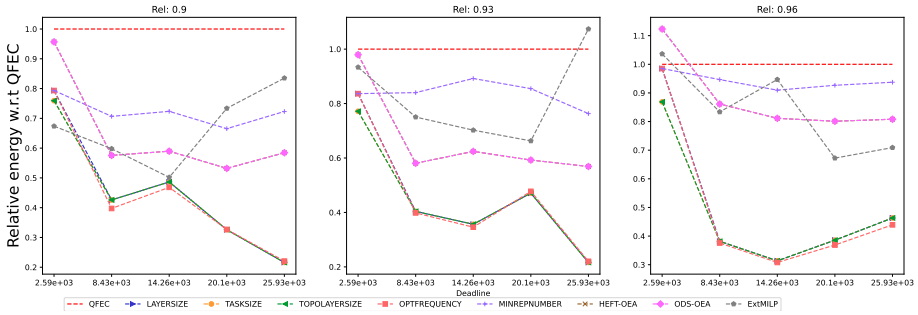

Figure 1700: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

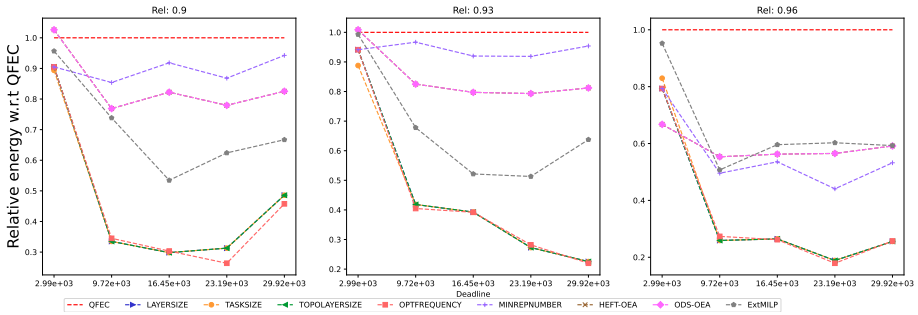

Figure 1701: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.3.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

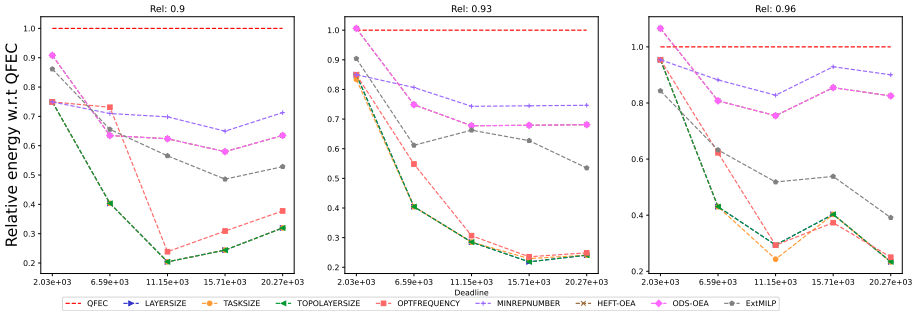

Figure 1702: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

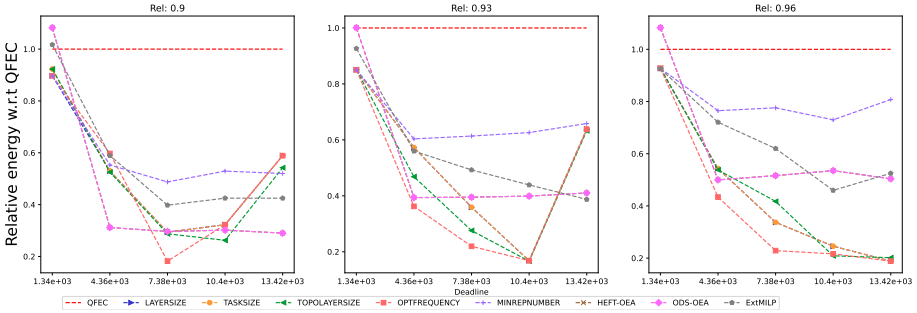

Figure 1703: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

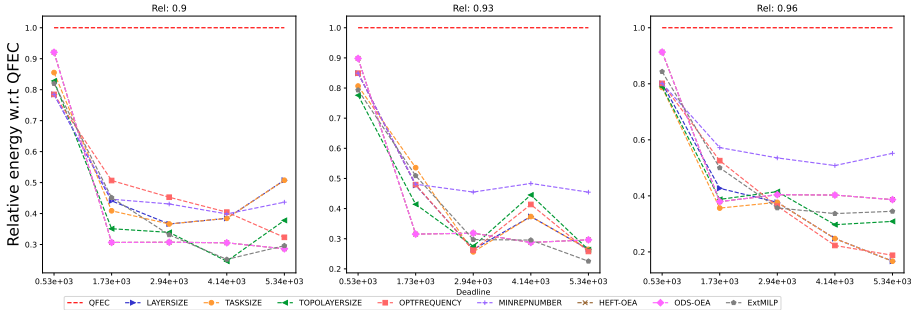

Figure 1704: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

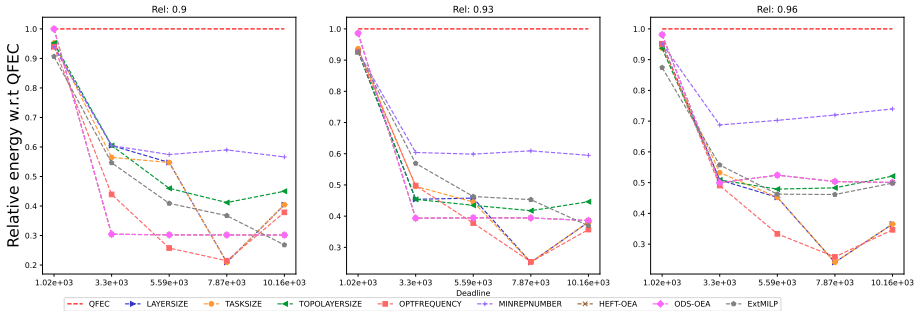

Figure 1705: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

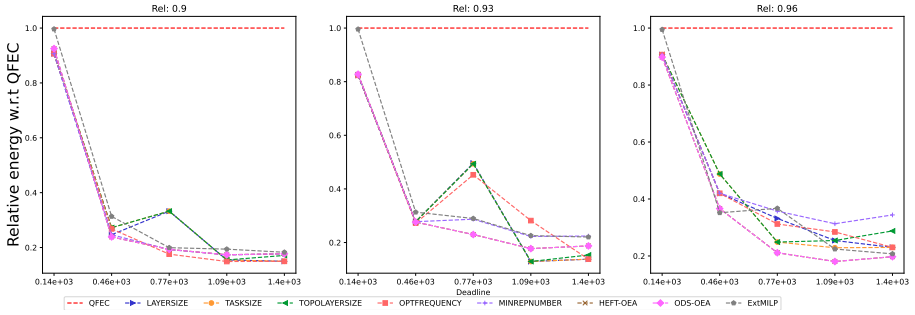

Figure 1706: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

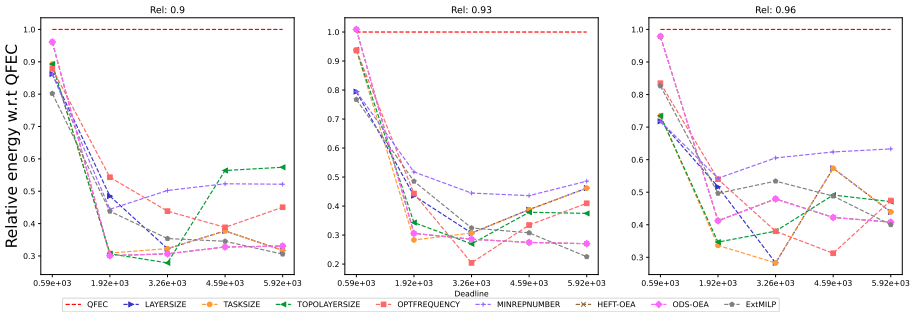

Figure 1707: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

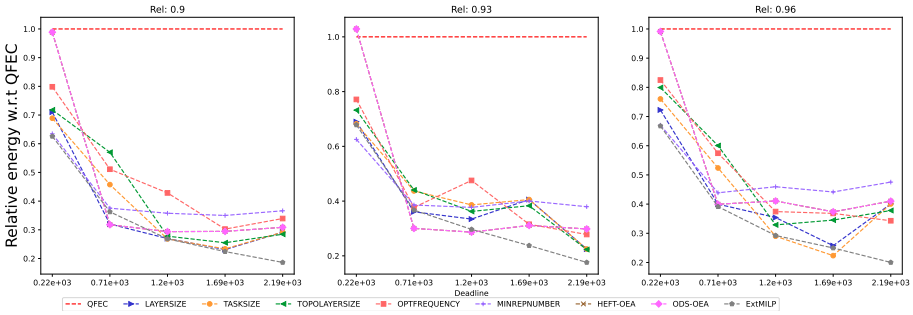

Figure 1708: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

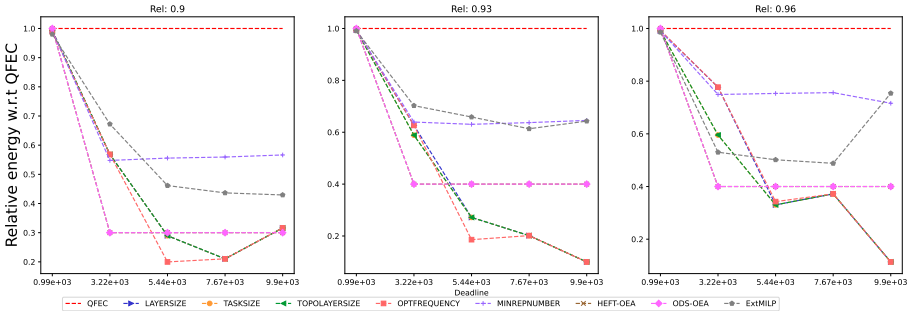

Figure 1709: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

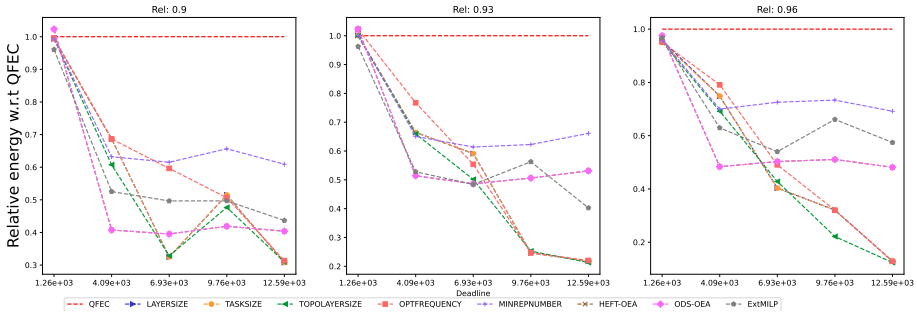

Figure 1710: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

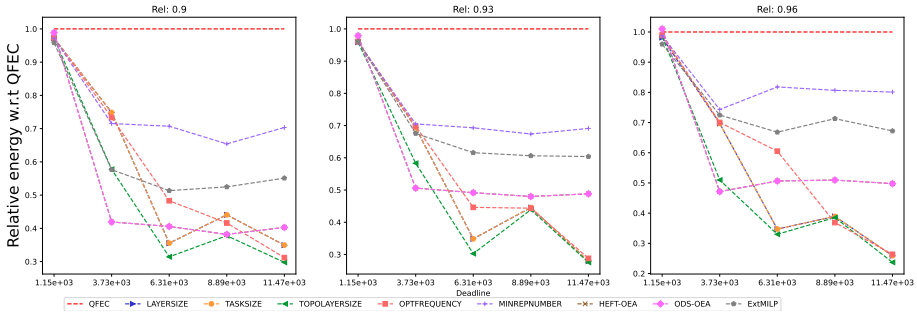

Figure 1711: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

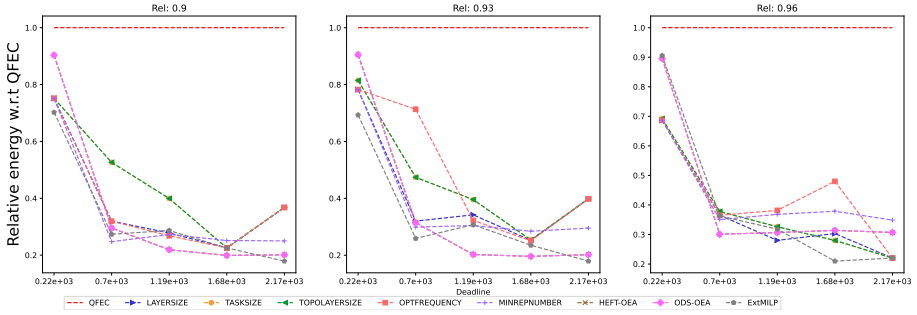

Figure 1712: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

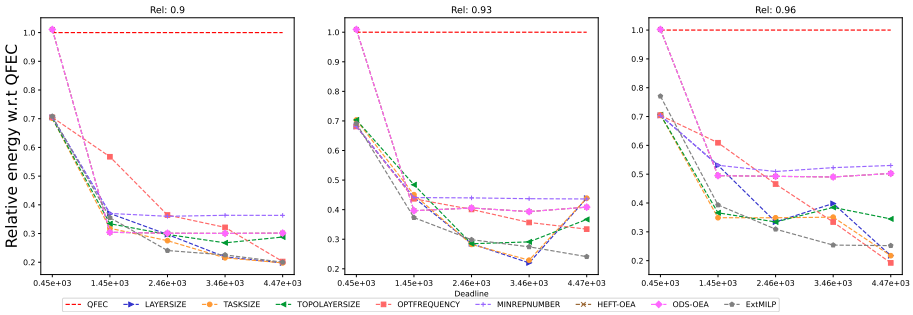

Figure 1713: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

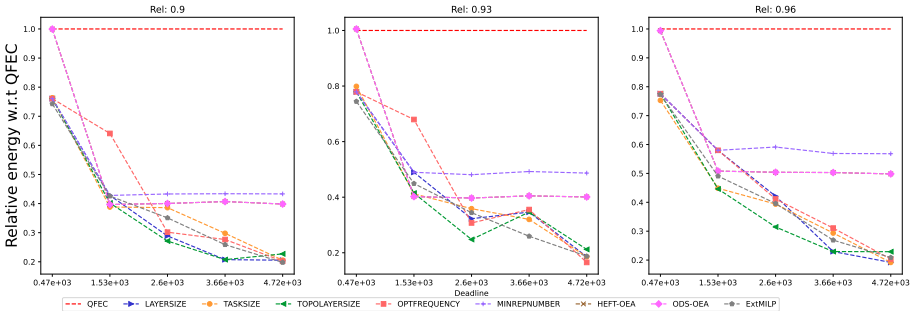

Figure 1714: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

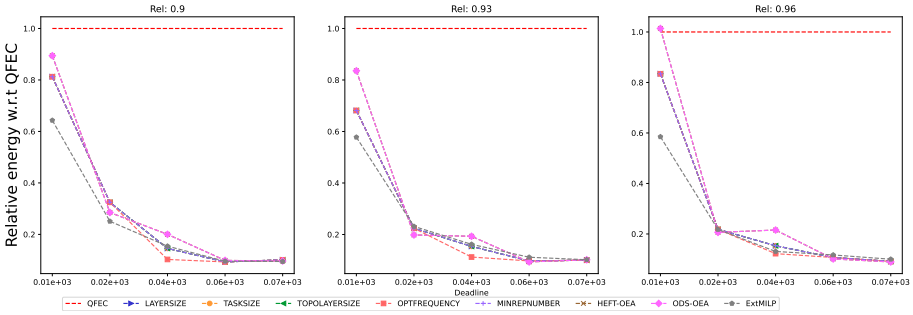

Figure 1715: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

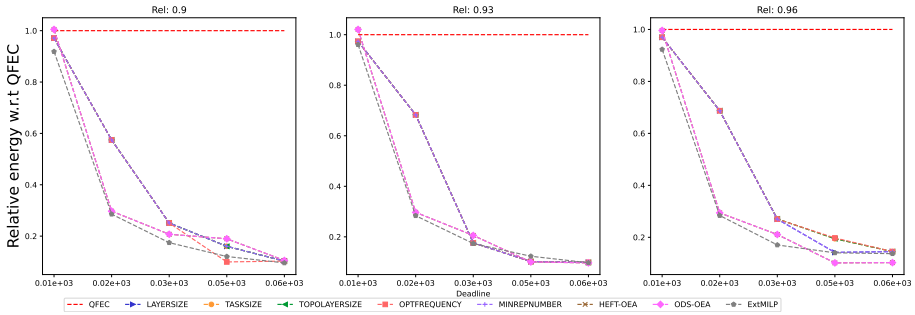

Figure 1716: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

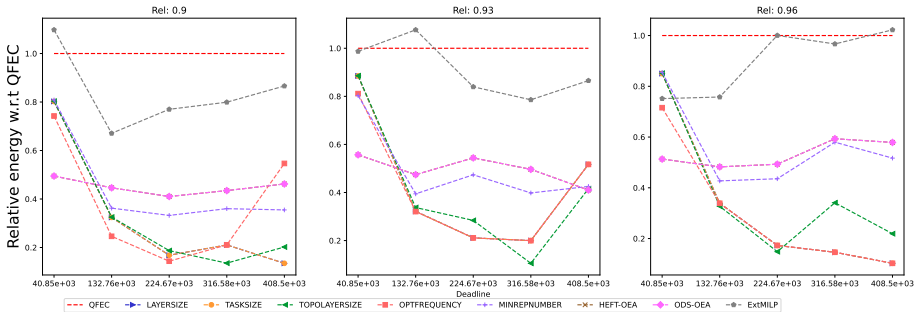

Figure 1717: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

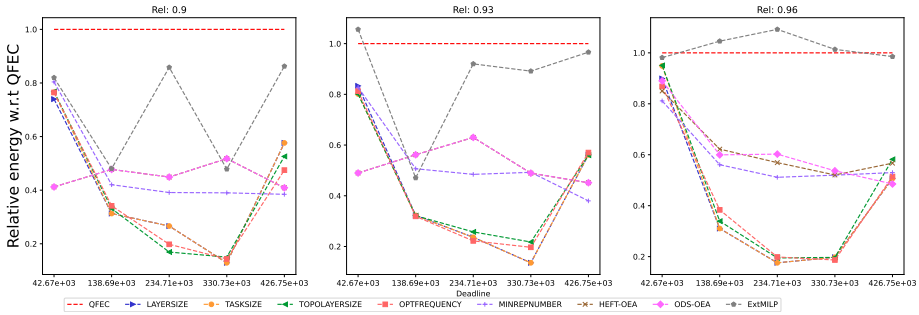

Figure 1718: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

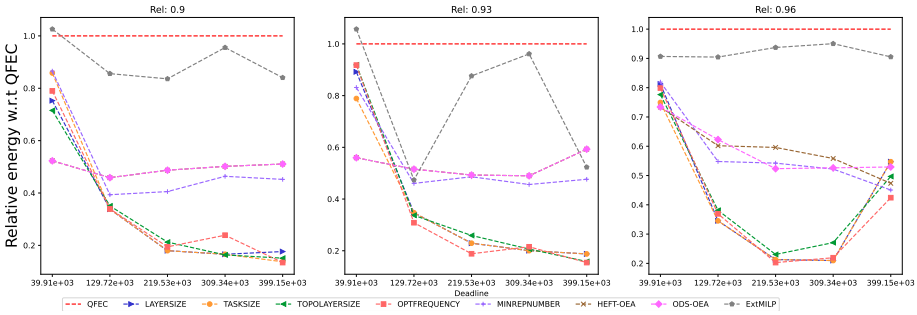

Figure 1719: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

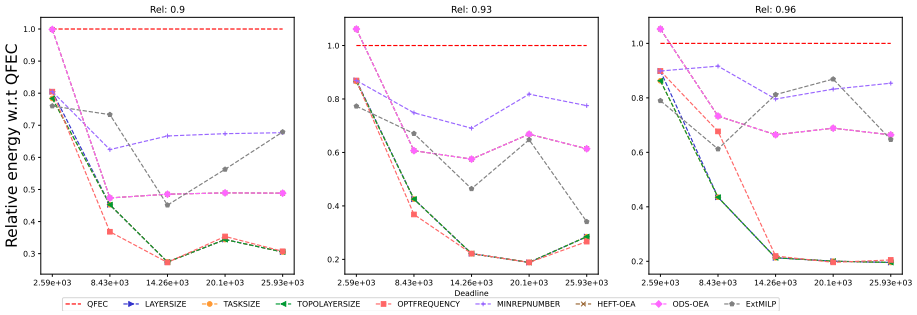

Figure 1720: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

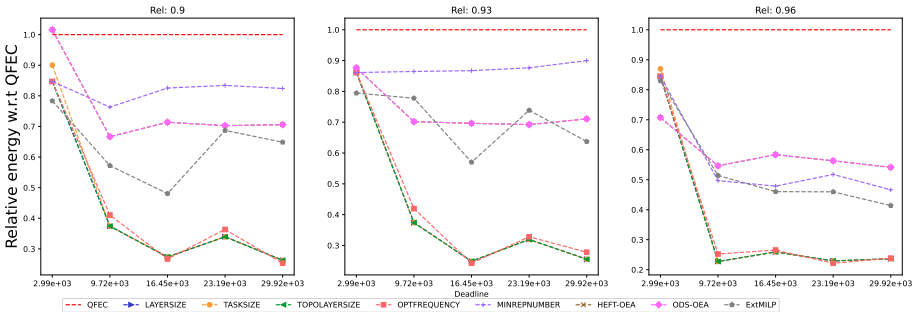

Figure 1721: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.3.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

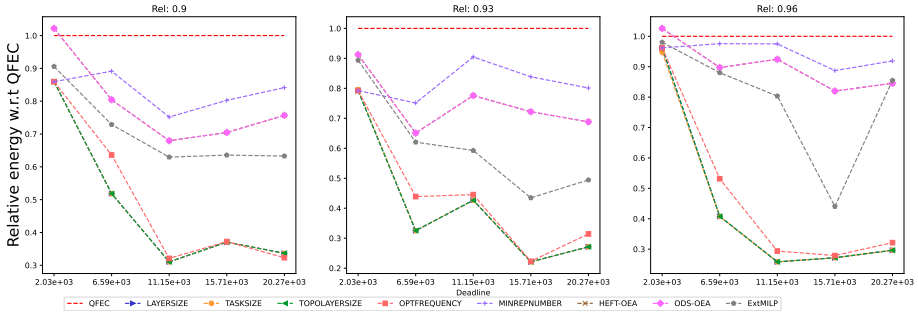

Figure 1722: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

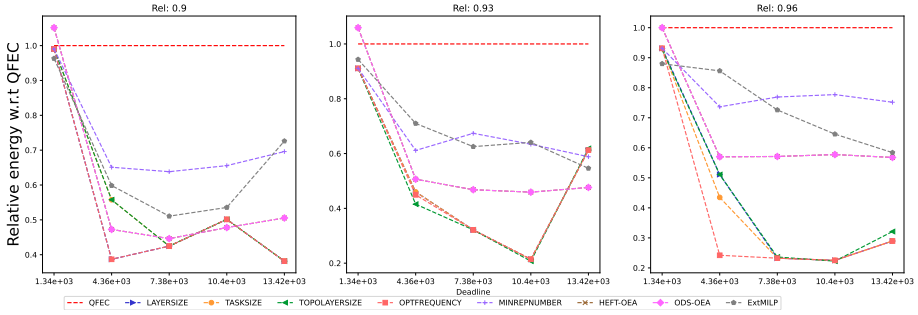

Figure 1723: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

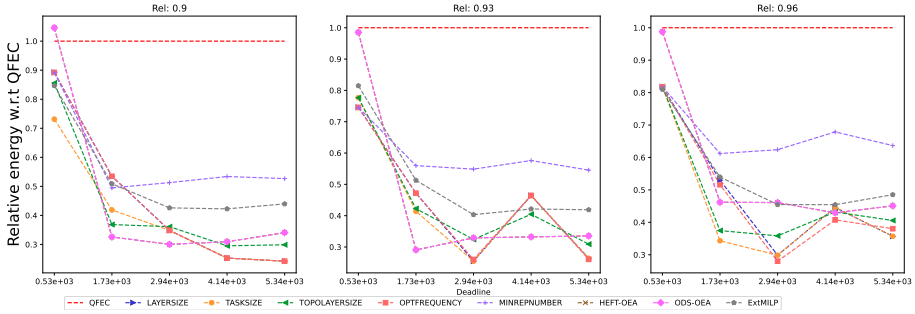

Figure 1724: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

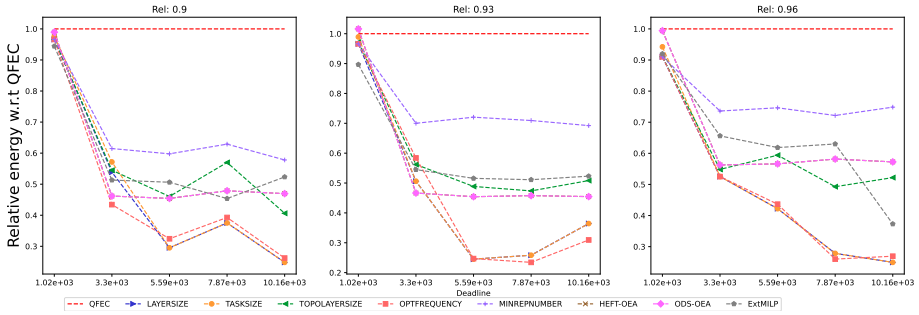

Figure 1725: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

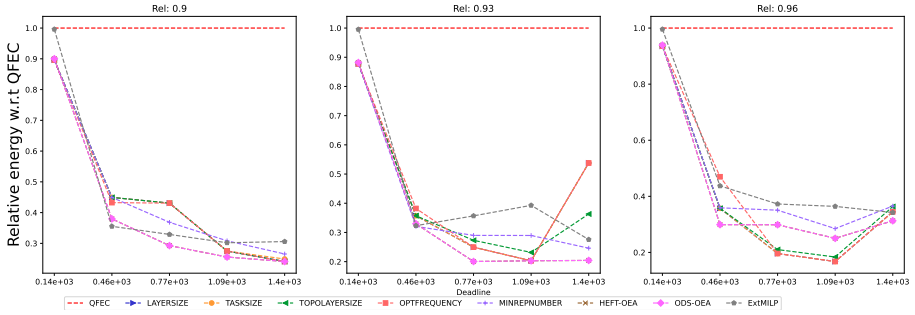

Figure 1726: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

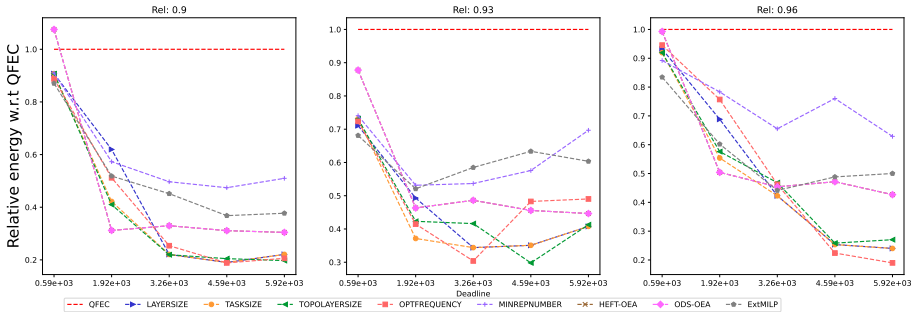

Figure 1727: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

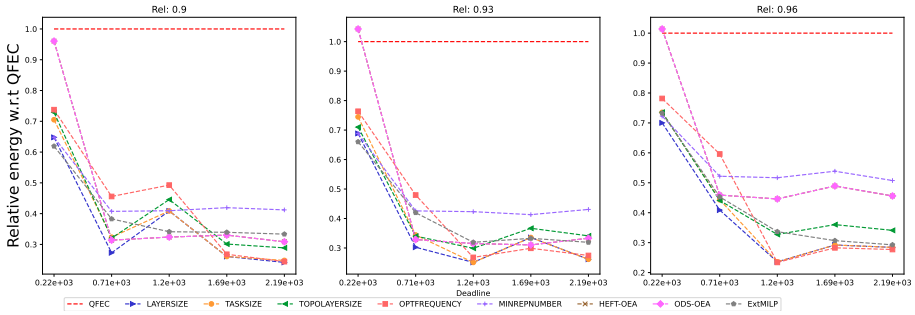

Figure 1728: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

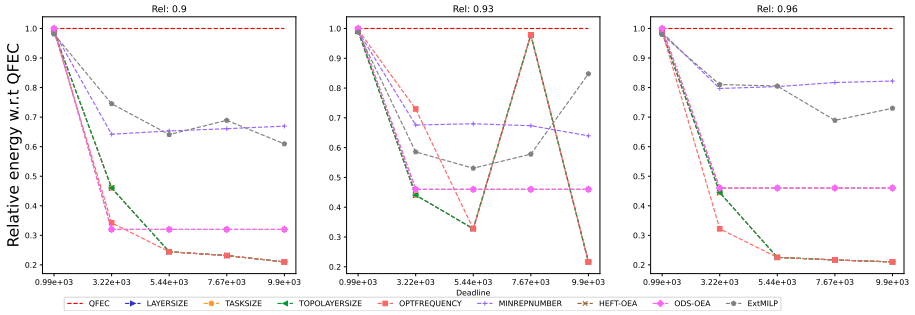

Figure 1729: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

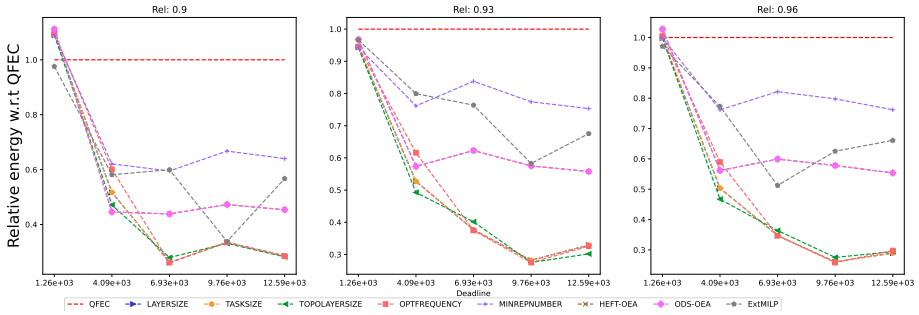

Figure 1730: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

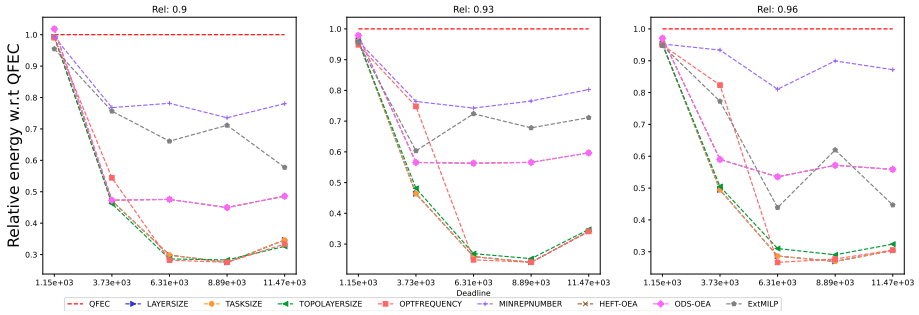

Figure 1731: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

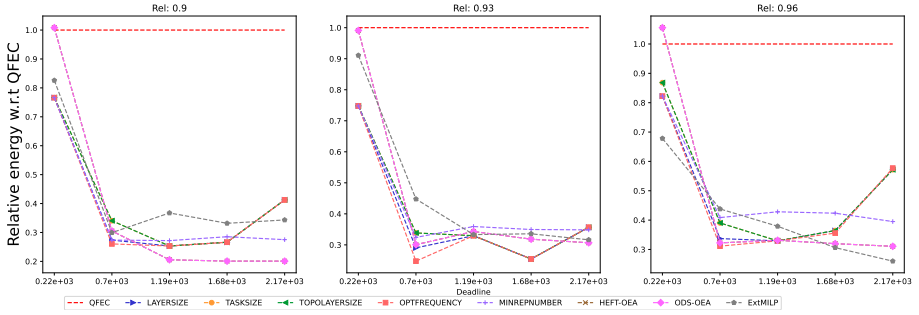

Figure 1732: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

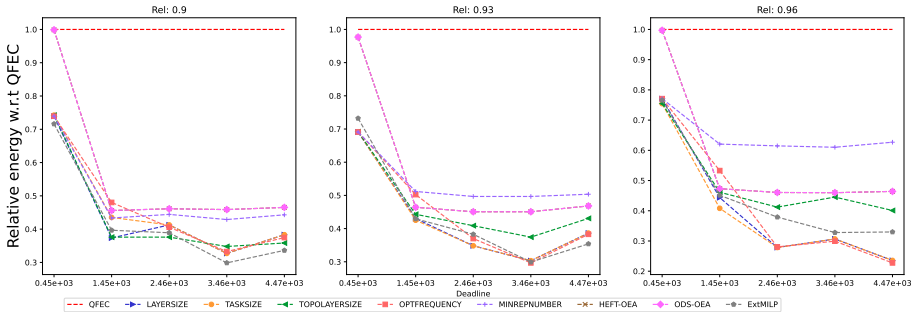

Figure 1733: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

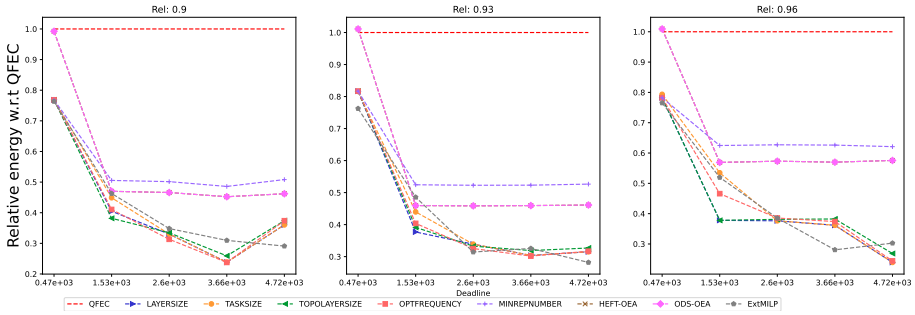

Figure 1734: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

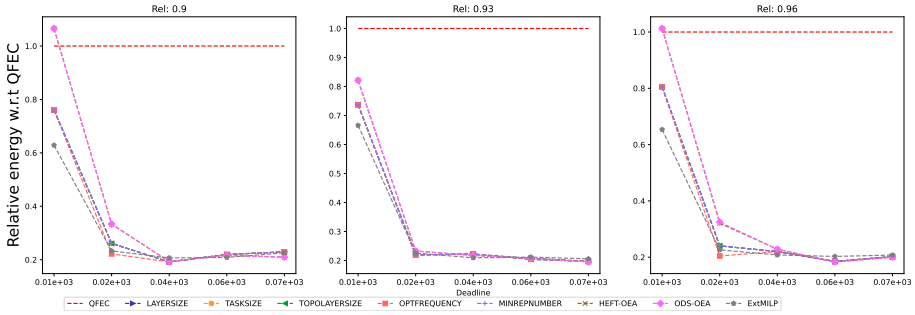

Figure 1735: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

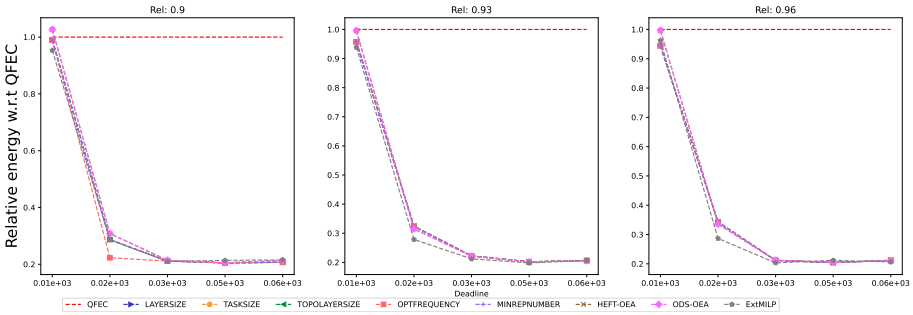

Figure 1736: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

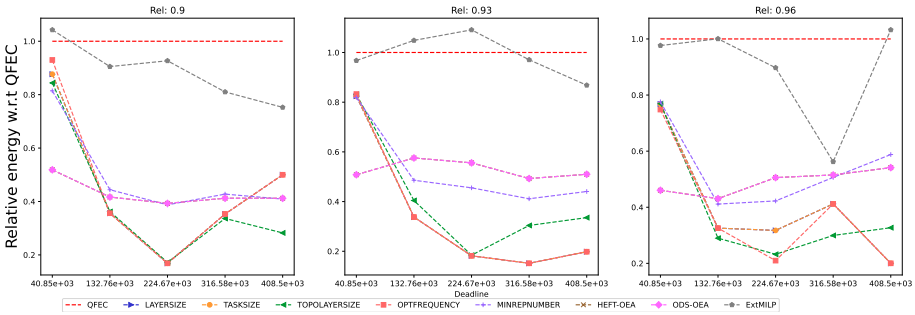

Figure 1737: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

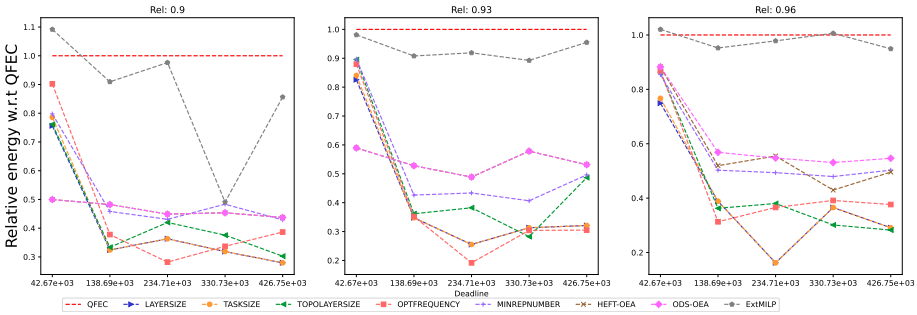

Figure 1738: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

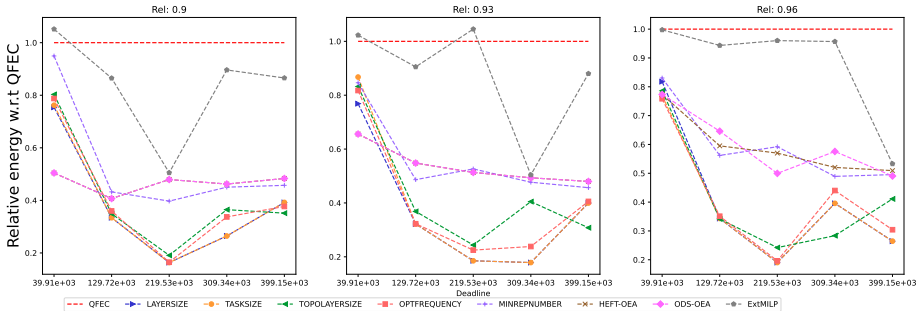

Figure 1739: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

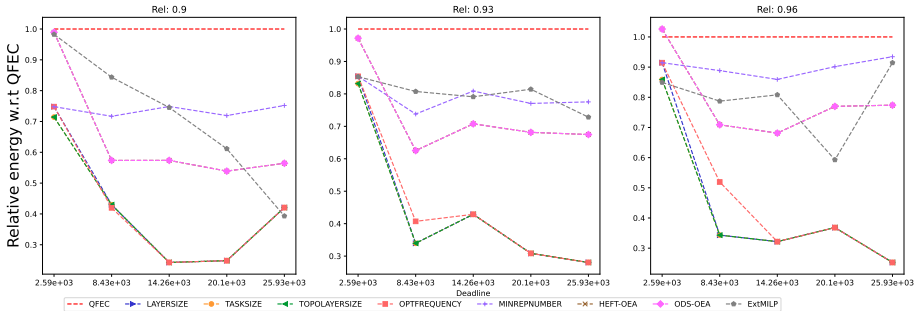

Figure 1740: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

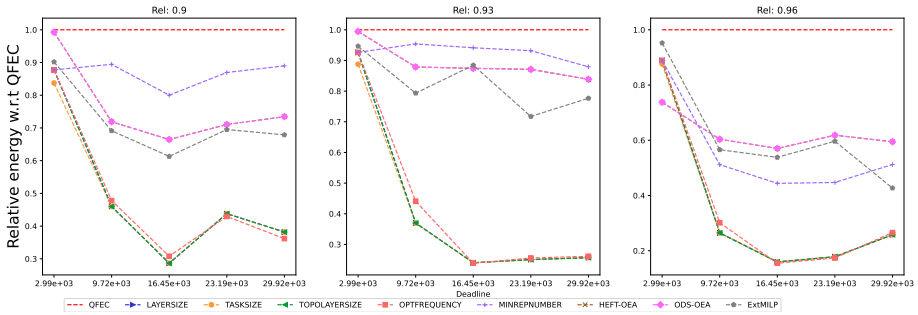

Figure 1741: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.3.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

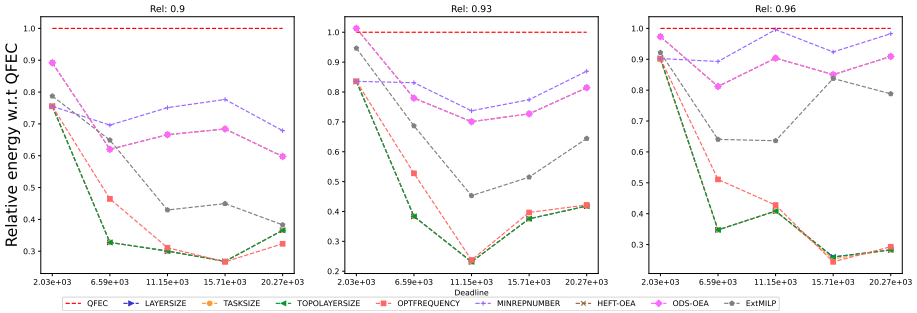

Figure 1742: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

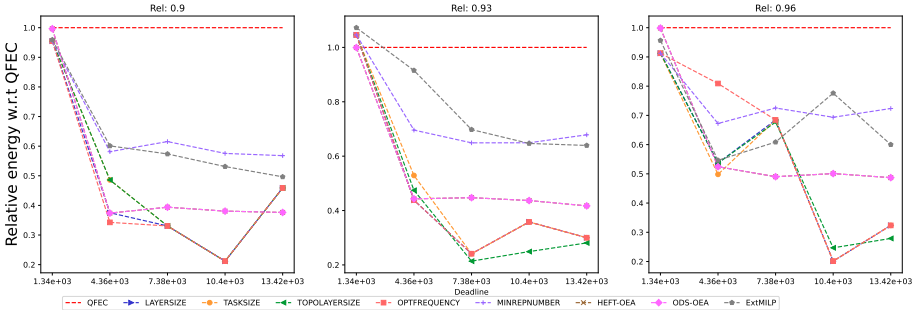

Figure 1743: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

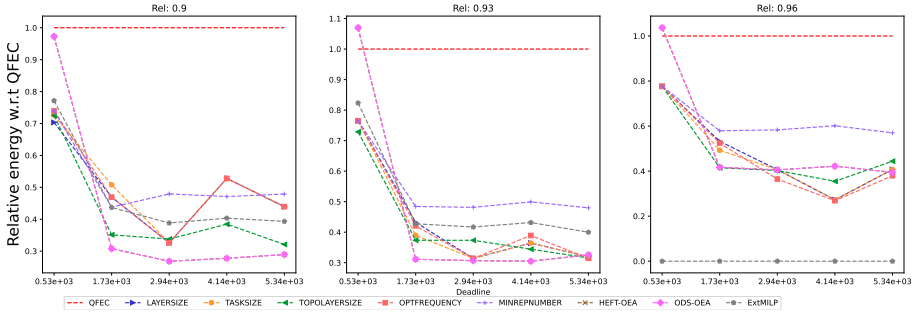

Figure 1744: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

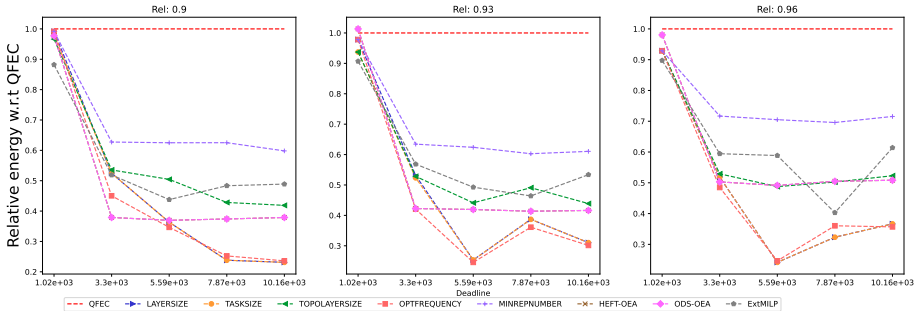

Figure 1745: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

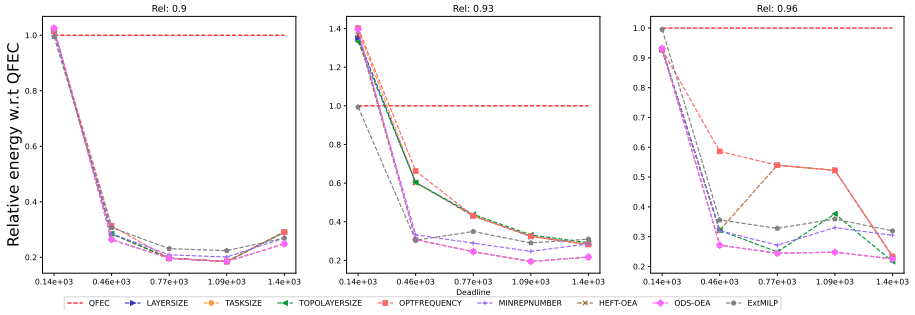

Figure 1746: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

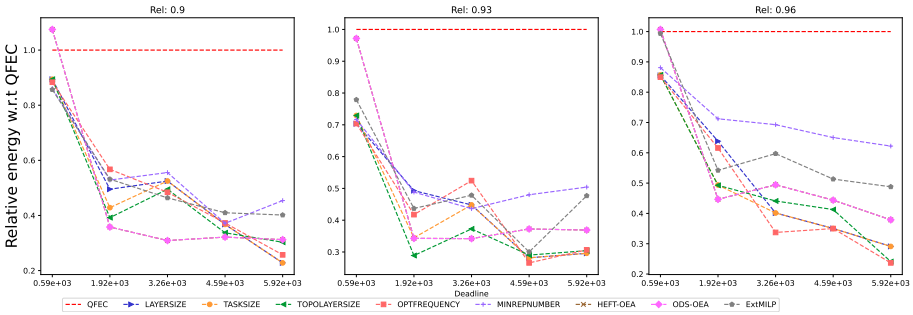

Figure 1747: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

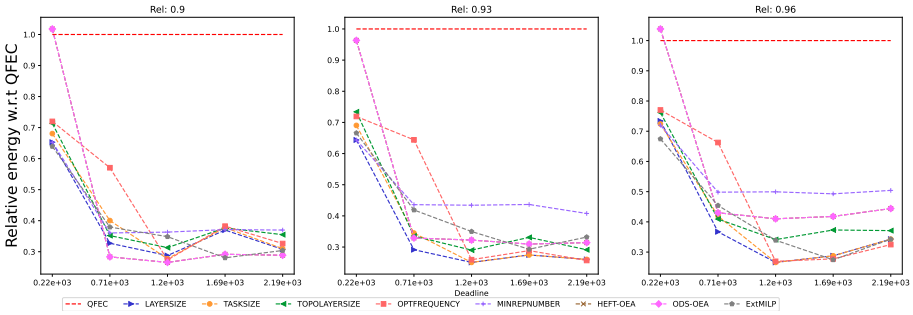

Figure 1748: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

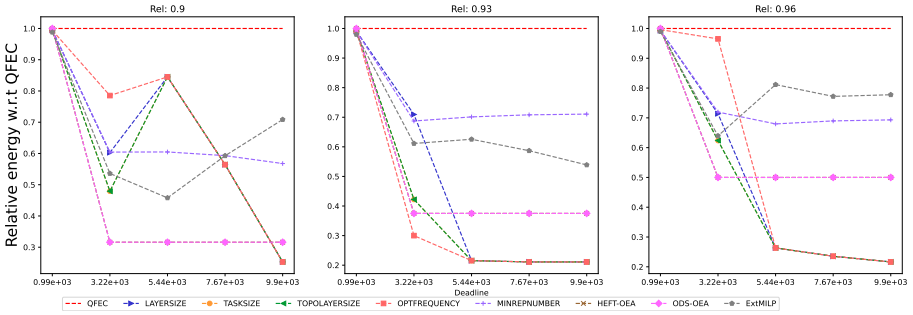

Figure 1749: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

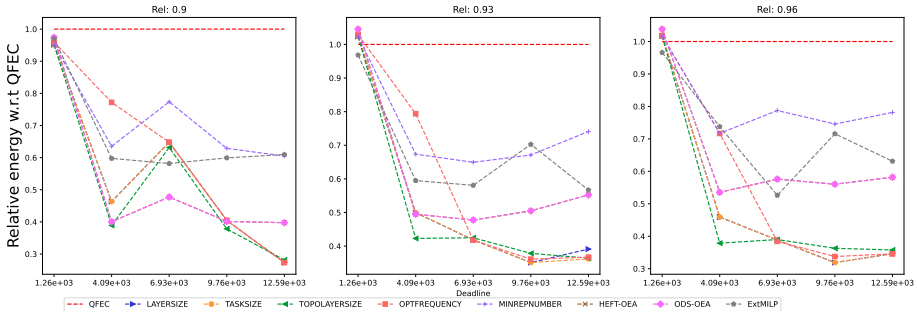

Figure 1750: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

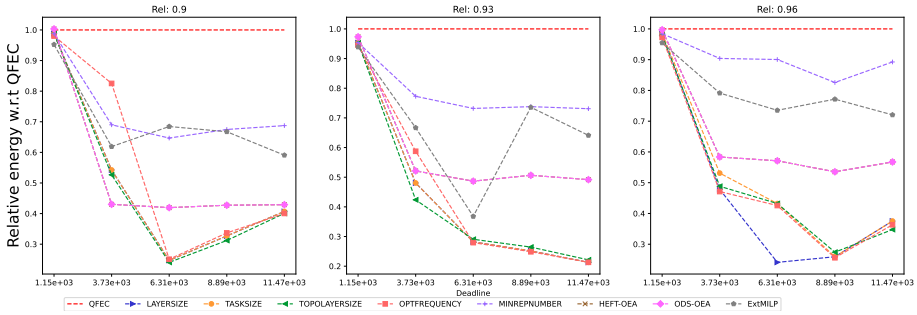

Figure 1751: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

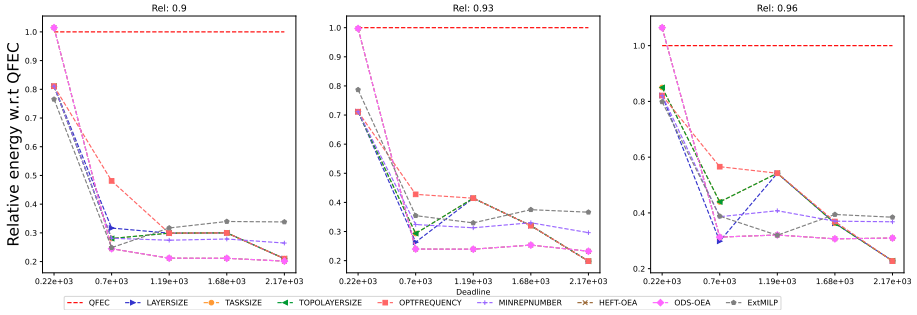

Figure 1752: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

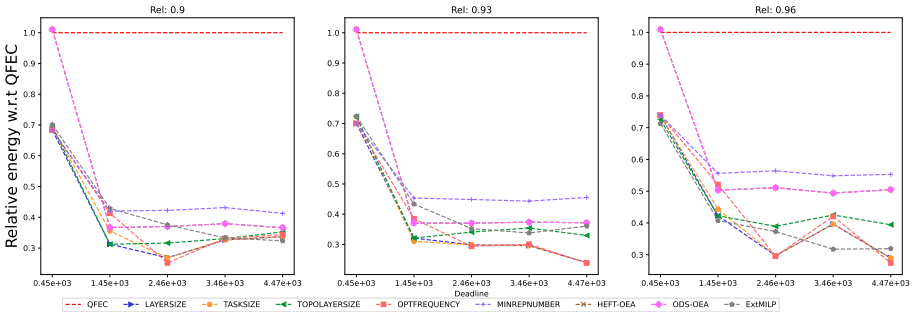

Figure 1753: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

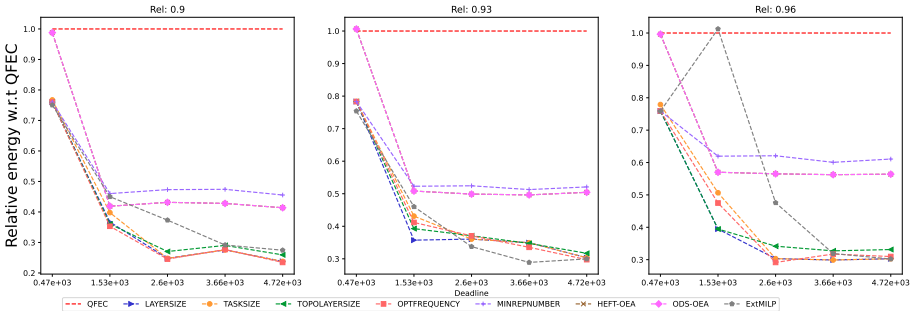

Figure 1754: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

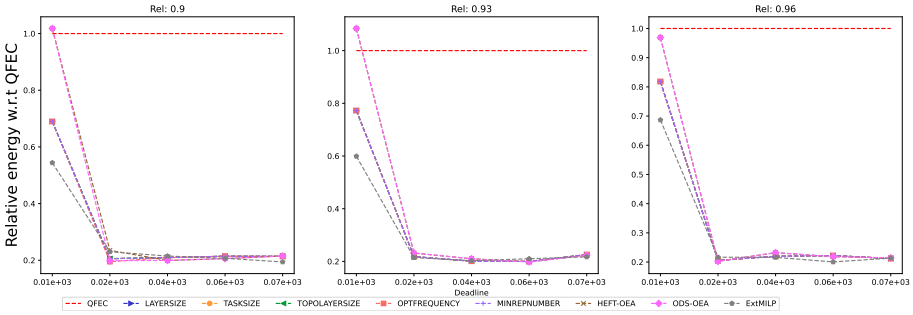

Figure 1755: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

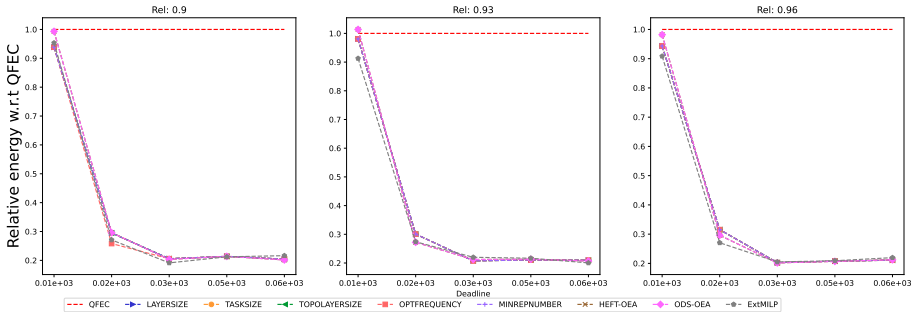

Figure 1756: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

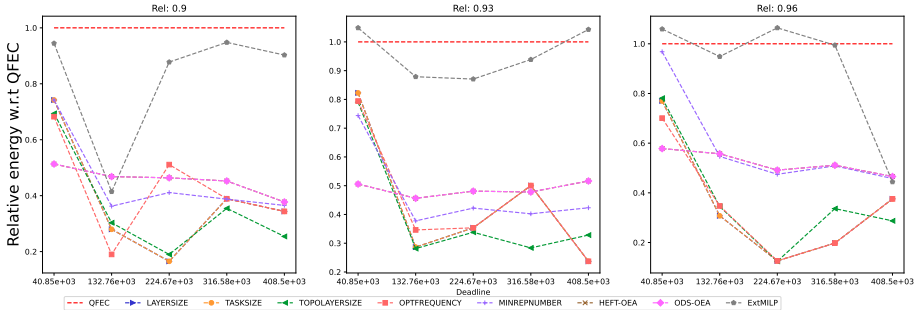

Figure 1757: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

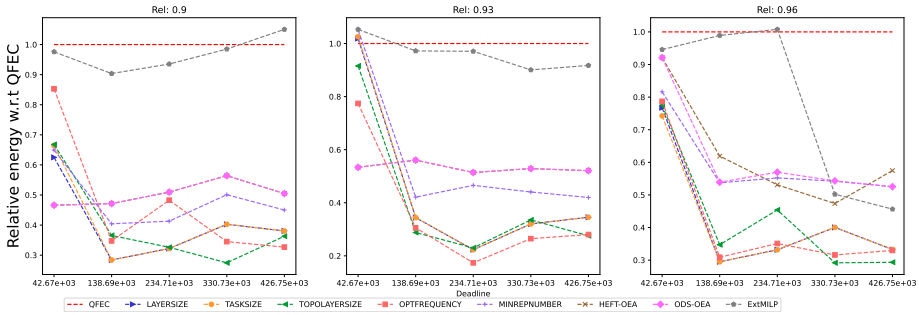

Figure 1758: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

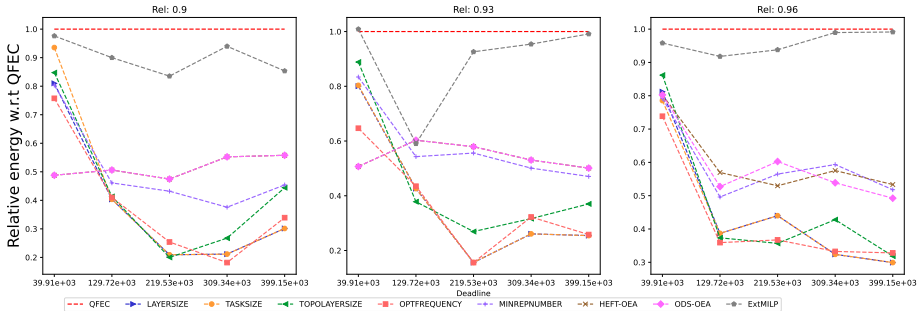

Figure 1759: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

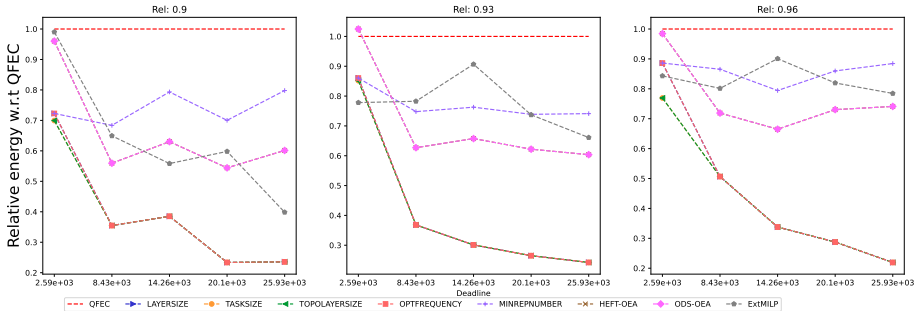

Figure 1760: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

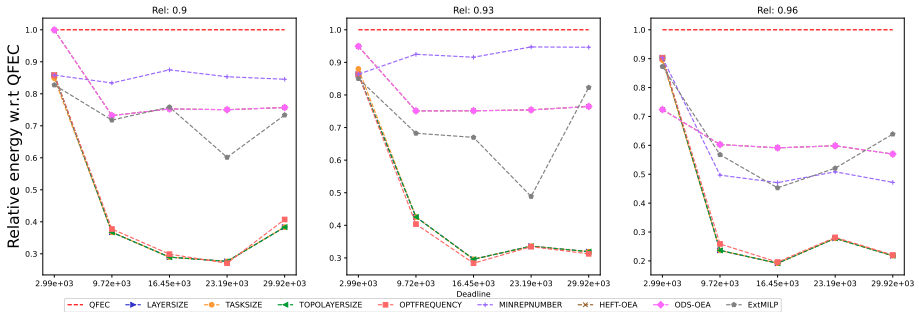

Figure 1761: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.4 $BC/WC = 0.4$

G.4.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

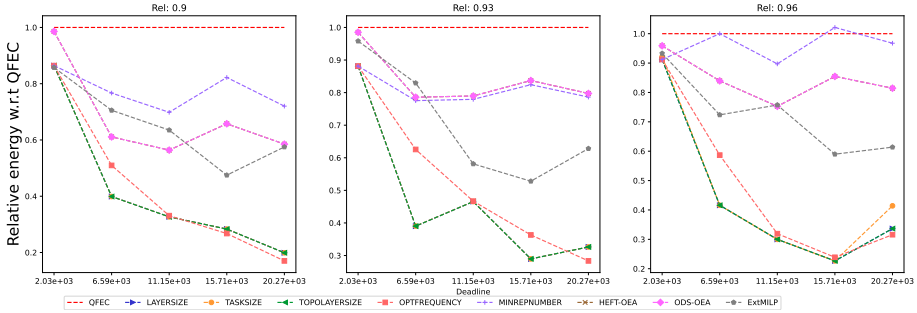

Figure 1762: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

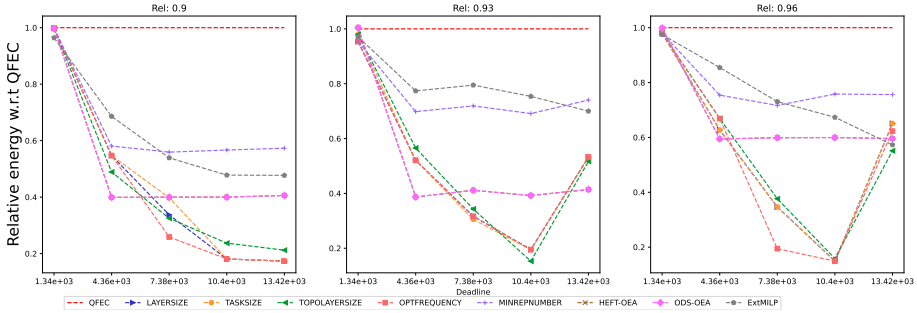

Figure 1763: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

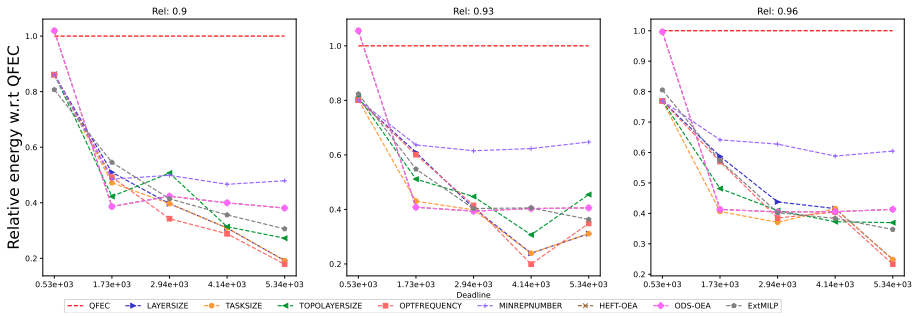

Figure 1764: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

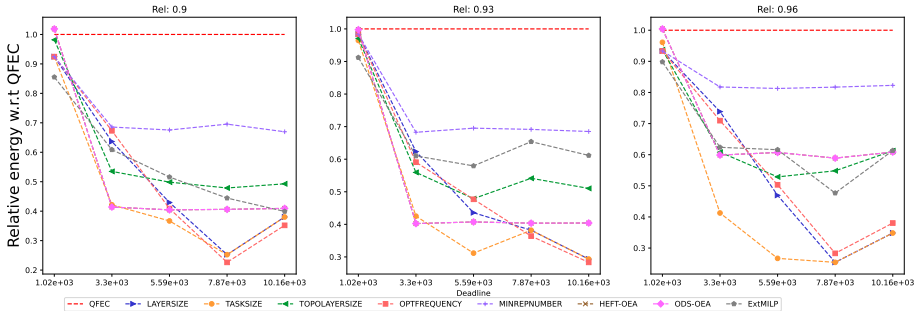

Figure 1765: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

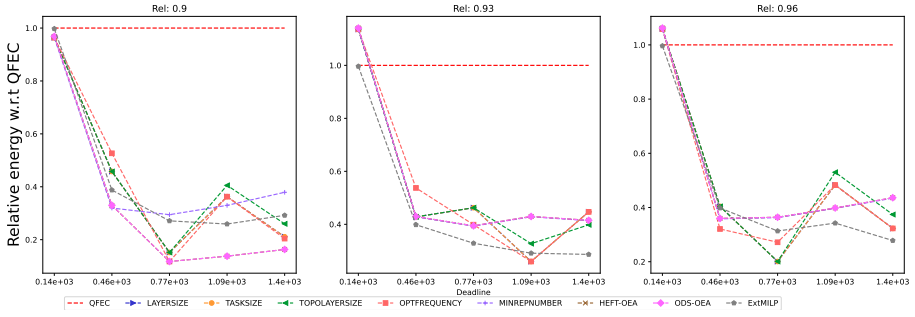

Figure 1766: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

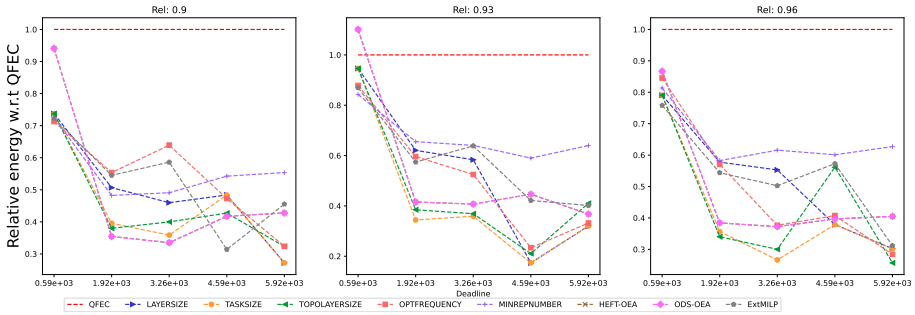

Figure 1767: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

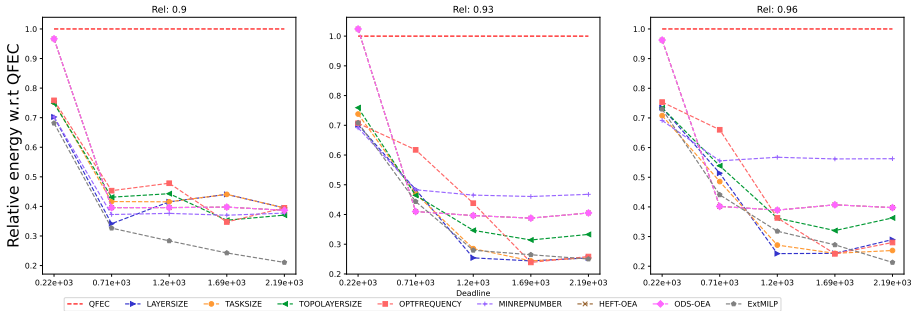

Figure 1768: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

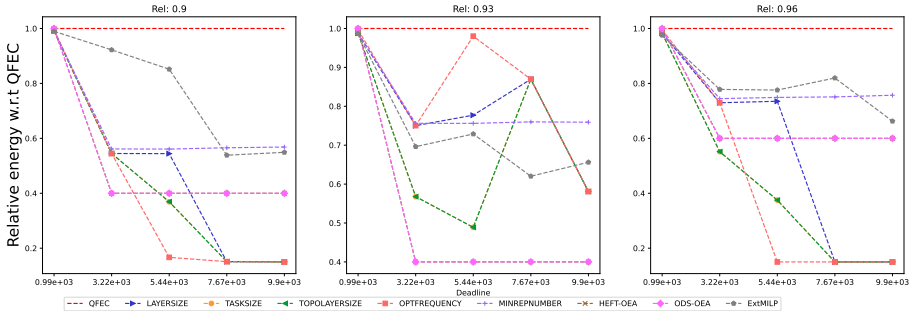

Figure 1769: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

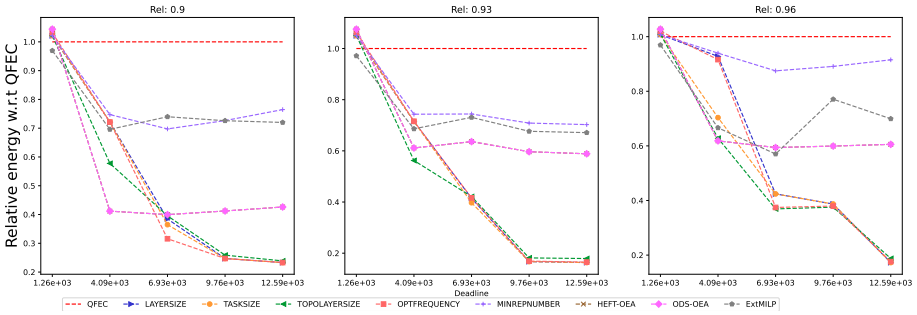

Figure 1770: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

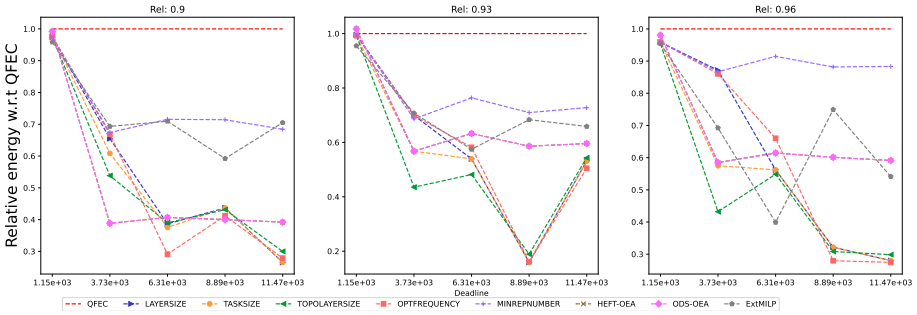

Figure 1771: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

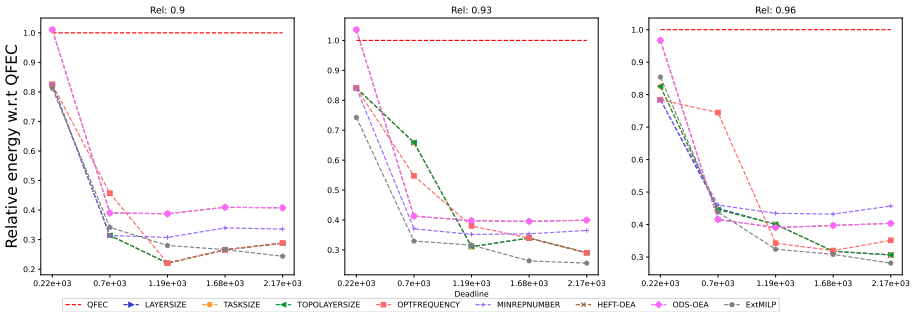

Figure 1772: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

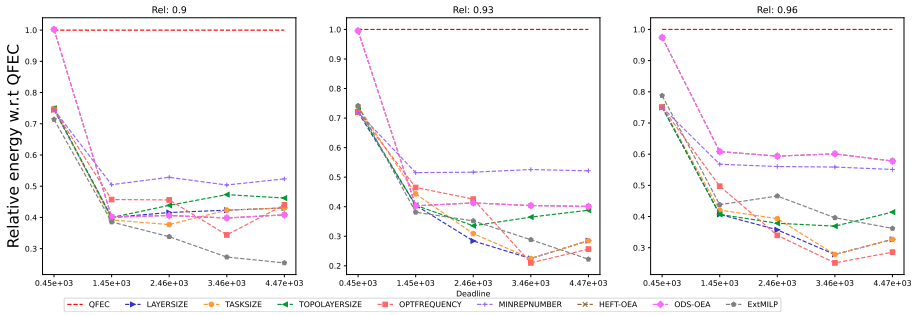

Figure 1773: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

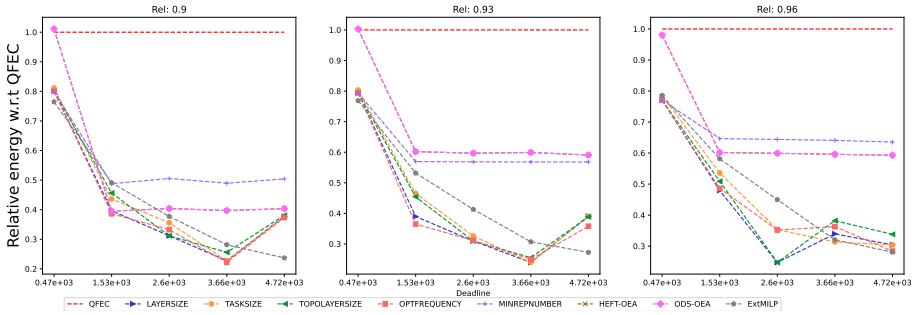

Figure 1774: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

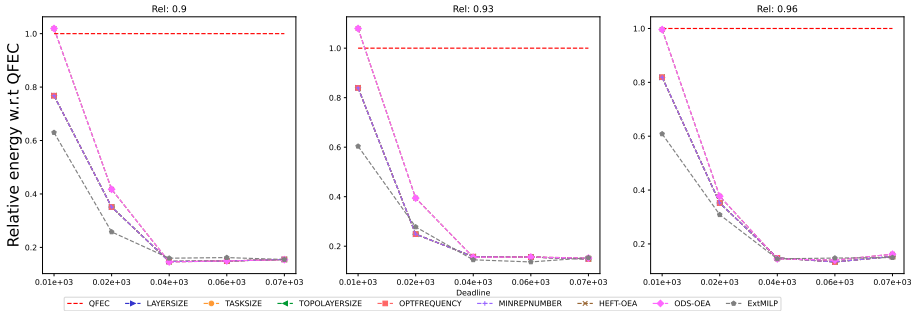

Figure 1775: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

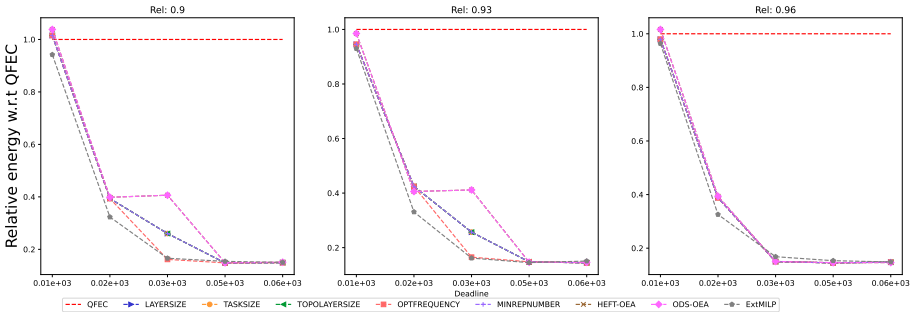

Figure 1776: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

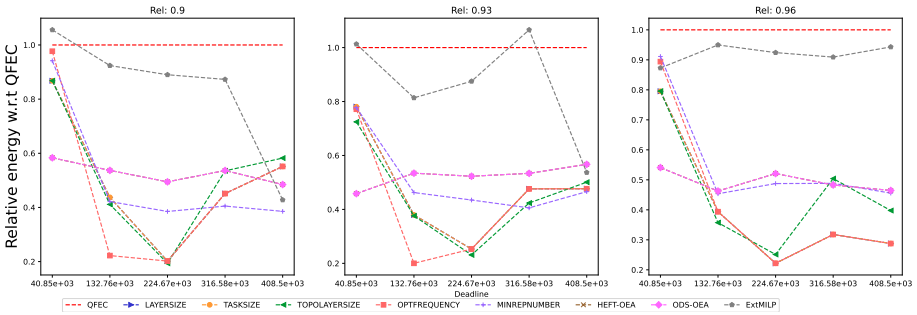

Figure 1777: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

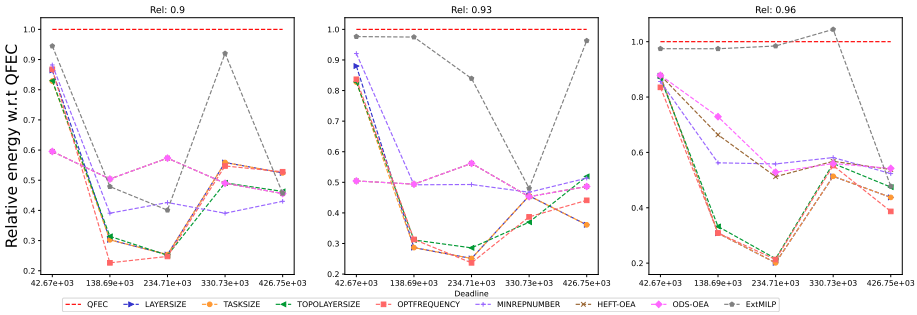

Figure 1778: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

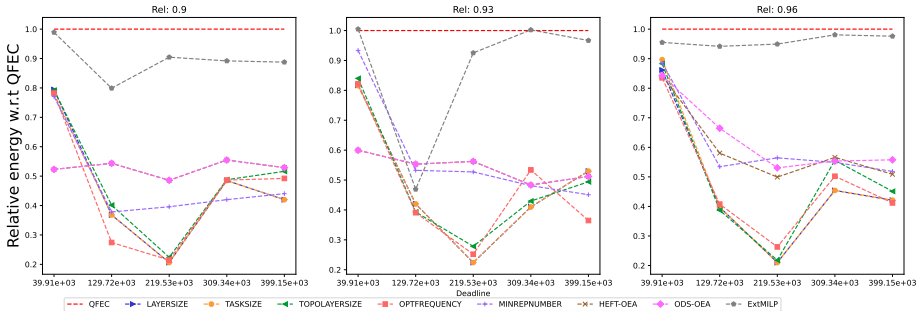

Figure 1779: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

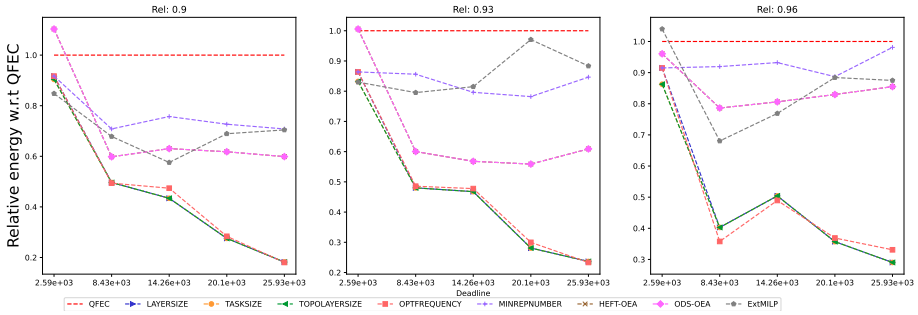

Figure 1780: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

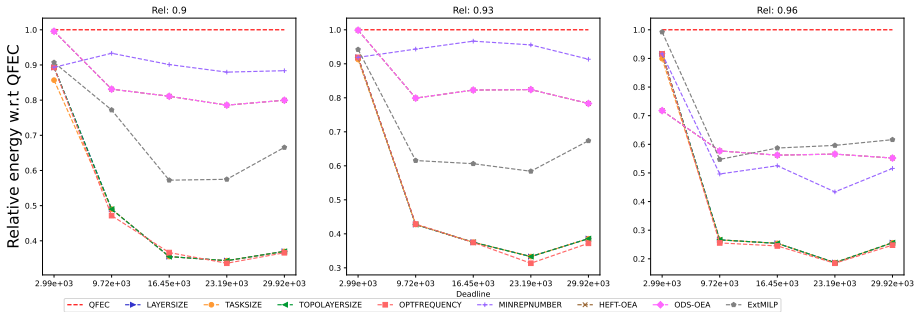

Figure 1781: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.4.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

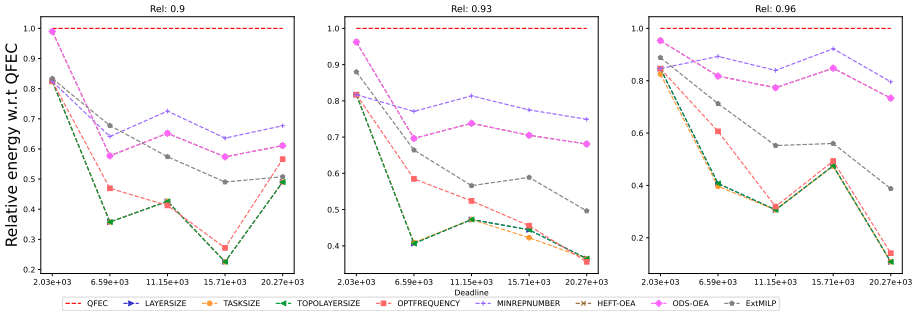

Figure 1782: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

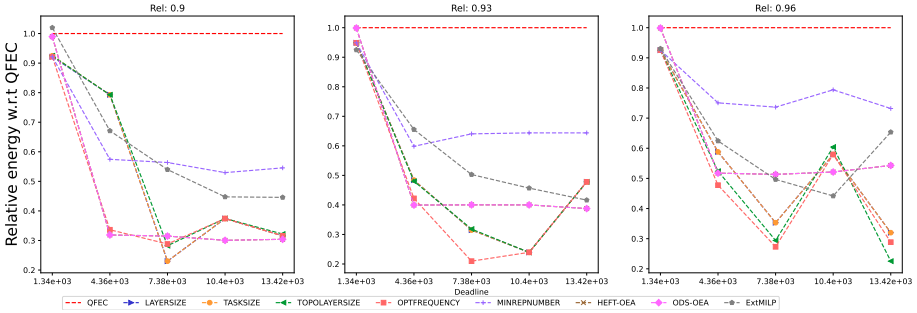

Figure 1783: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

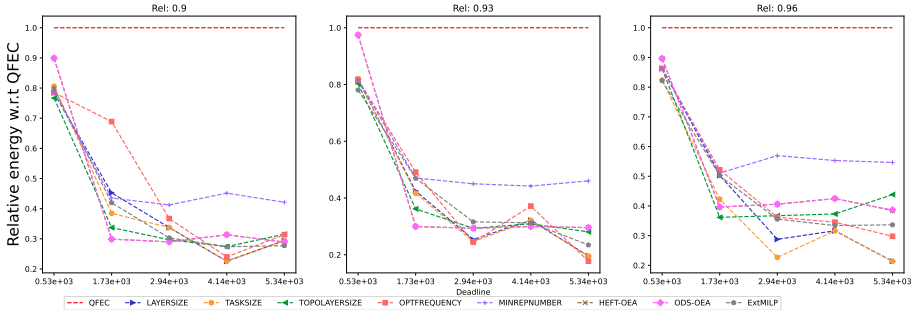

Figure 1784: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

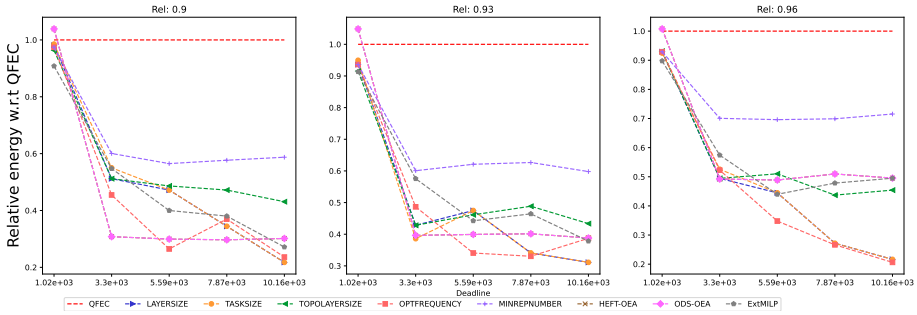

Figure 1785: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

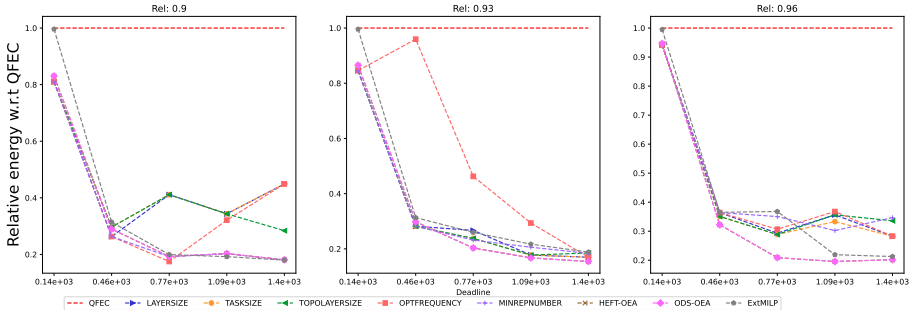

Figure 1786: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

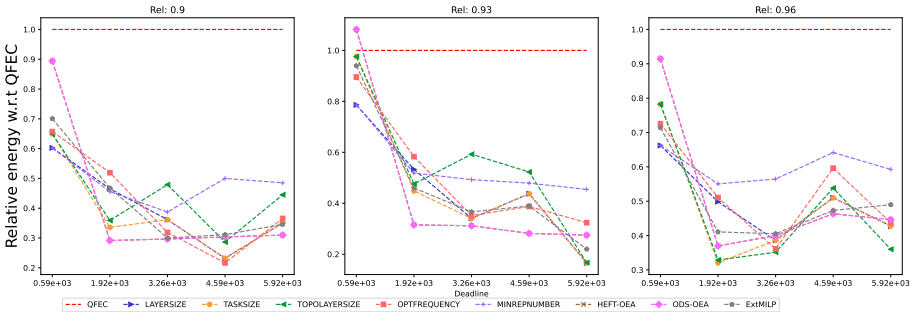

Figure 1787: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

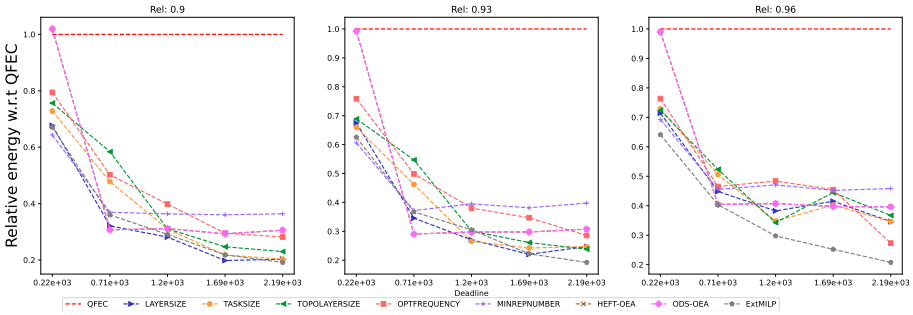

Figure 1788: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

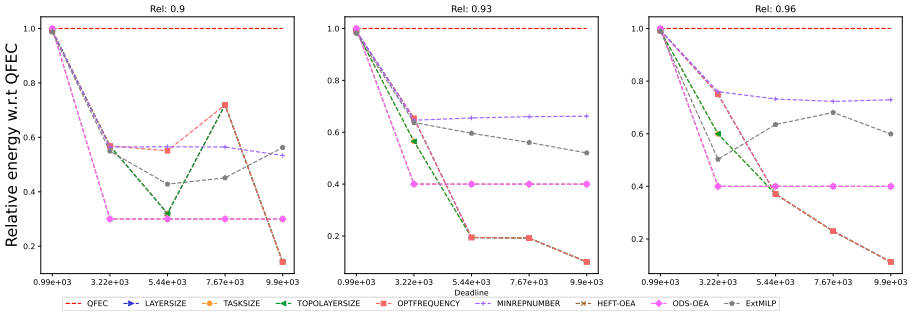

Figure 1789: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

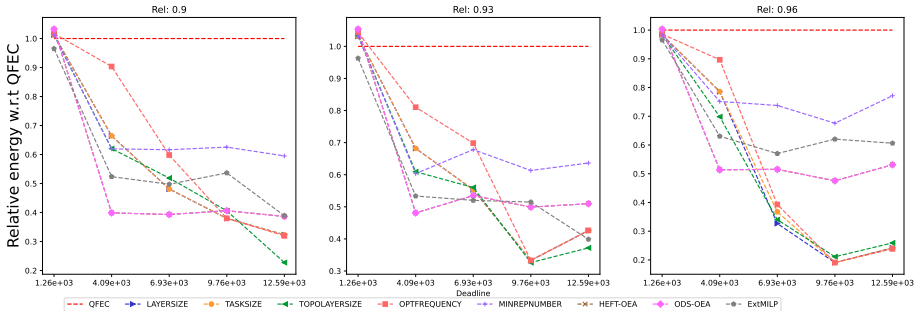

Figure 1790: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

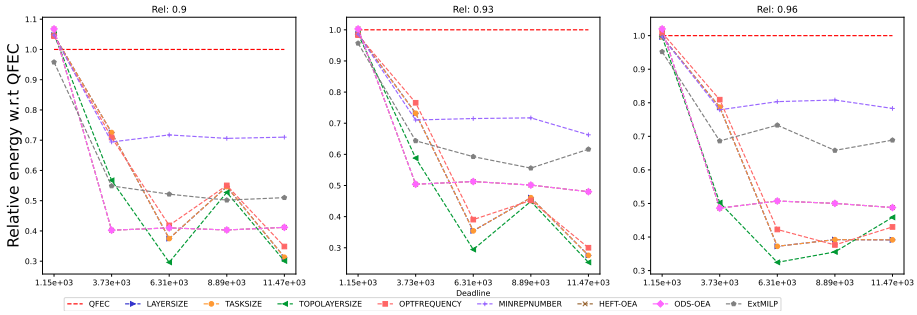

Figure 1791: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

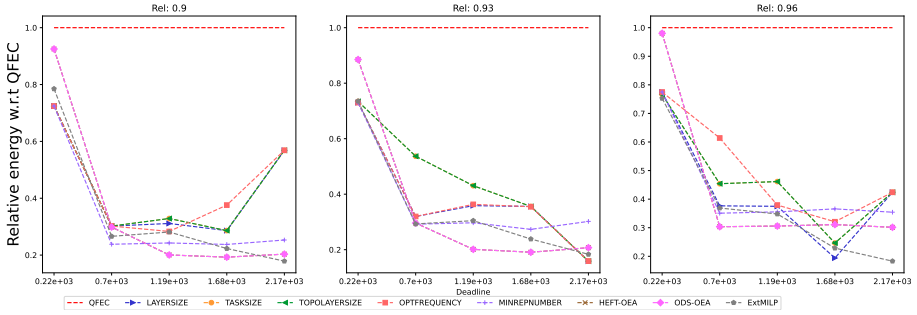

Figure 1792: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

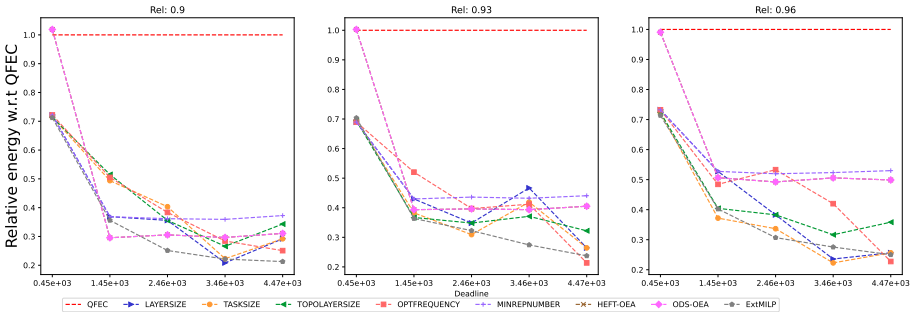

Figure 1793: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

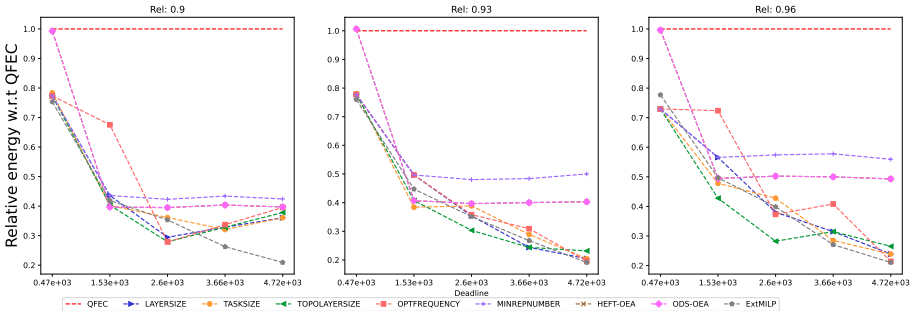

Figure 1794: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

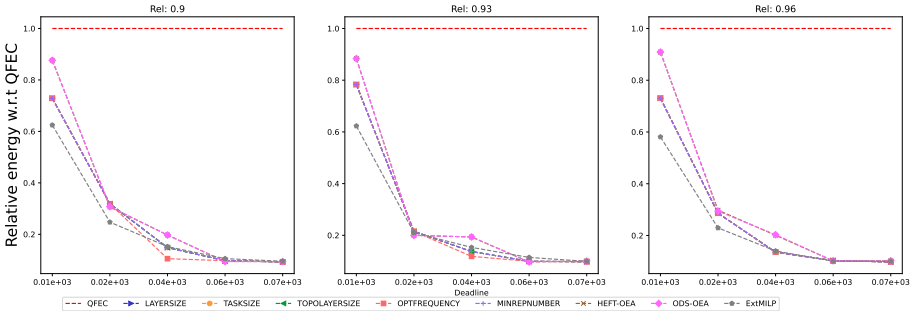

Figure 1795: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

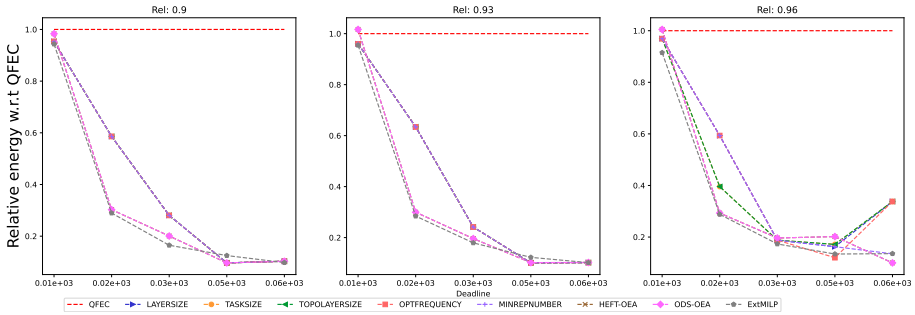

Figure 1796: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

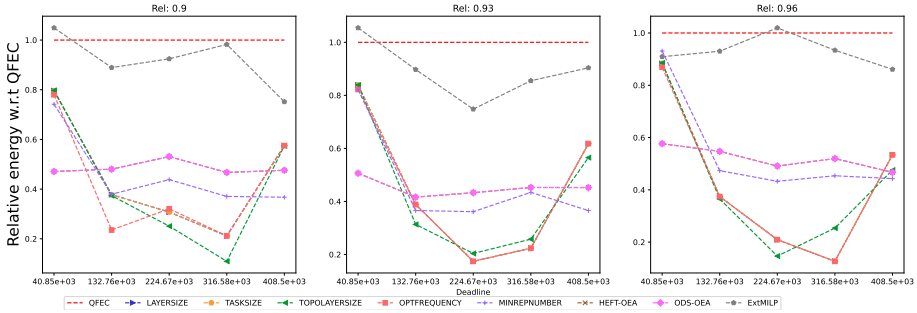

Figure 1797: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

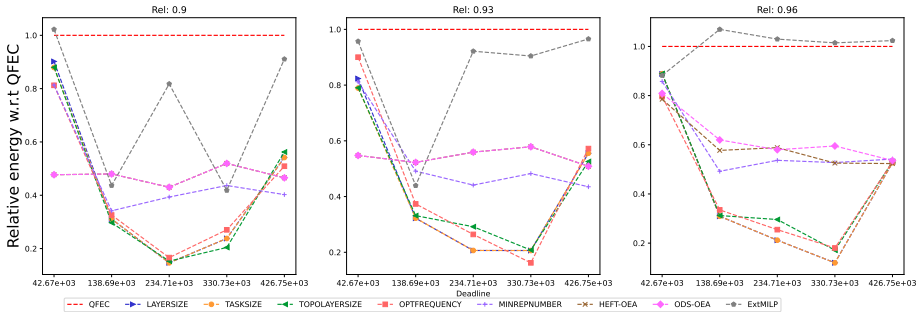

Figure 1798: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

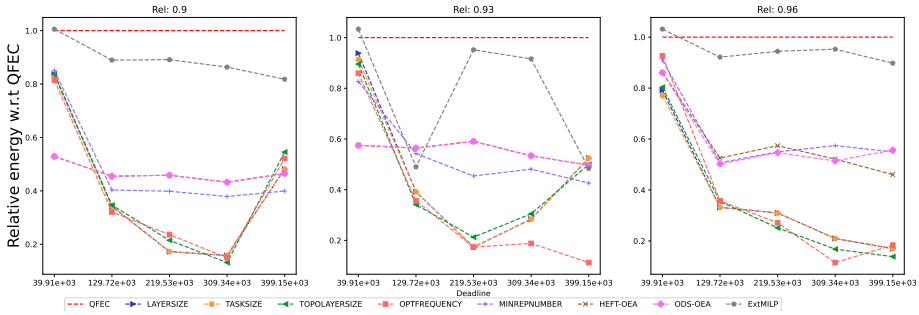

Figure 1799: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

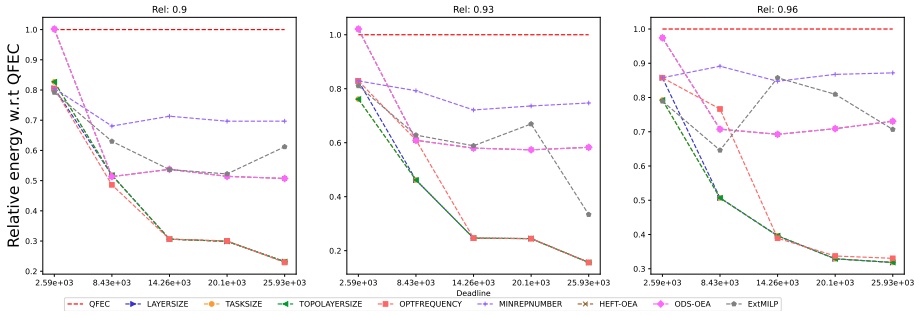

Figure 1800: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

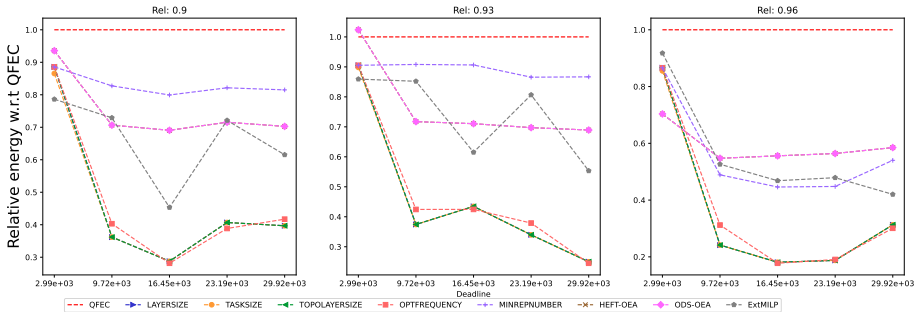

Figure 1801: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.4.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

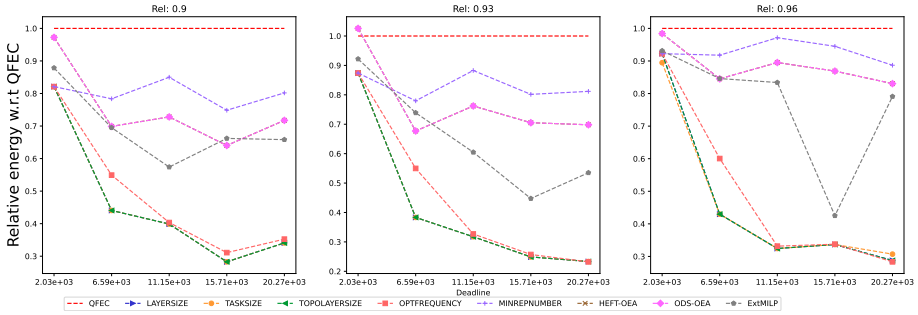

Figure 1802: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

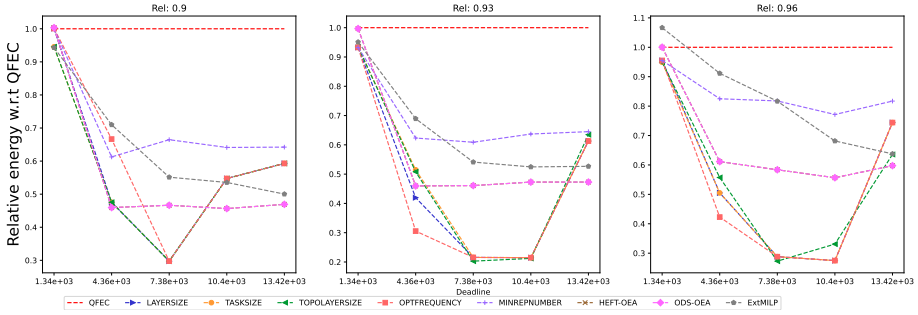

Figure 1803: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

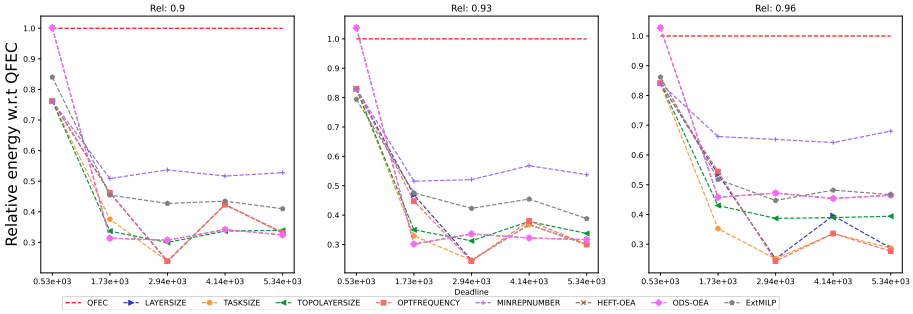

Figure 1804: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

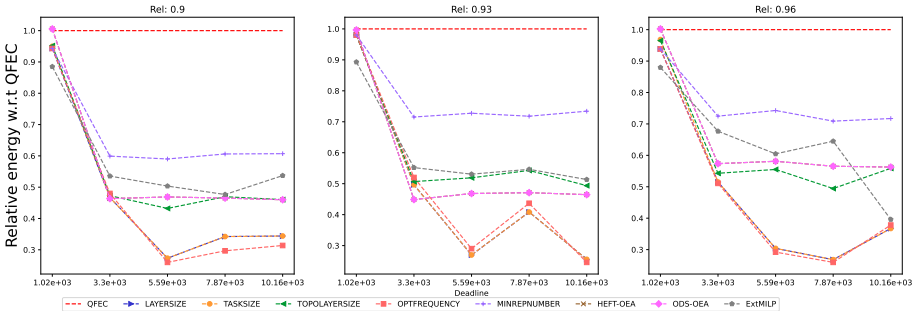

Figure 1805: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

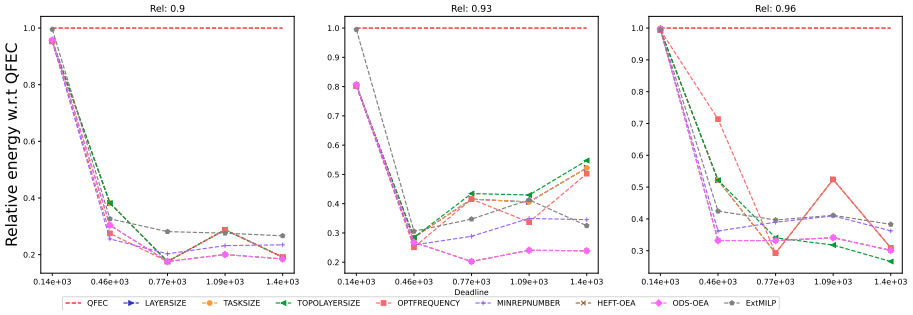

Figure 1806: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

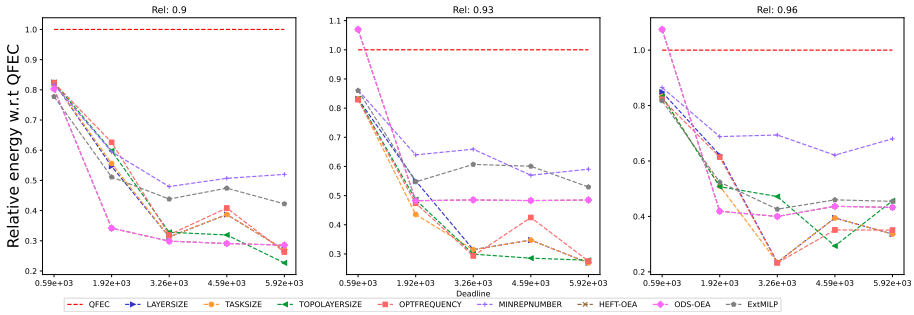

Figure 1807: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

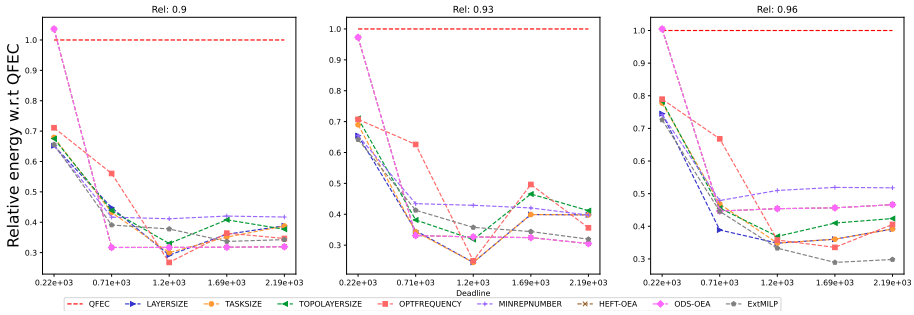

Figure 1808: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

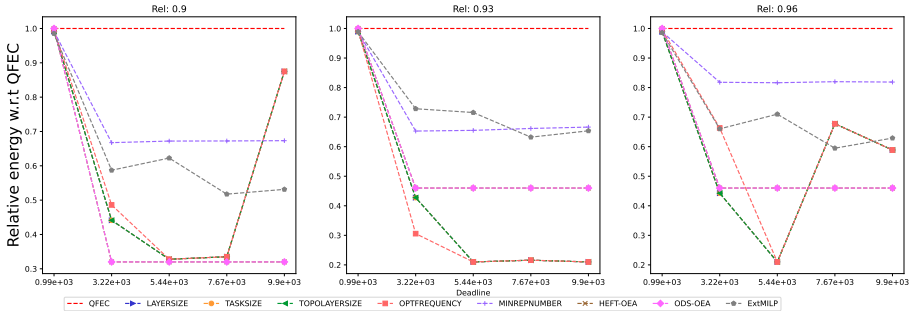

Figure 1809: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

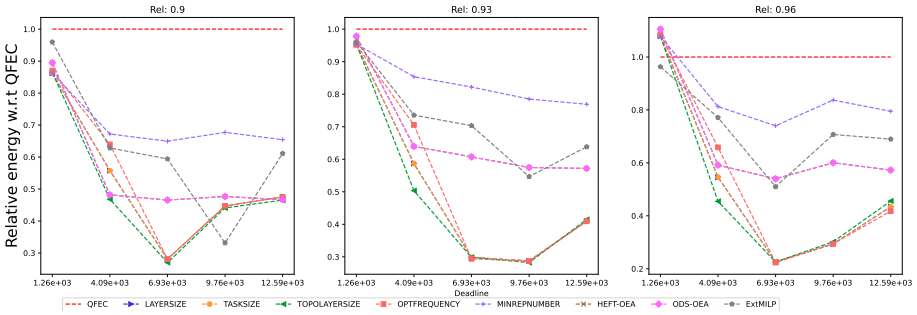

Figure 1810: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

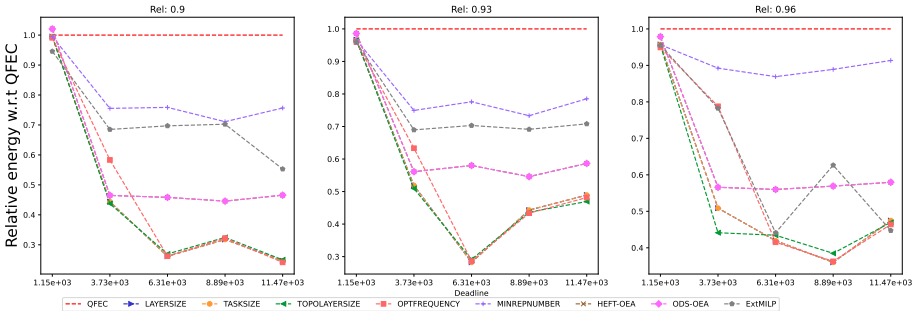

Figure 1811: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

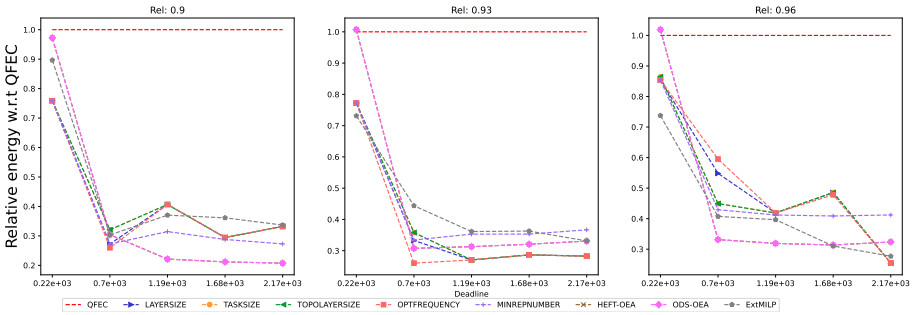

Figure 1812: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

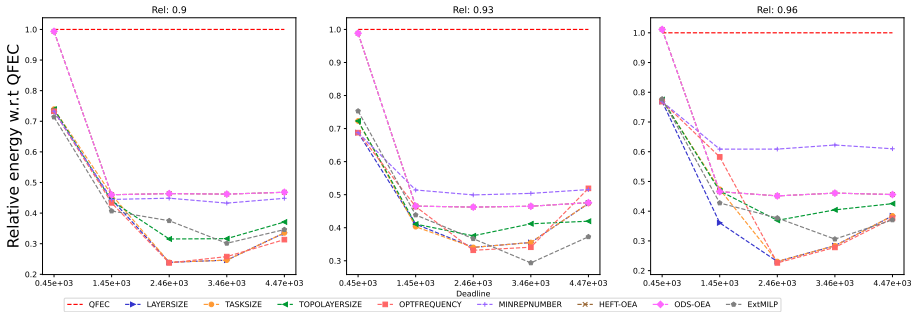

Figure 1813: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

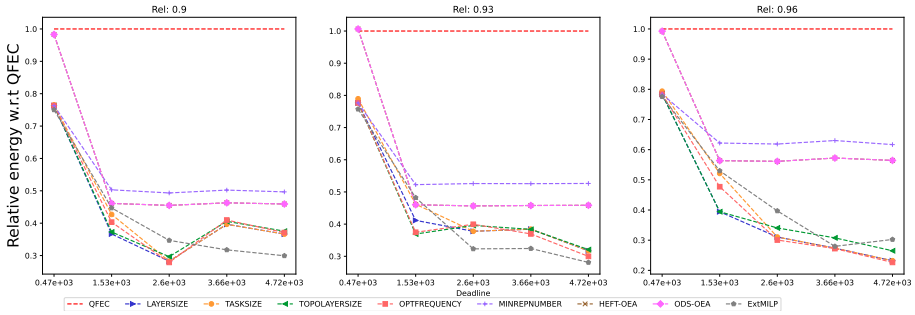

Figure 1814: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

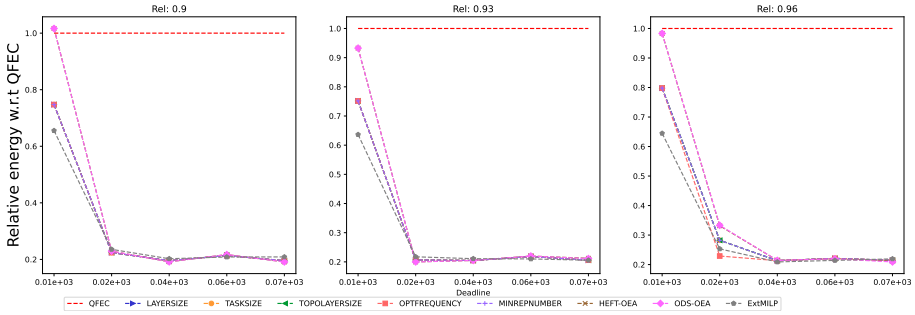

Figure 1815: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

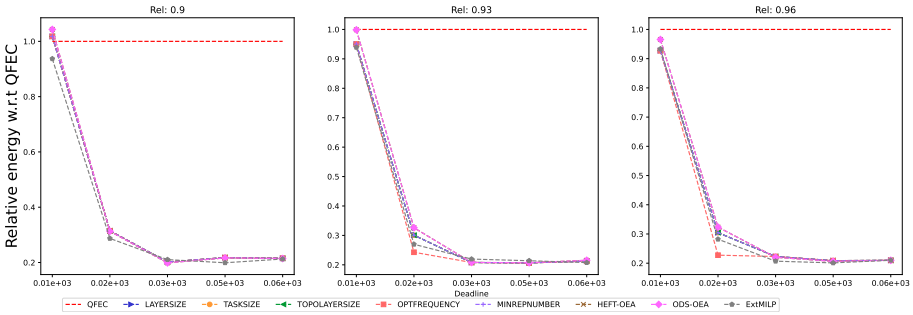

Figure 1816: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

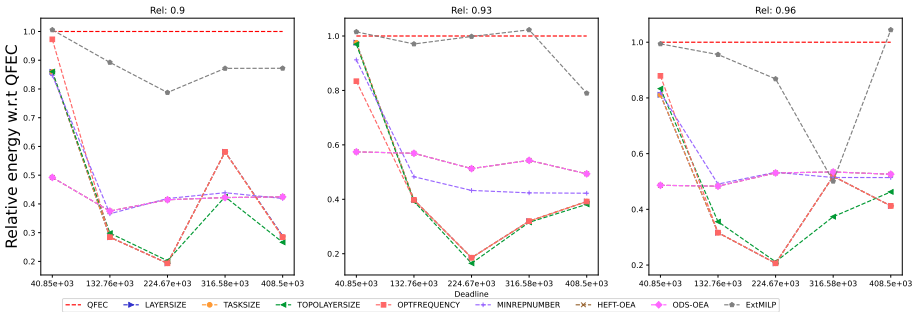

Figure 1817: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

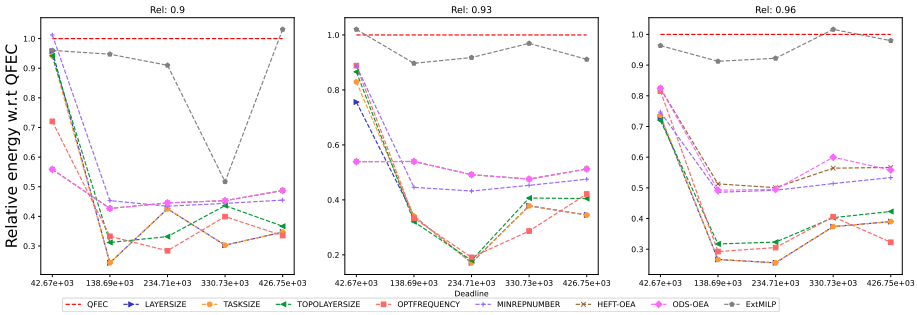

Figure 1818: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

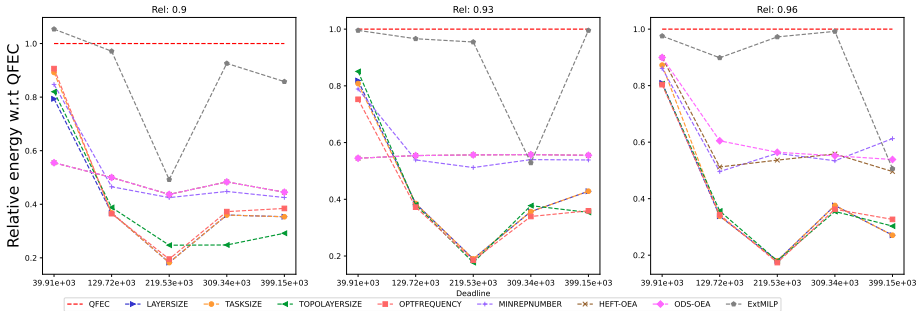

Figure 1819: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

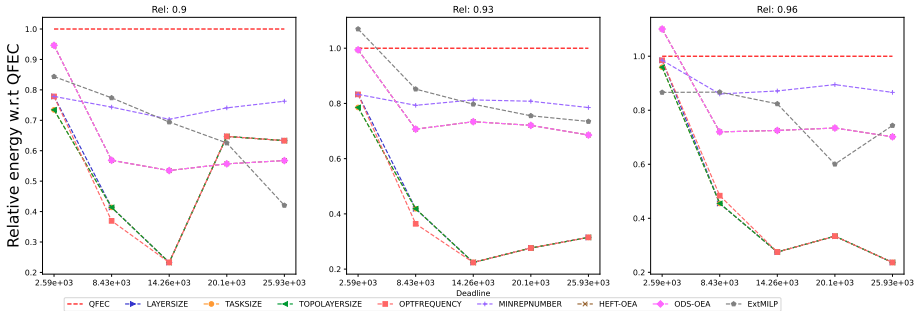

Figure 1820: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

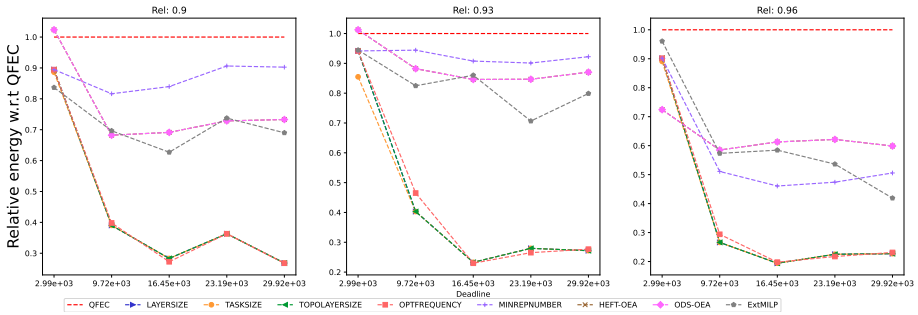

Figure 1821: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.4.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

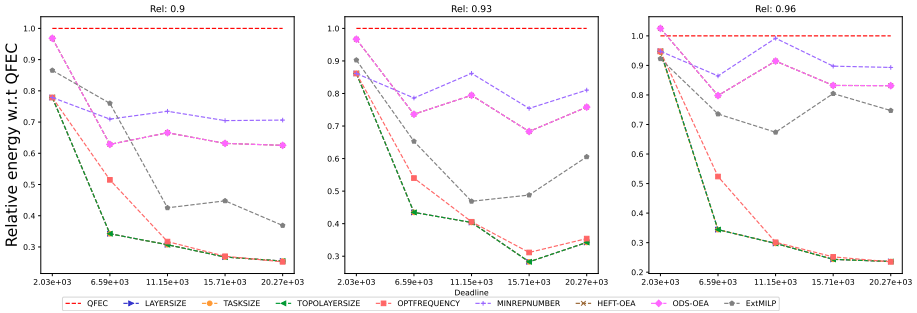

Figure 1822: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

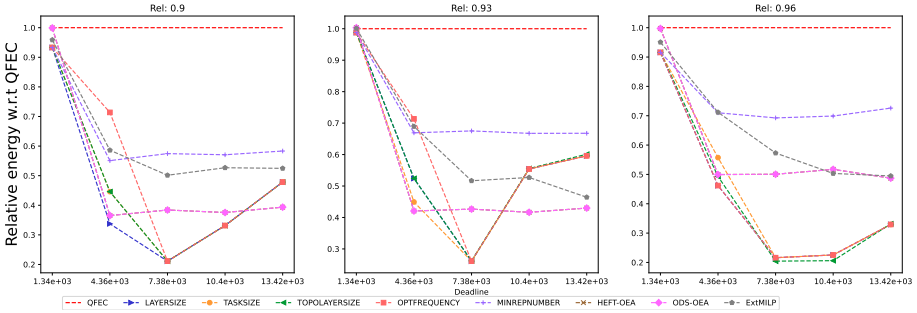

Figure 1823: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

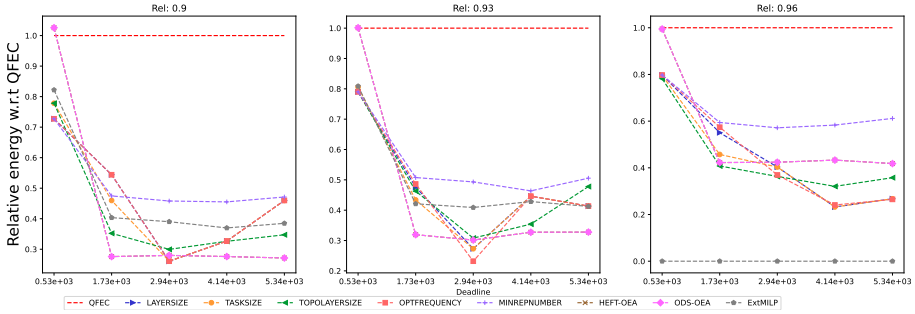

Figure 1824: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

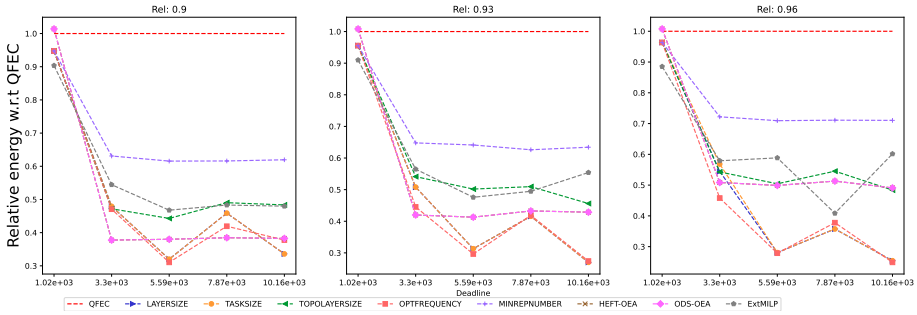

Figure 1825: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

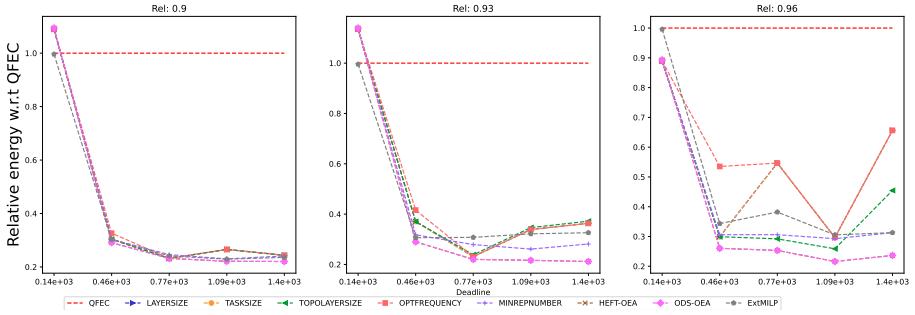

Figure 1826: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

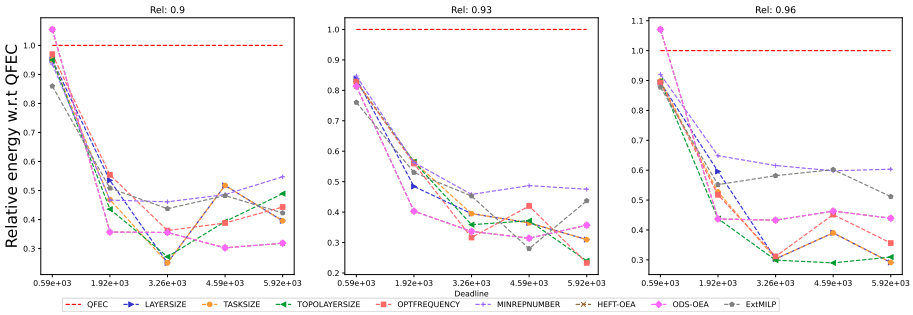

Figure 1827: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

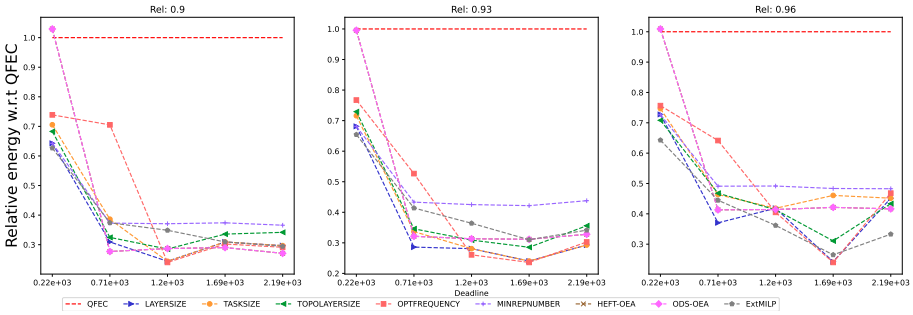

Figure 1828: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

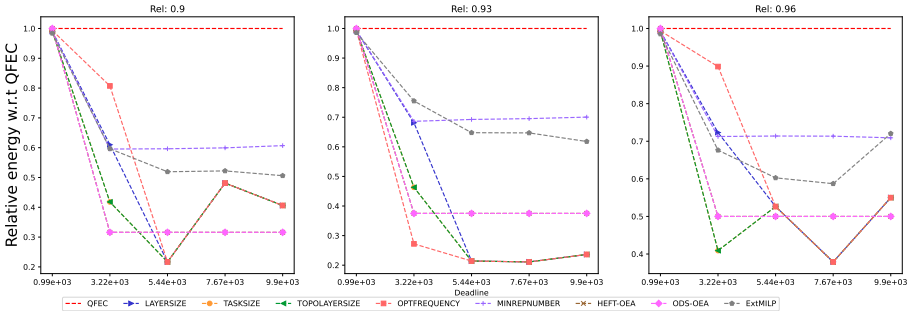

Figure 1829: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

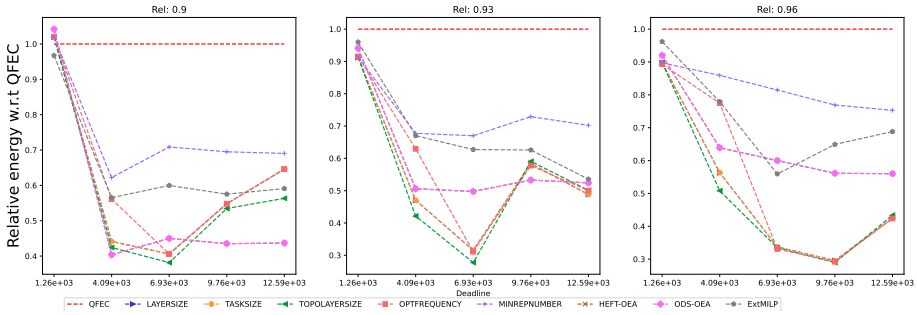

Figure 1830: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

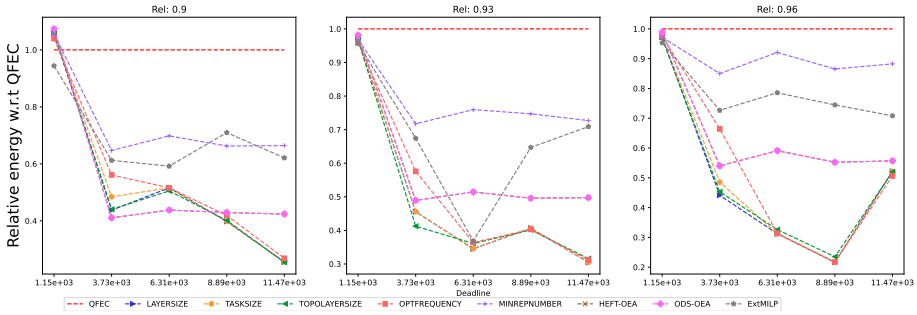

Figure 1831: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

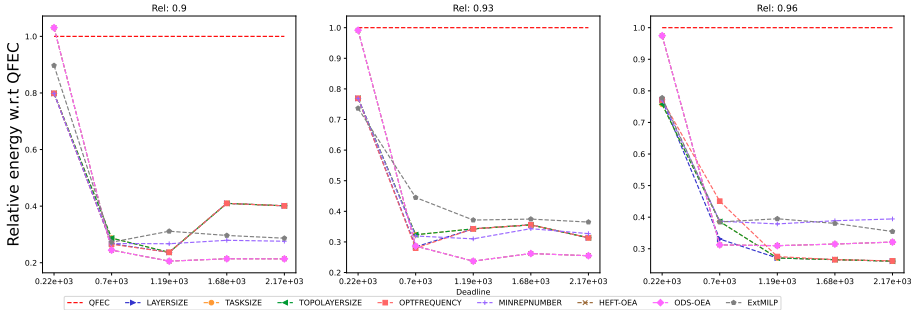

Figure 1832: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

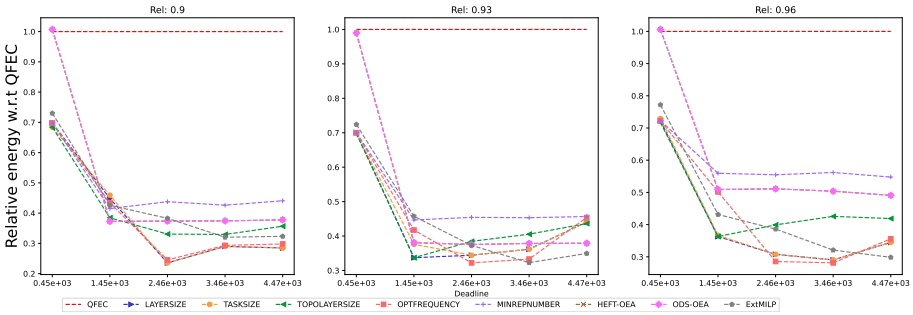

Figure 1833: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

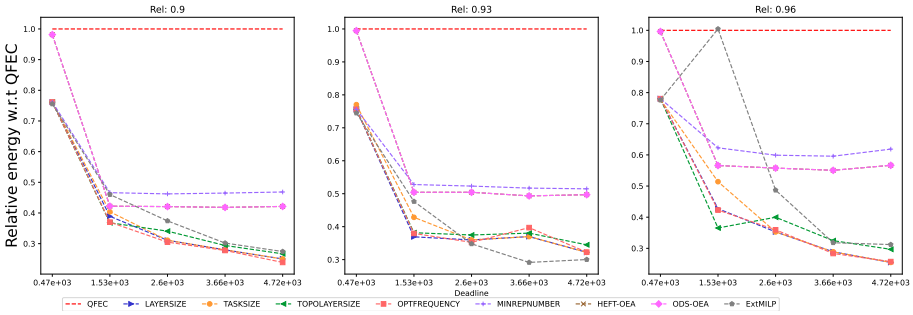

Figure 1834: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

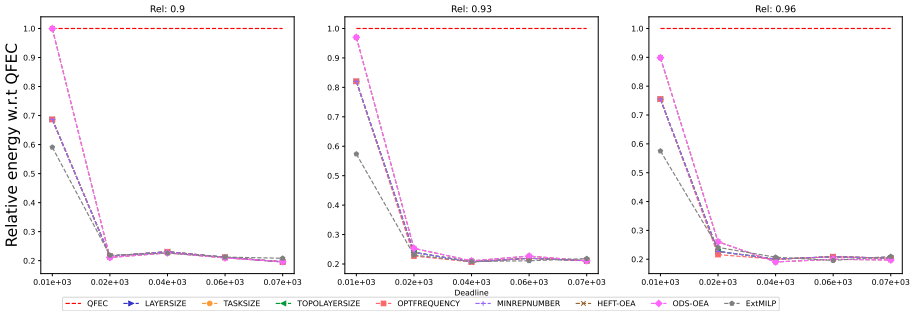

Figure 1835: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

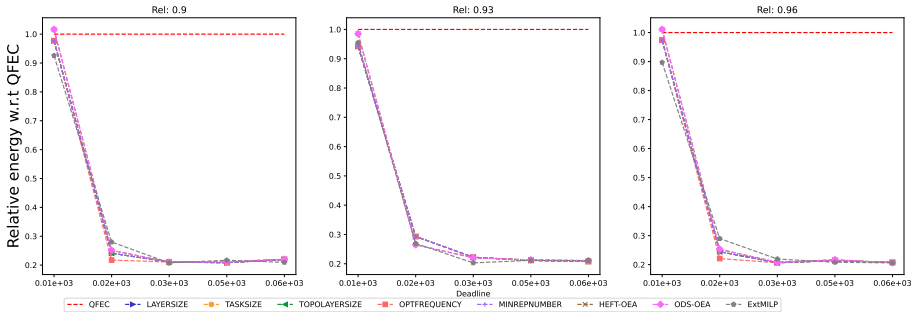

Figure 1836: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

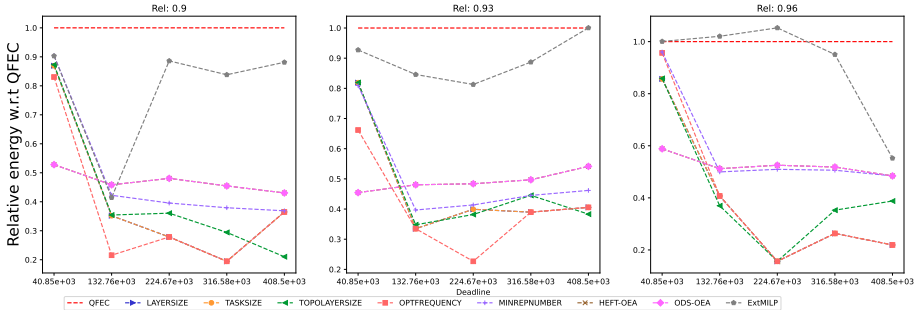

Figure 1837: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

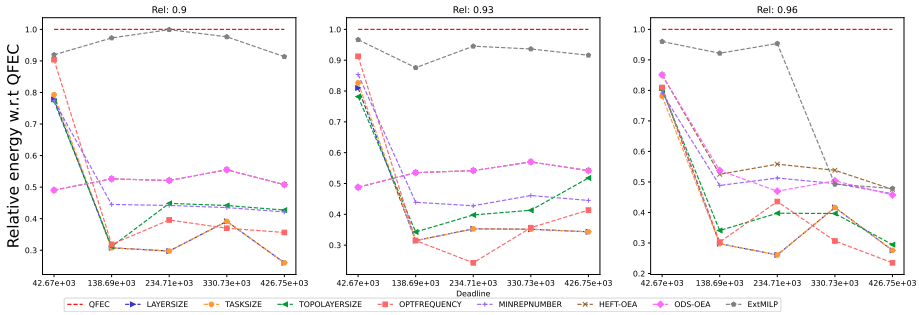

Figure 1838: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

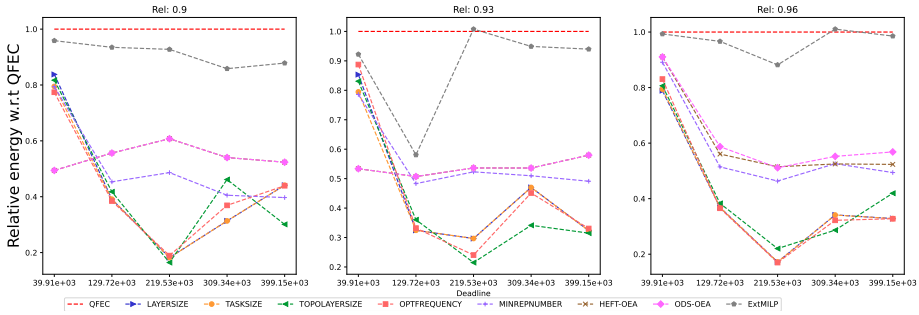

Figure 1839: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

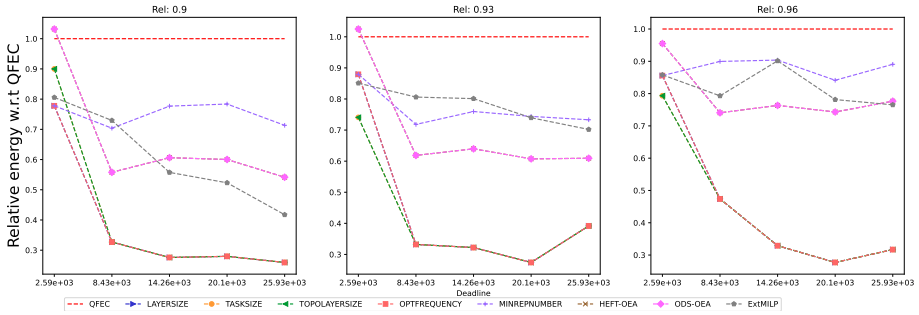

Figure 1840: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

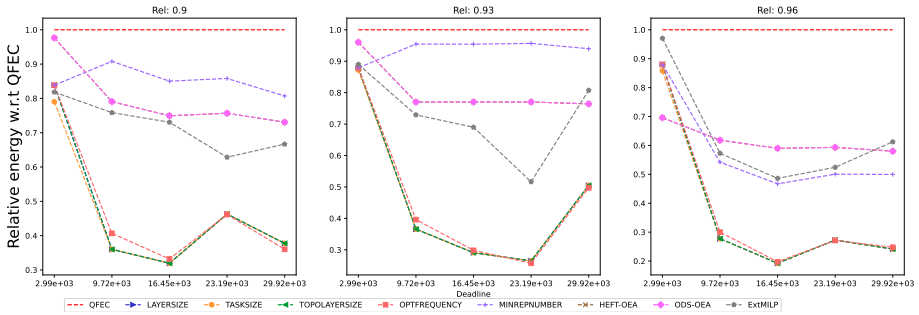

Figure 1841: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.5 $BC/WC = 0.5$

G.5.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

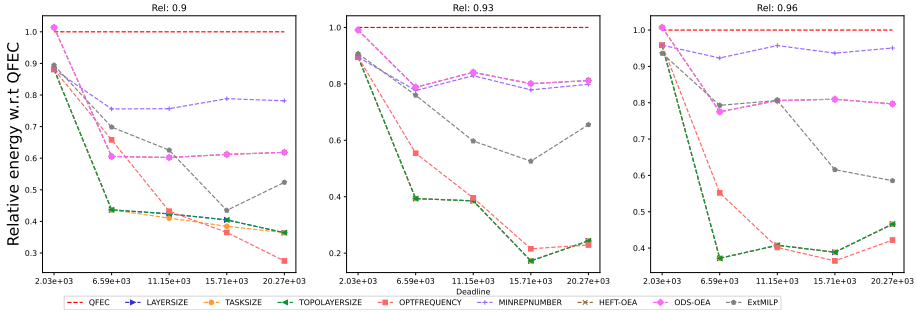

Figure 1842: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

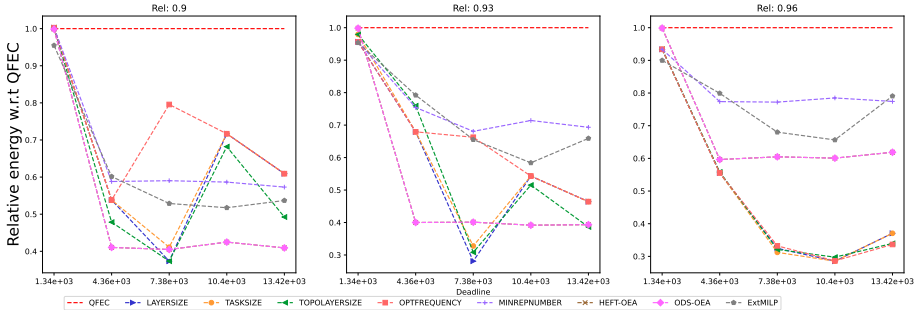

Figure 1843: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

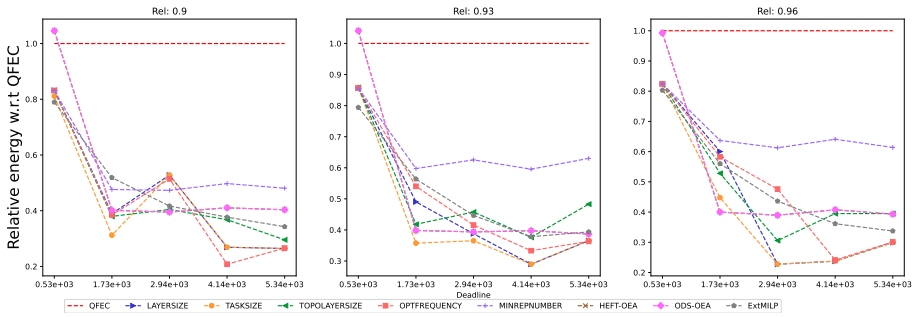

Figure 1844: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

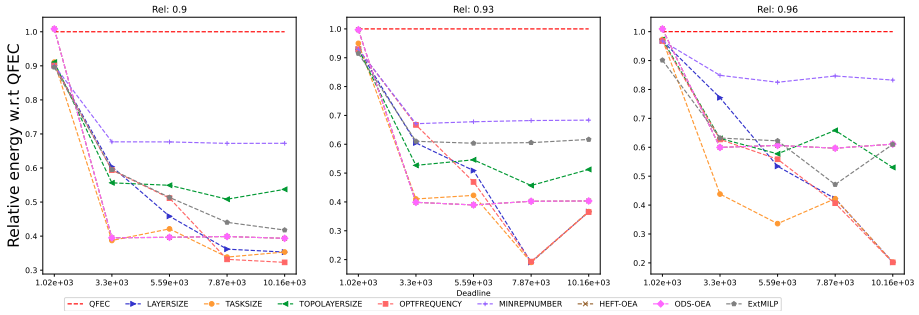

Figure 1845: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

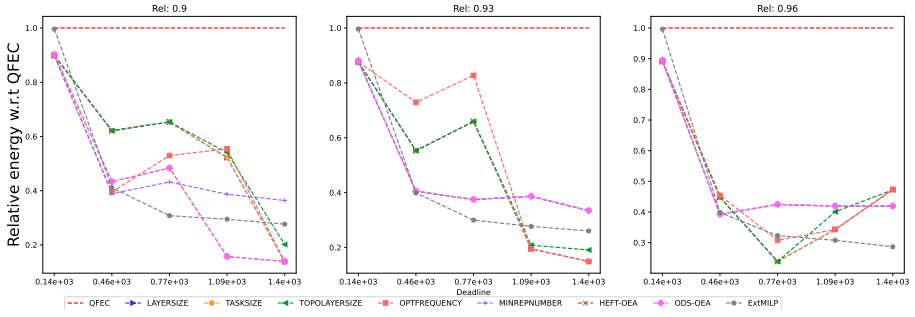

Figure 1846: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

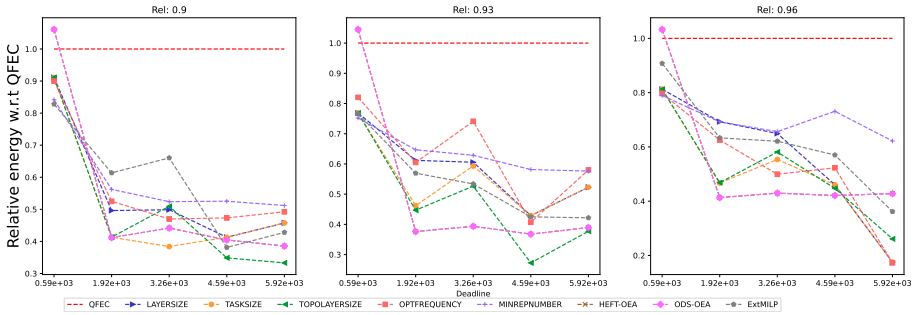

Figure 1847: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

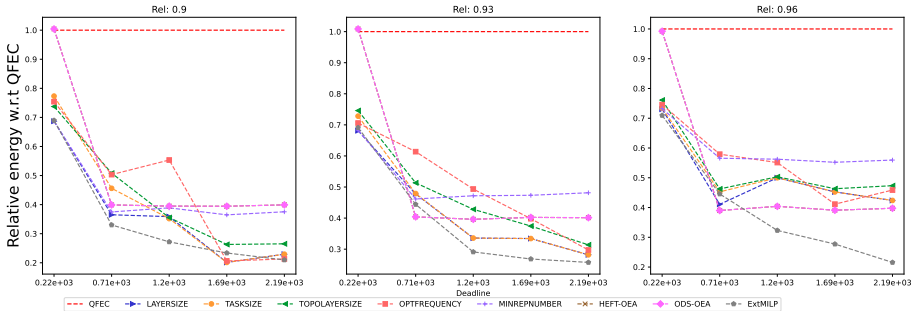

Figure 1848: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

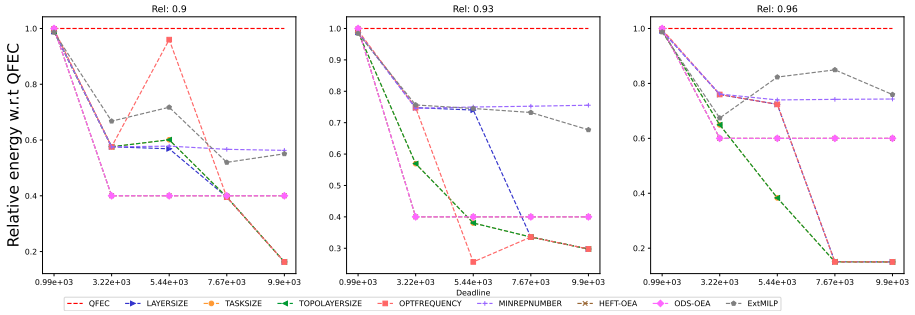

Figure 1849: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

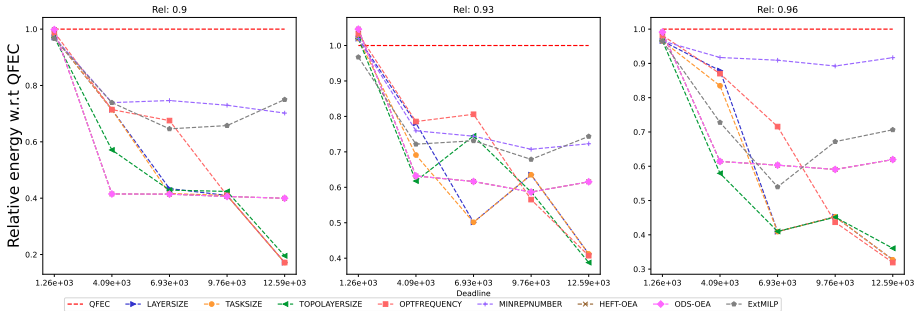

Figure 1850: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

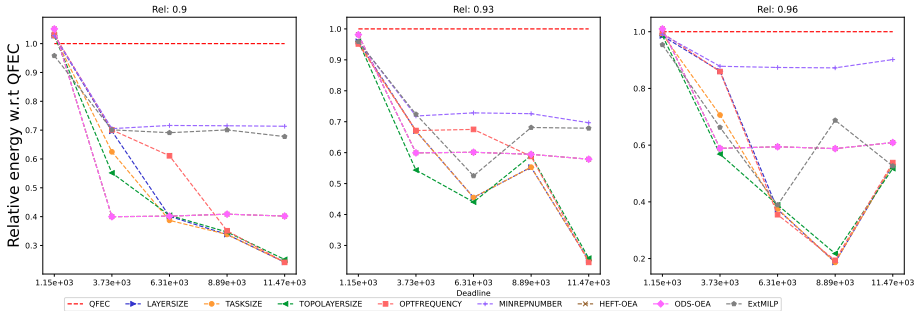

Figure 1851: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

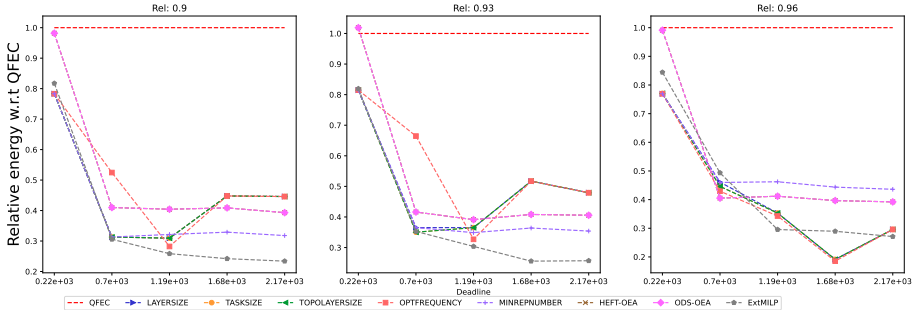

Figure 1852: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

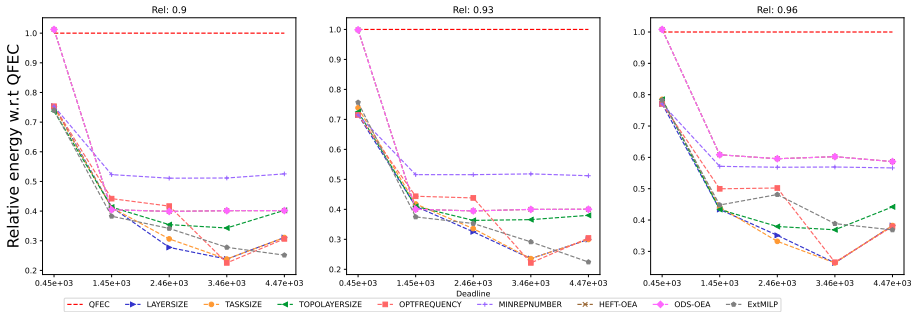

Figure 1853: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

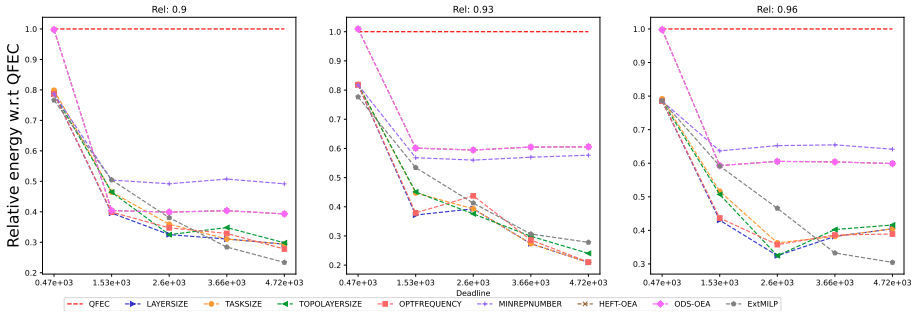

Figure 1854: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

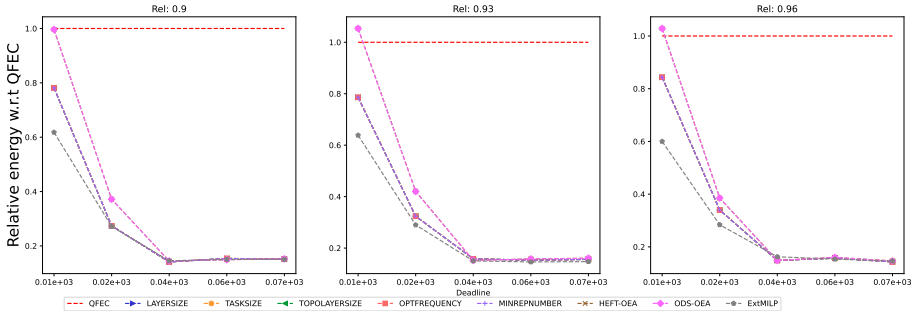

Figure 1855: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

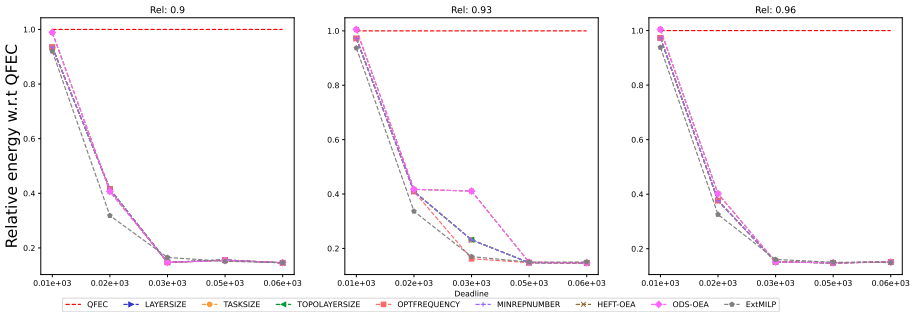

Figure 1856: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

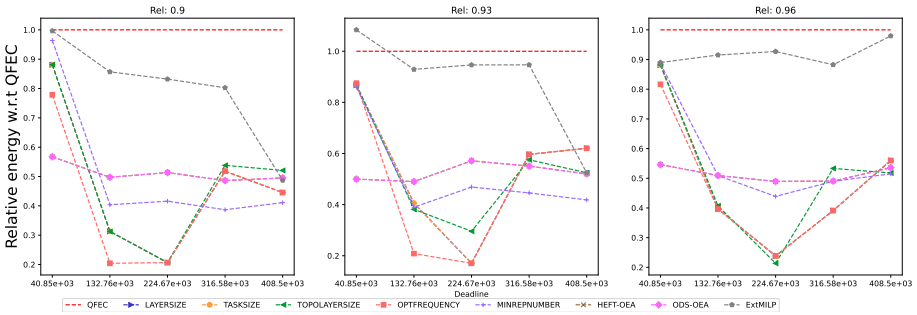

Figure 1857: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

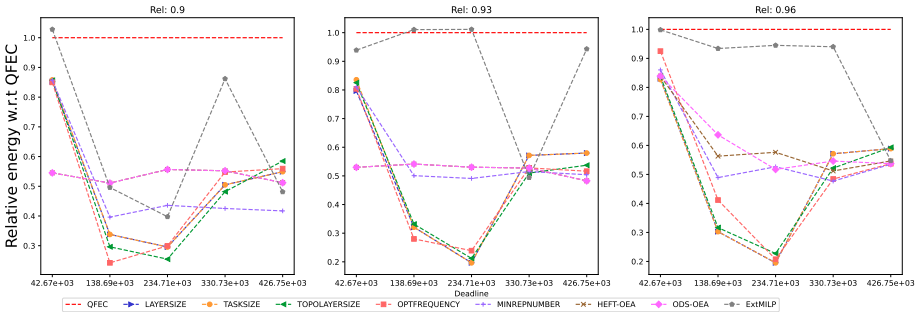

Figure 1858: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

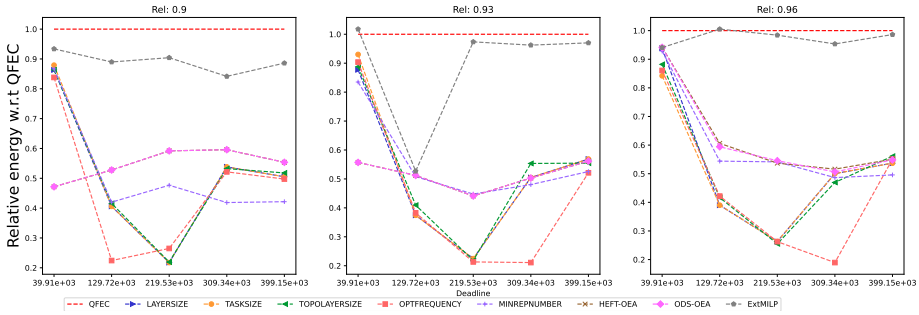

Figure 1859: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

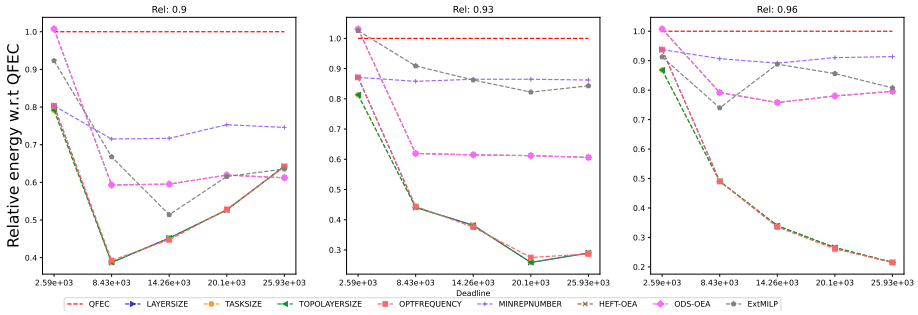

Figure 1860: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

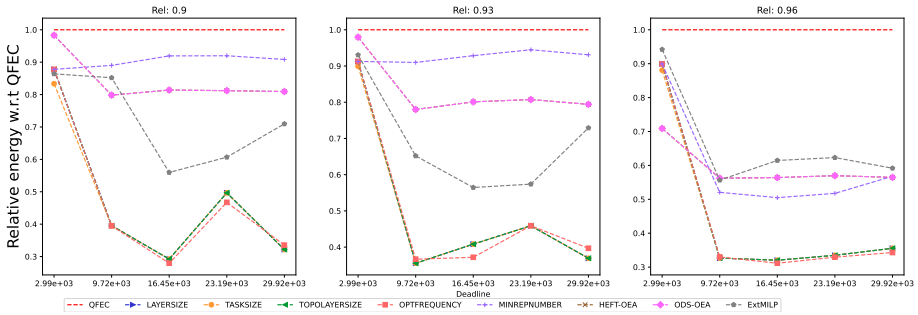

Figure 1861: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.5.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

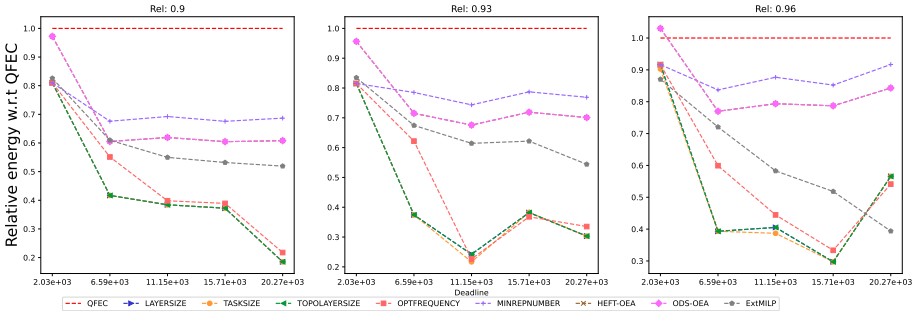

Figure 1862: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

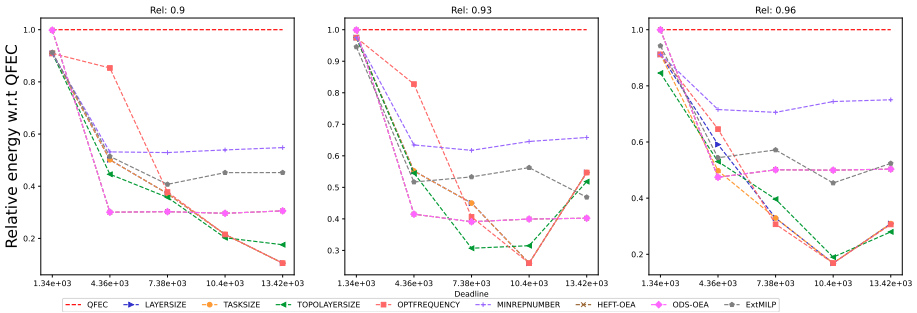

Figure 1863: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

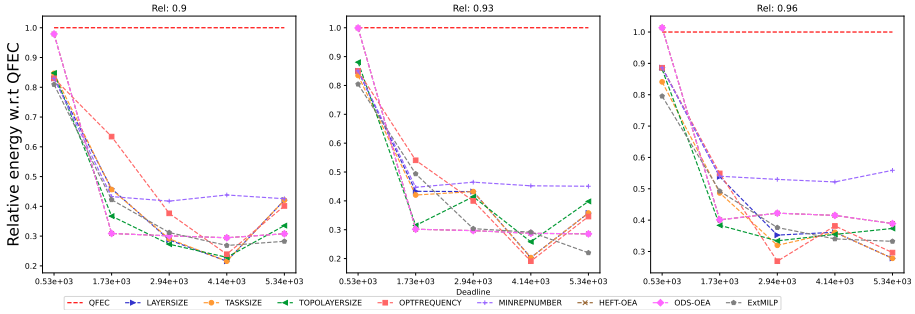

Figure 1864: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

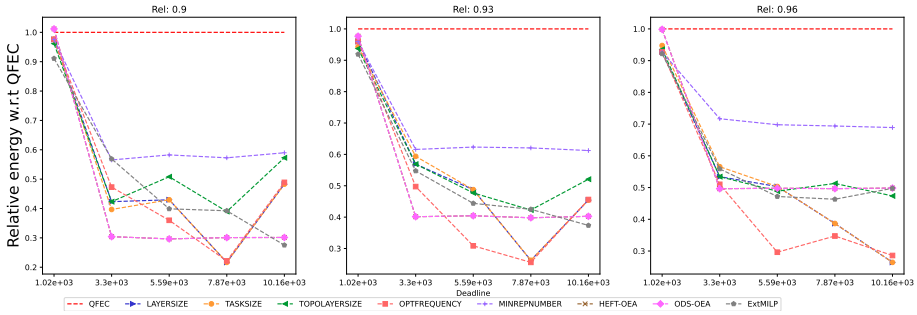

Figure 1865: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

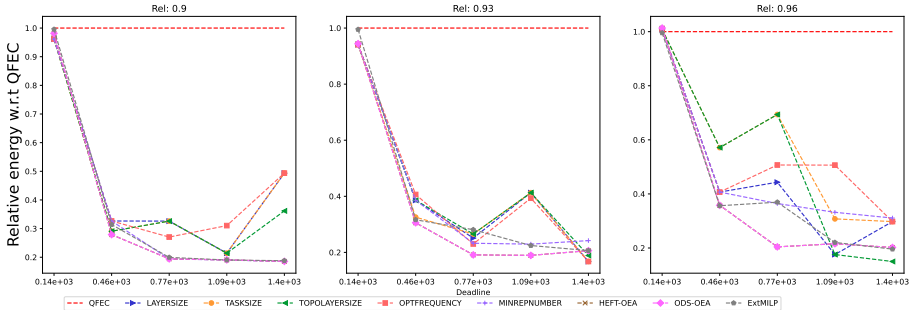

Figure 1866: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

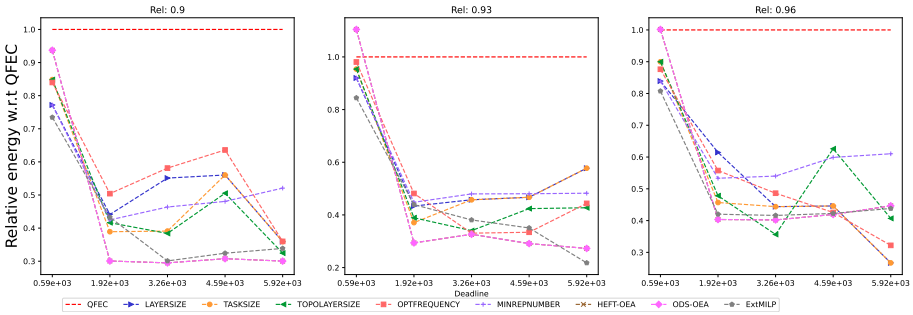

Figure 1867: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

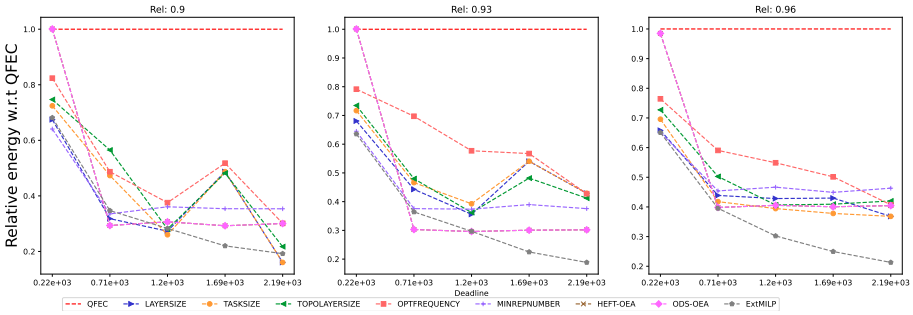

Figure 1868: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

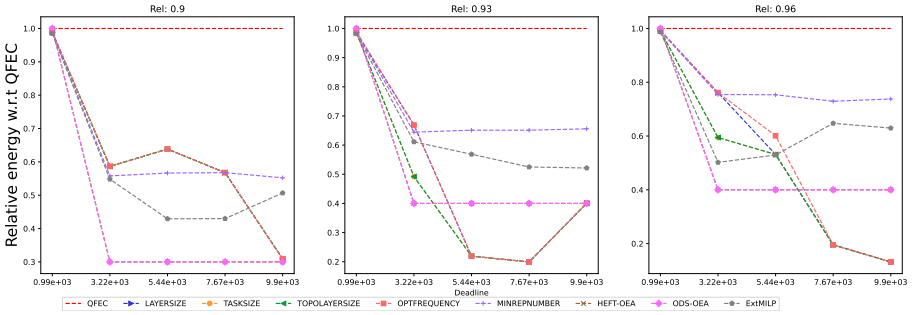

Figure 1869: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

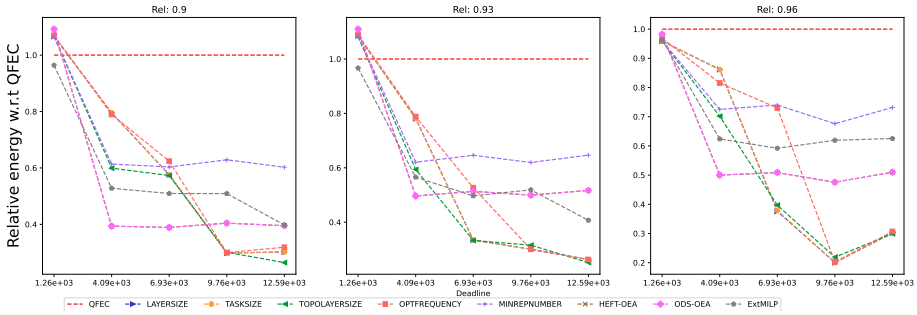

Figure 1870: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

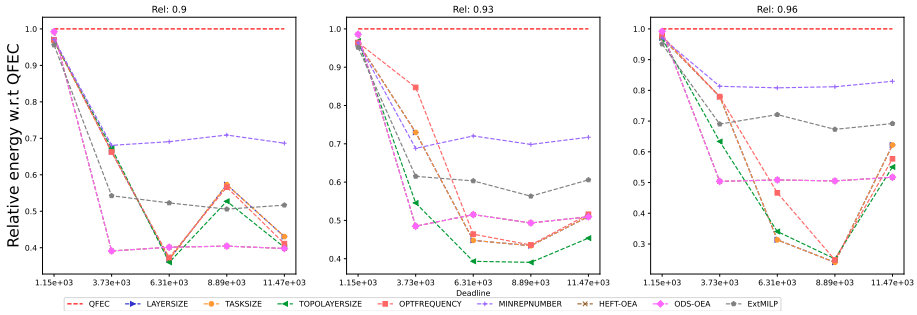

Figure 1871: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

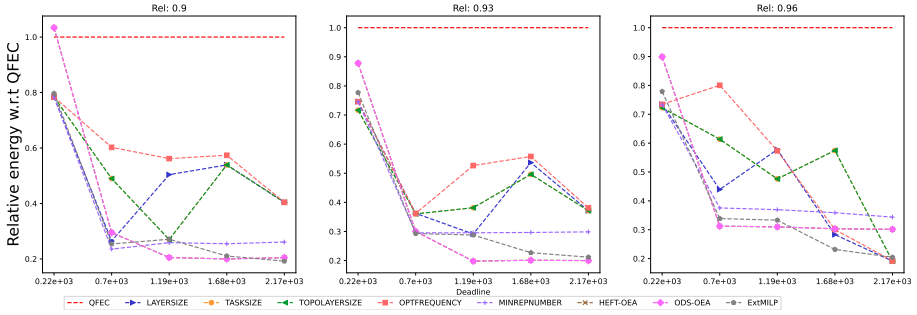

Figure 1872: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

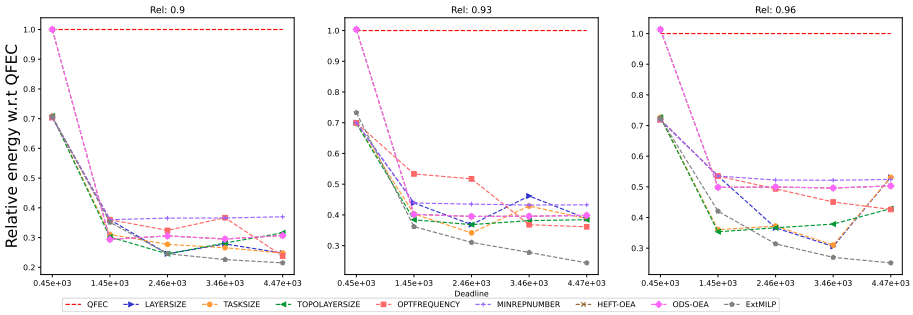

Figure 1873: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

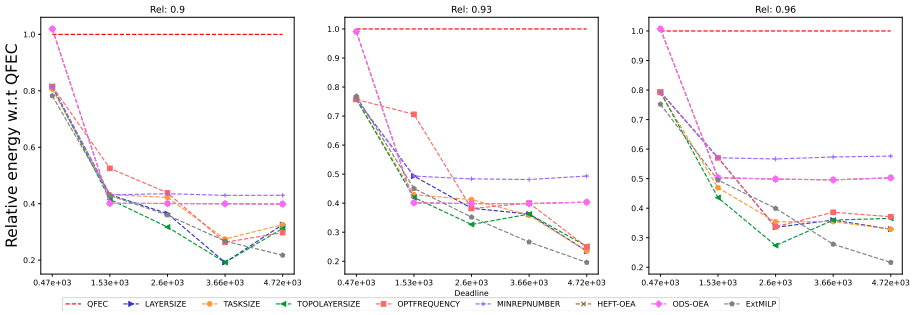

Figure 1874: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

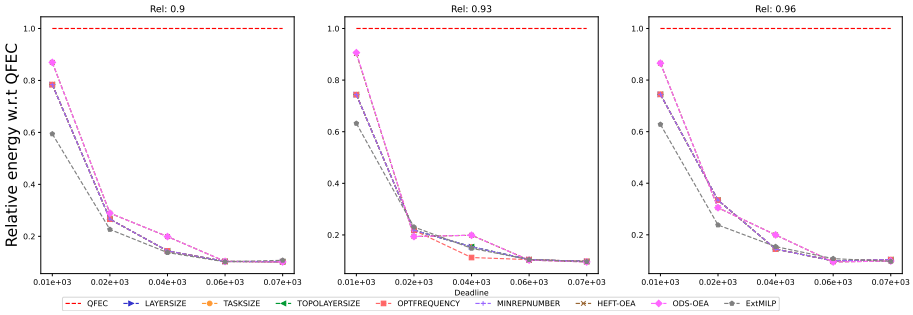

Figure 1875: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

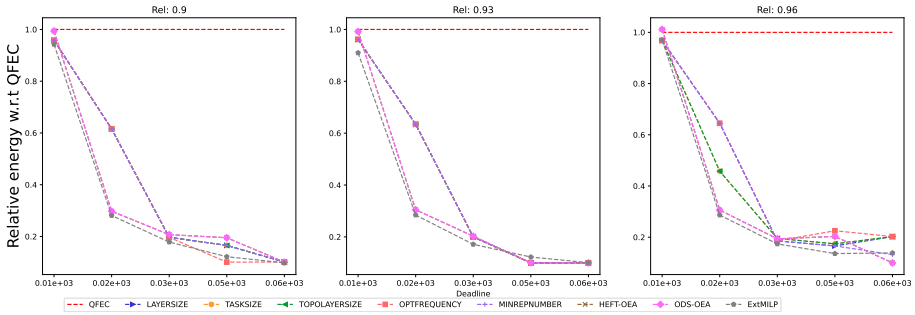

Figure 1876: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

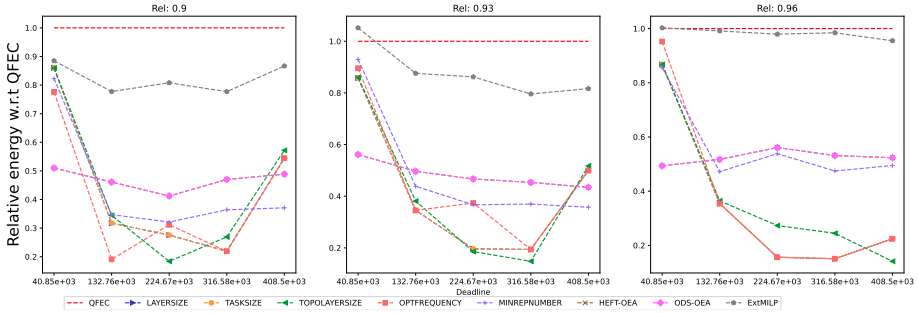

Figure 1877: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

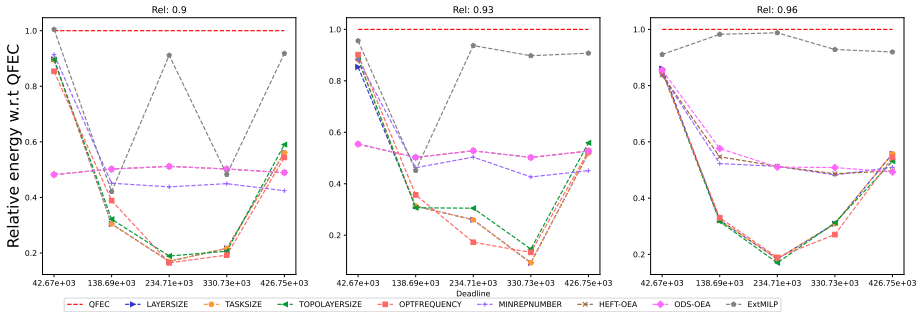

Figure 1878: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

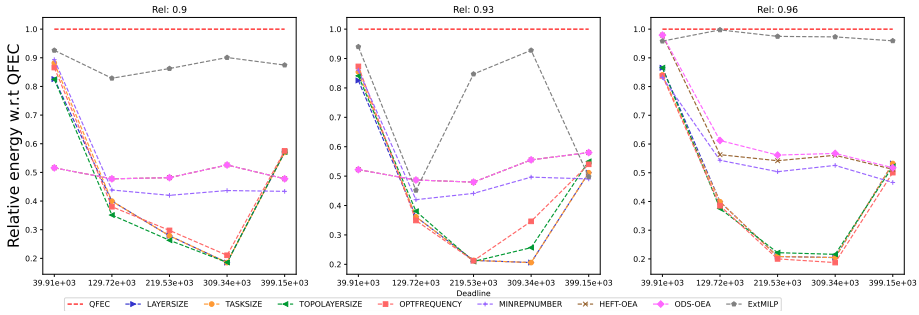

Figure 1879: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

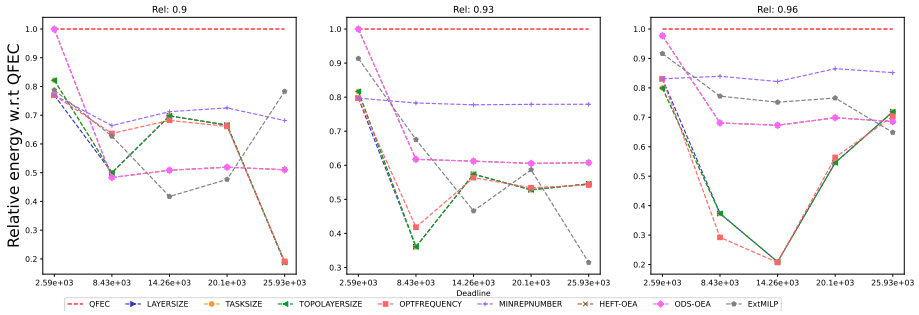

Figure 1880: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

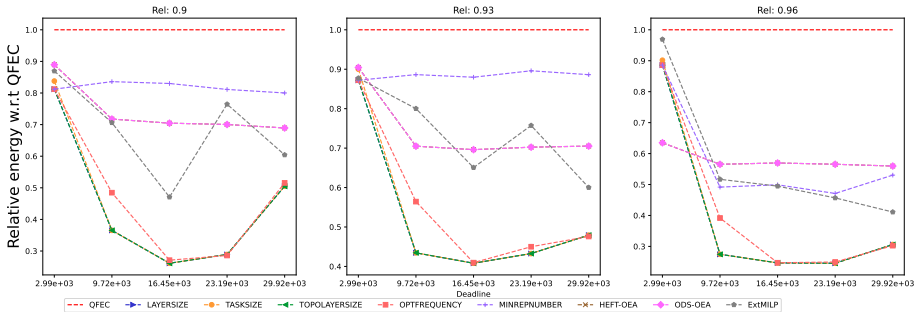

Figure 1881: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.5.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

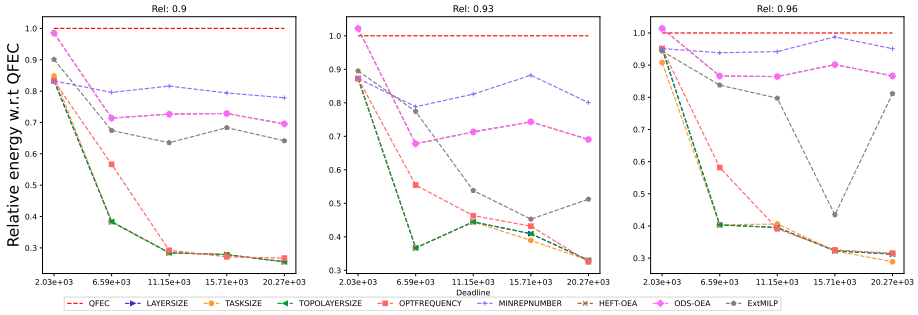

Figure 1882: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

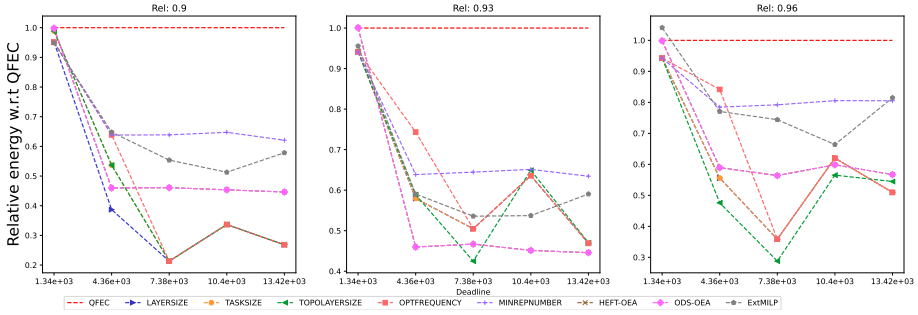

Figure 1883: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

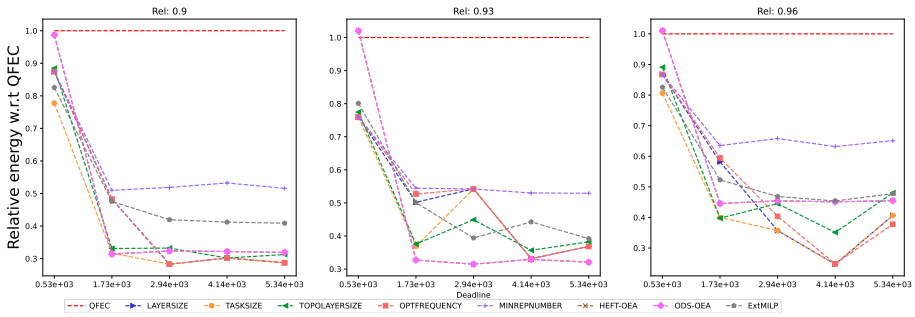

Figure 1884: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

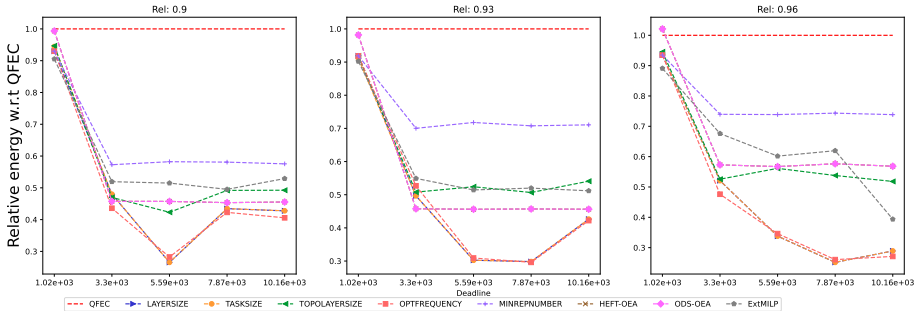

Figure 1885: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

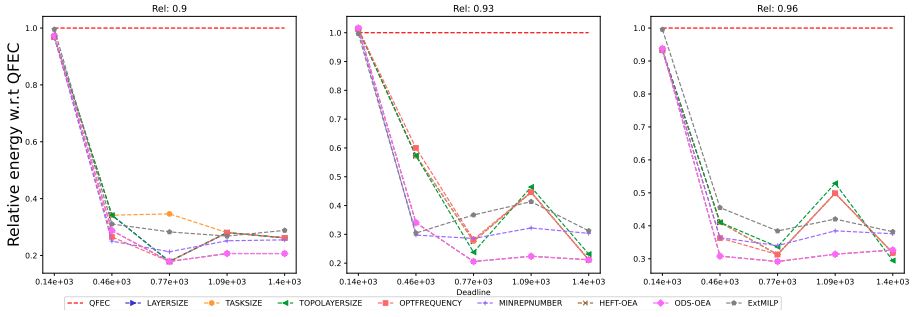

Figure 1886: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

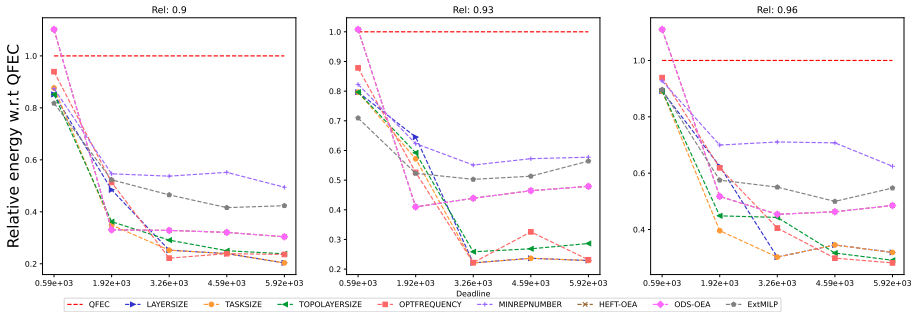

Figure 1887: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

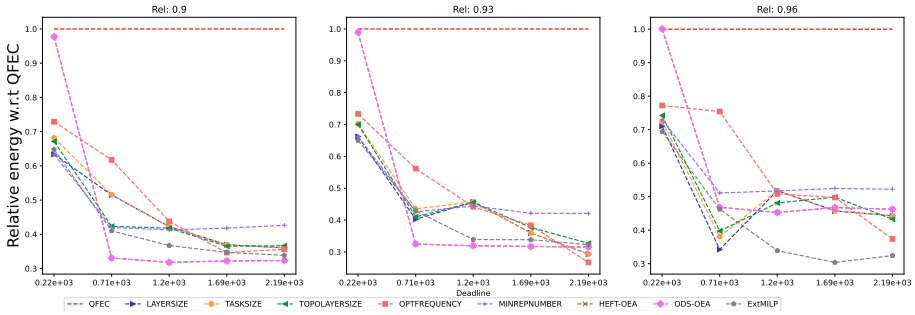

Figure 1888: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

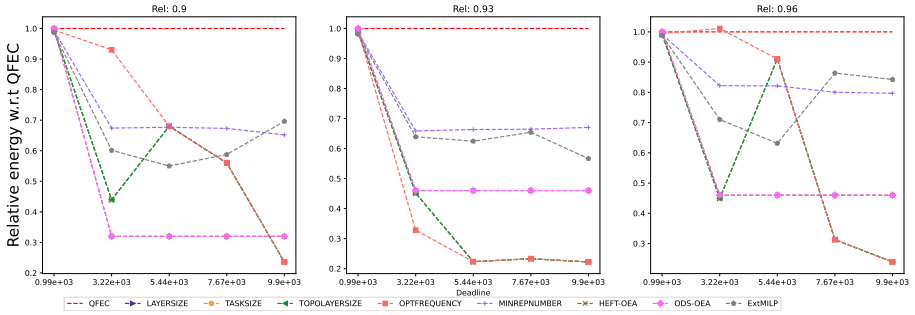

Figure 1889: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

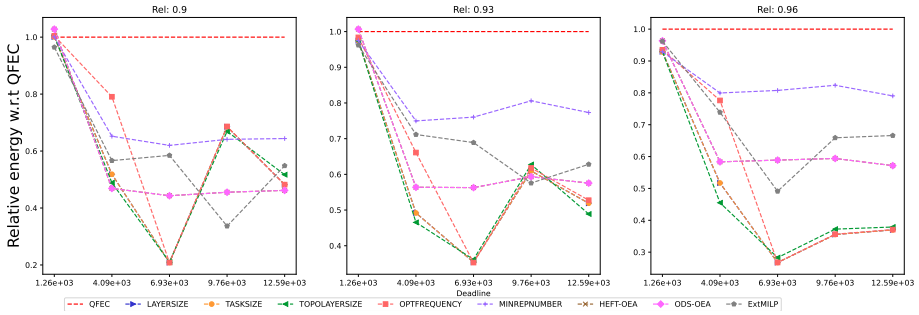

Figure 1890: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

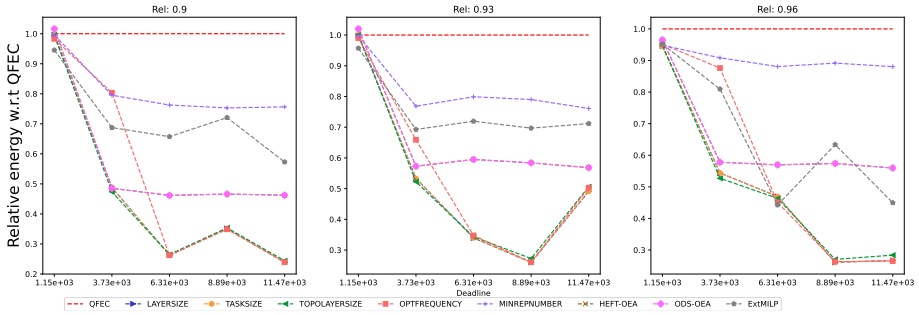

Figure 1891: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

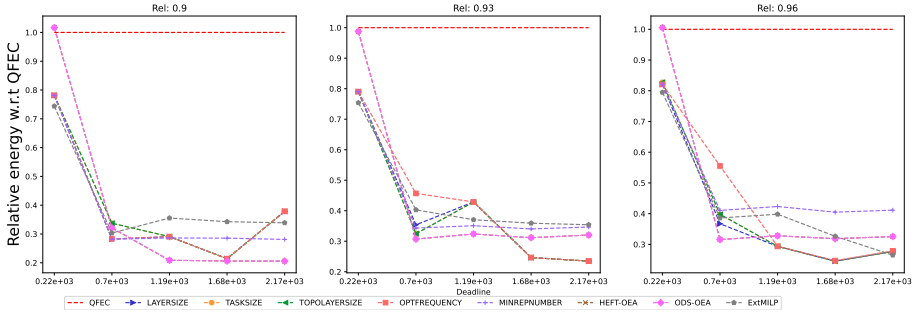

Figure 1892: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

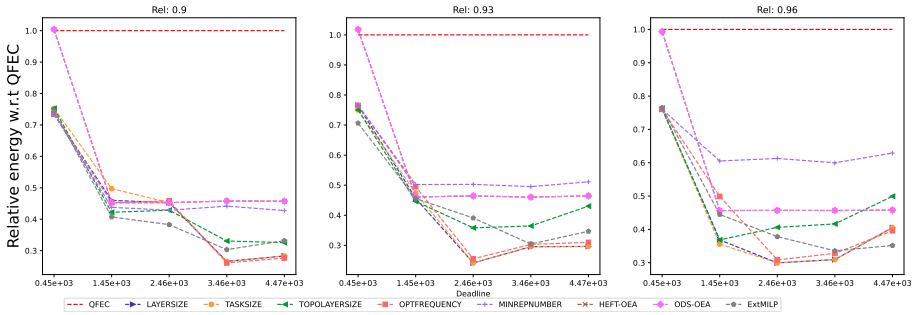

Figure 1893: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

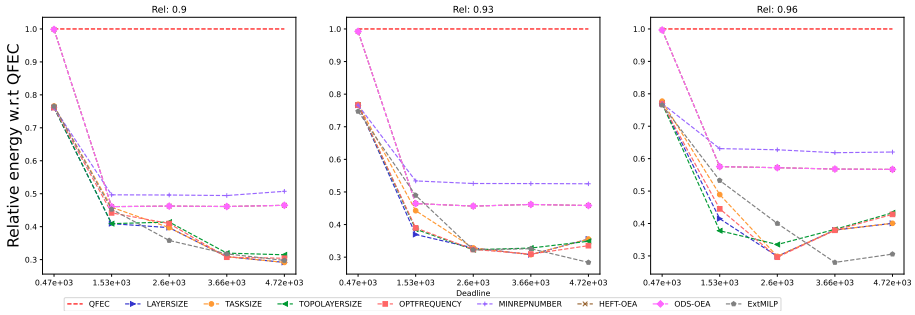

Figure 1894: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

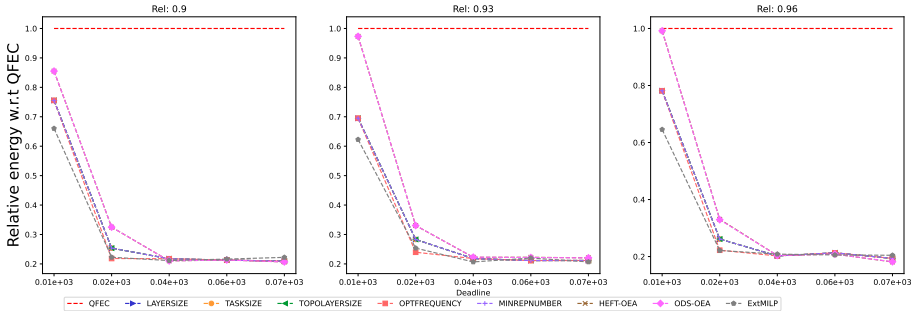

Figure 1895: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

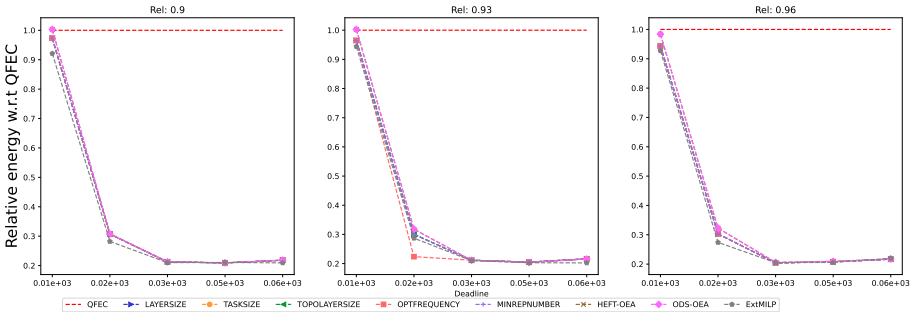

Figure 1896: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

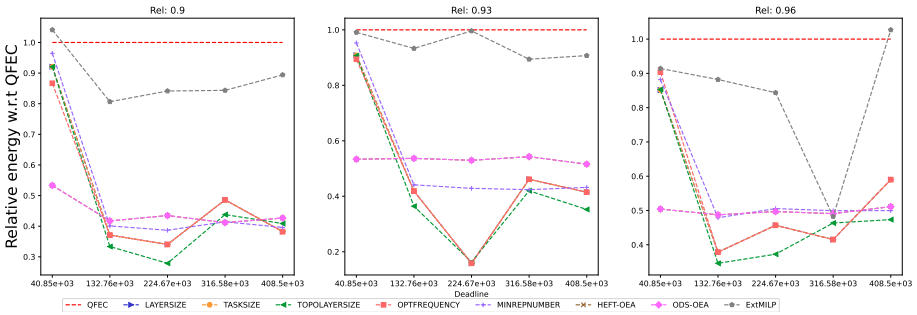

Figure 1897: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

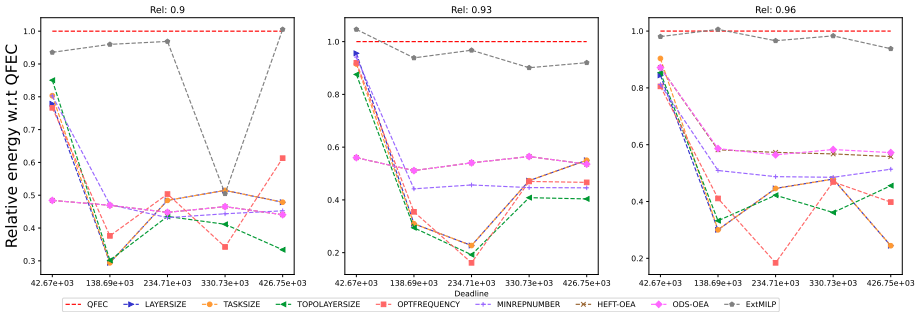

Figure 1898: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

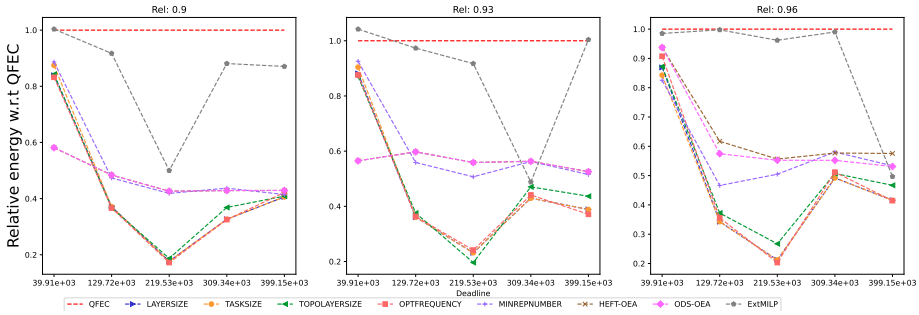

Figure 1899: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

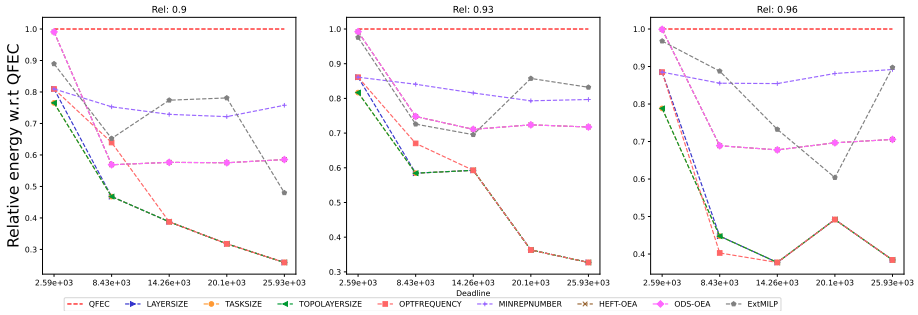

Figure 1900: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

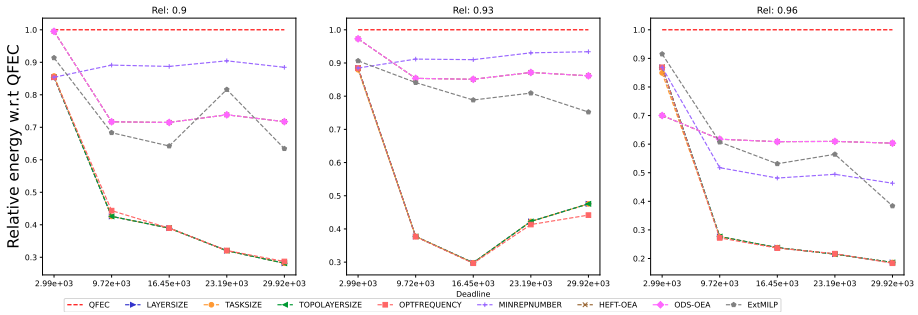

Figure 1901: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.5.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

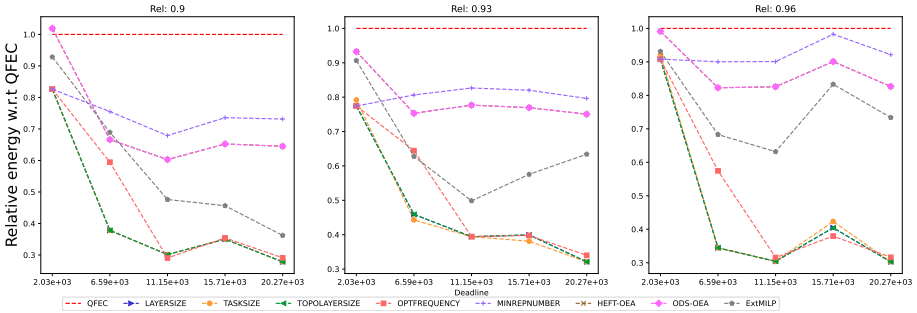

Figure 1902: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

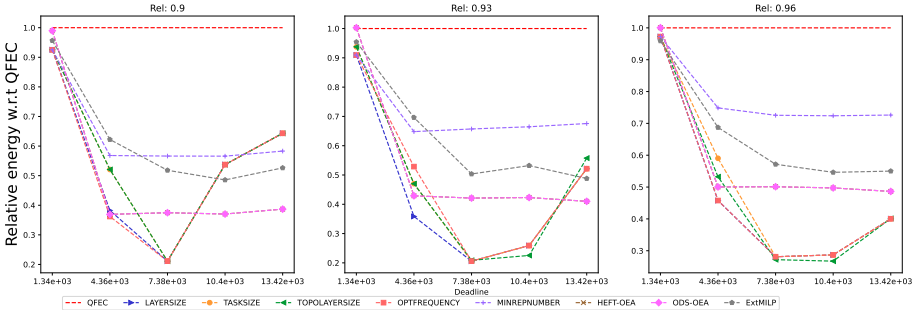

Figure 1903: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

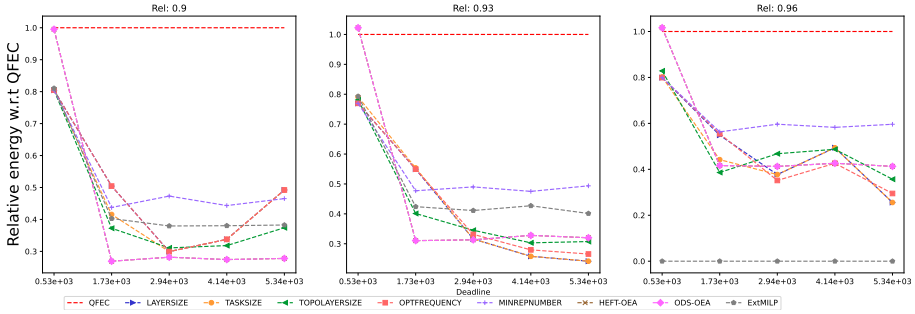

Figure 1904: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

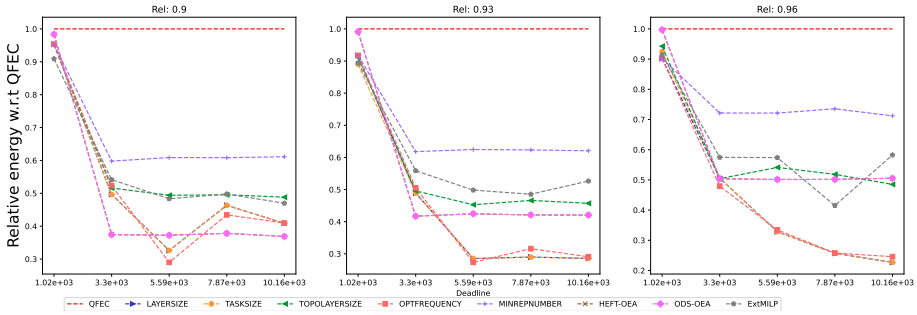

Figure 1905: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

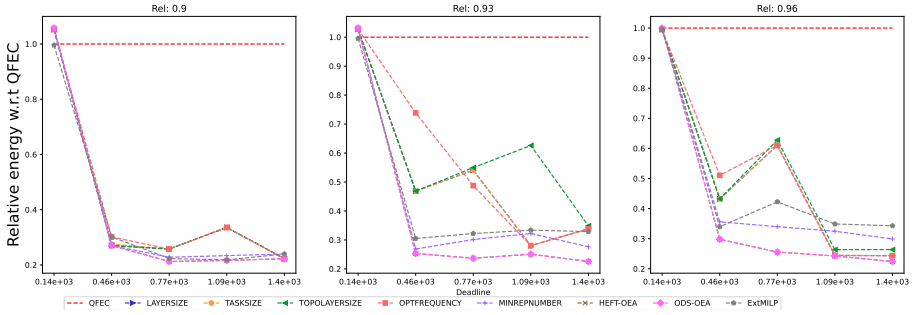

Figure 1906: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

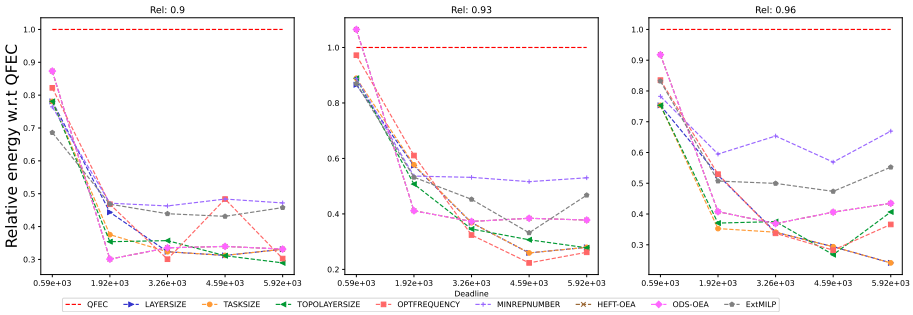

Figure 1907: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

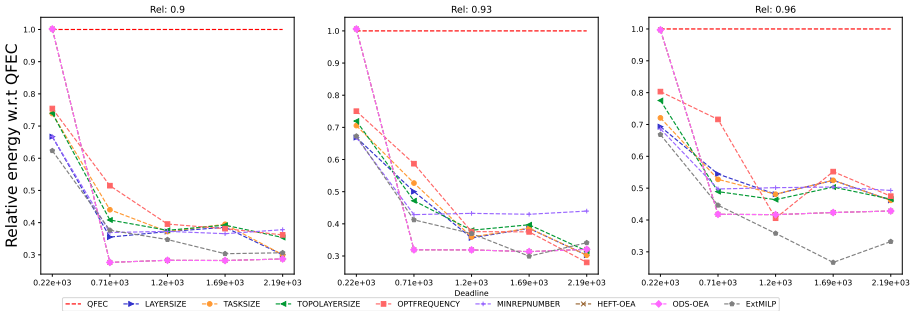

Figure 1908: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

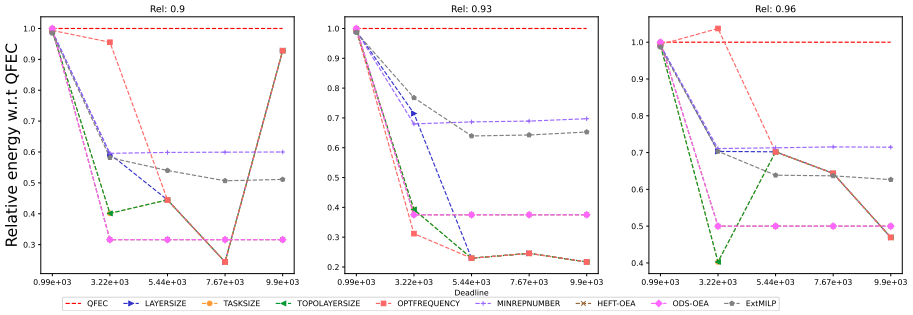

Figure 1909: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

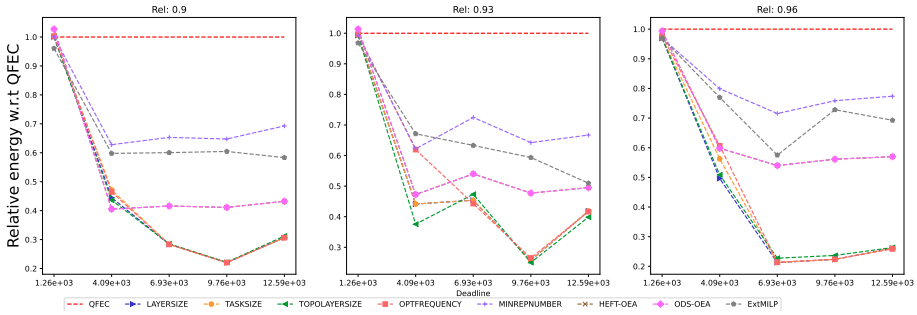

Figure 1910: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

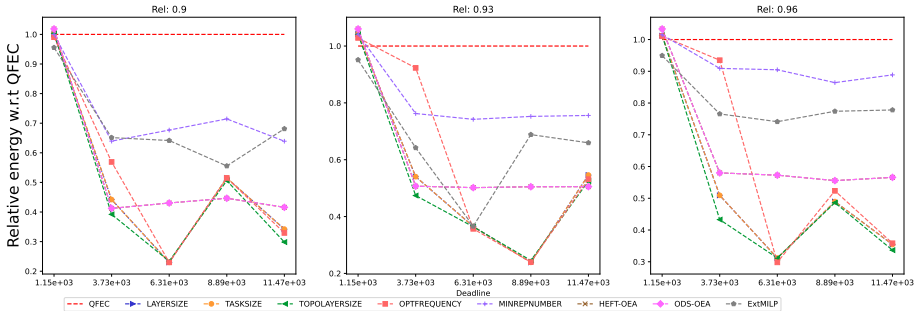

Figure 1911: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

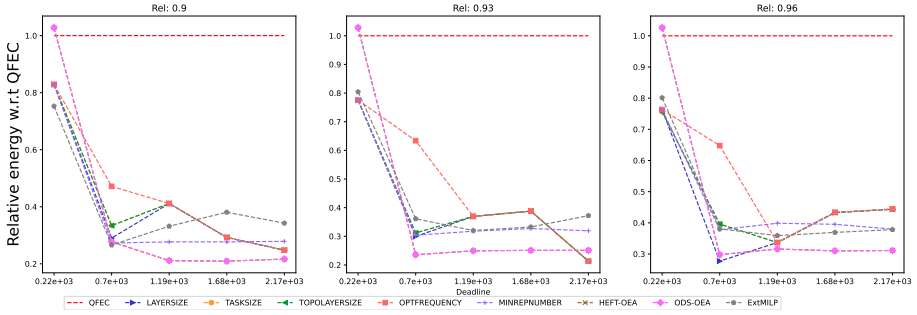

Figure 1912: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

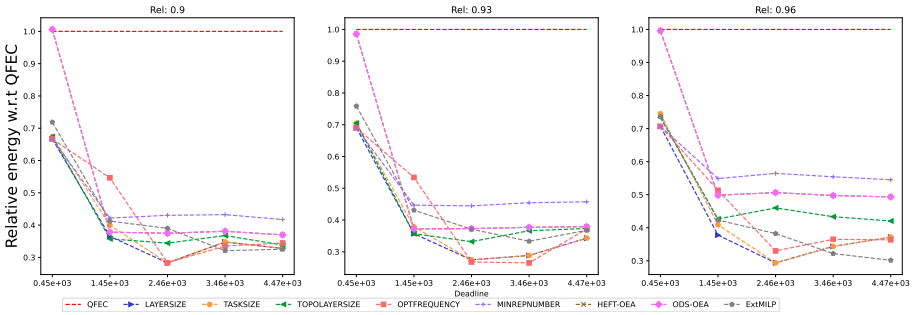

Figure 1913: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

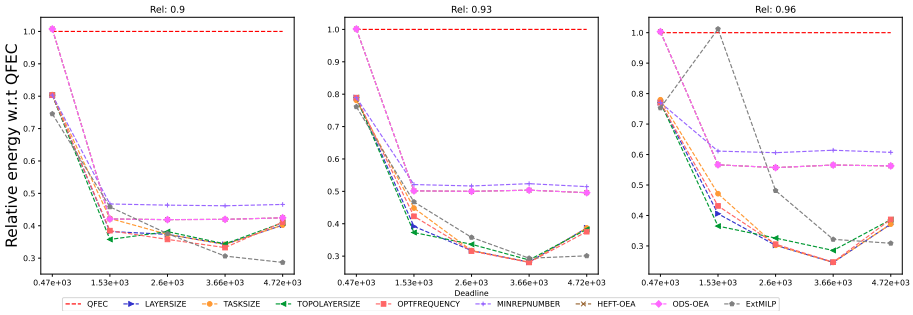

Figure 1914: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

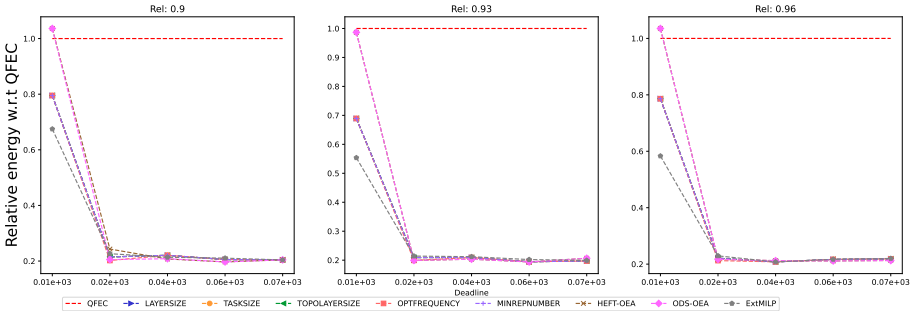

Figure 1915: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

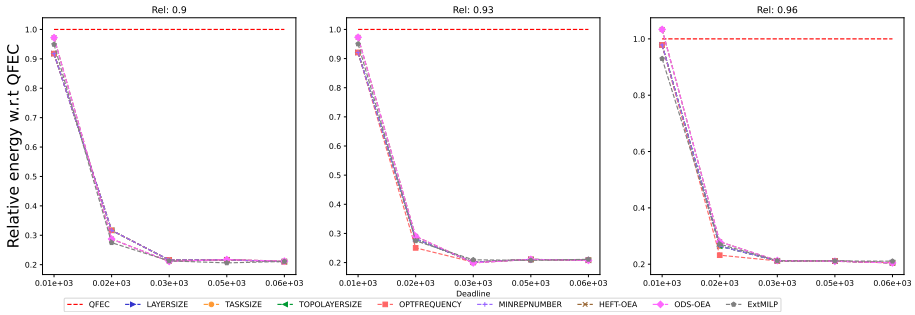

Figure 1916: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

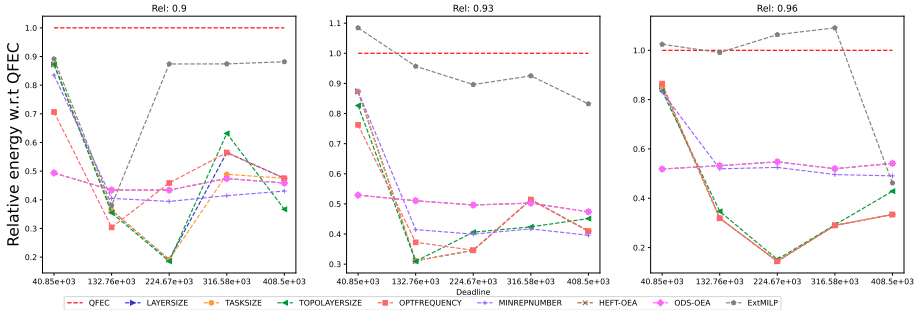

Figure 1917: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

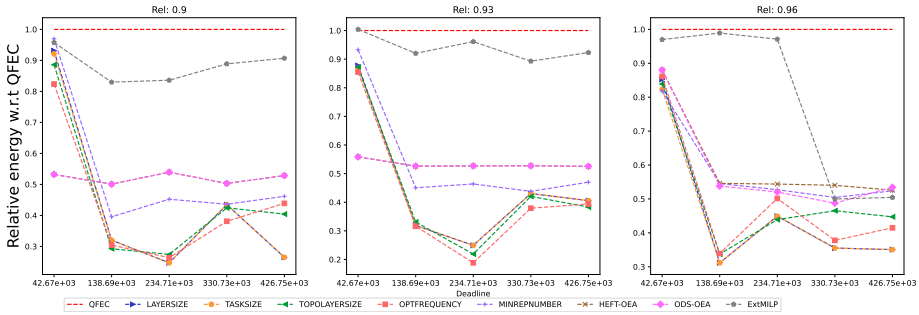

Figure 1918: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

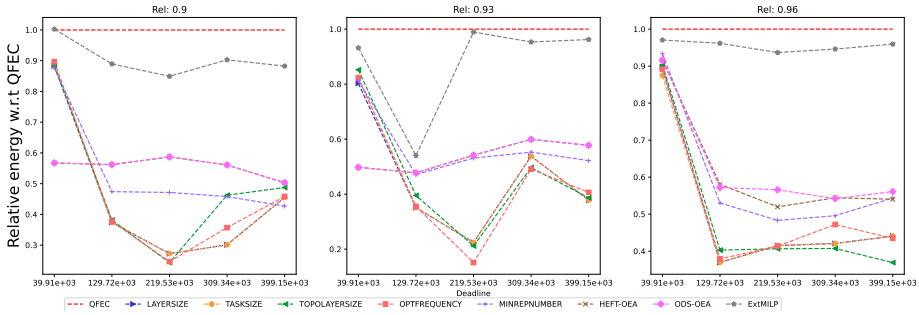

Figure 1919: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

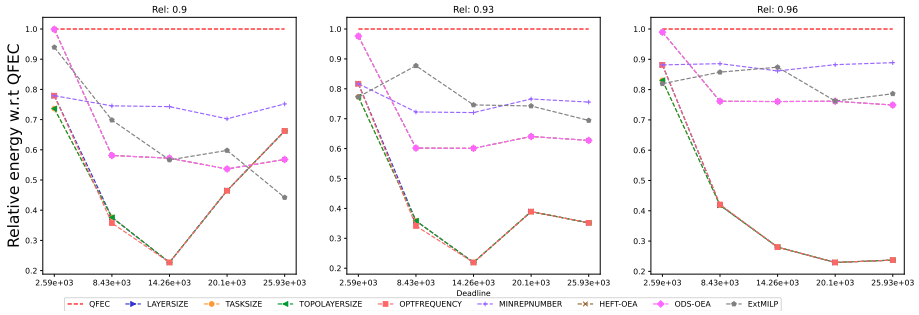

Figure 1920: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

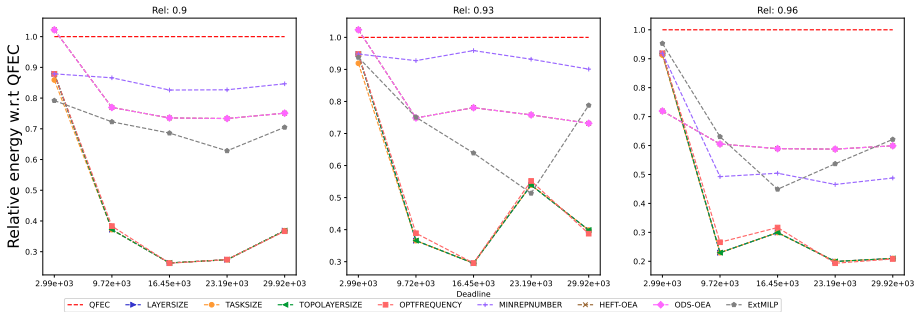

Figure 1921: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.6 $BC/WC = 0.6$

G.6.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

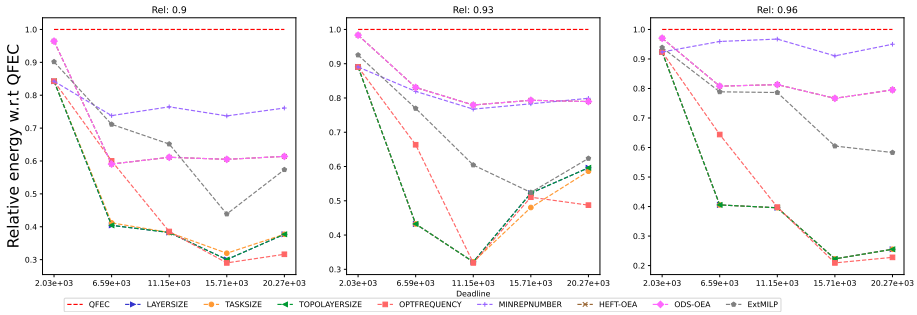

Figure 1922: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

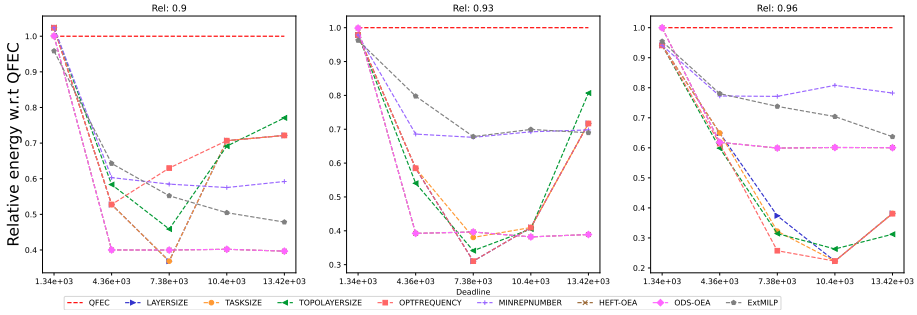

Figure 1923: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

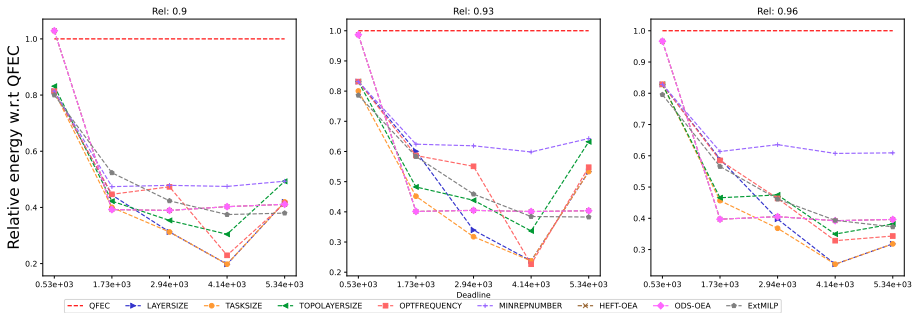

Figure 1924: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

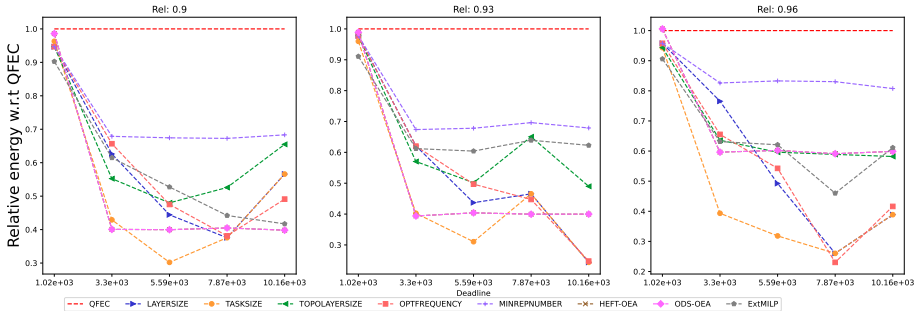

Figure 1925: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

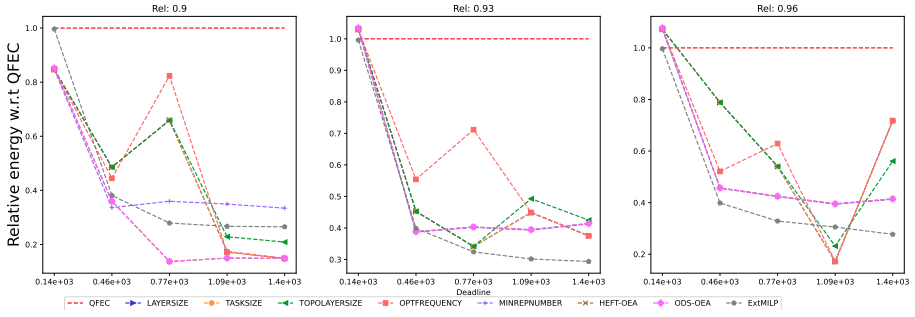

Figure 1926: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

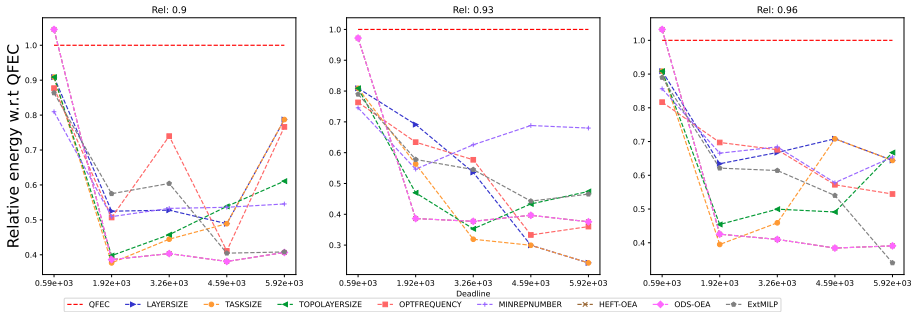

Figure 1927: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

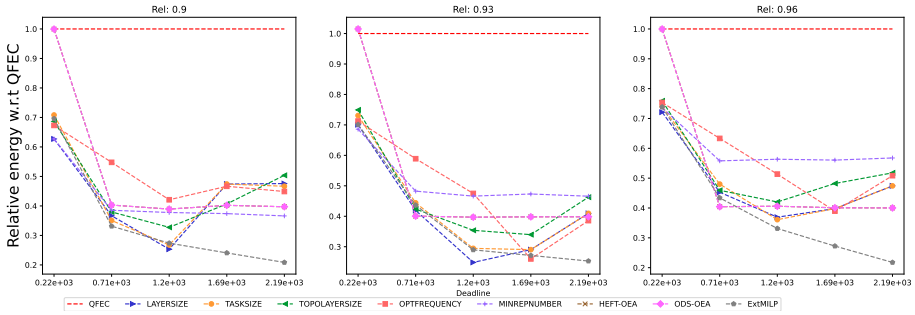

Figure 1928: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

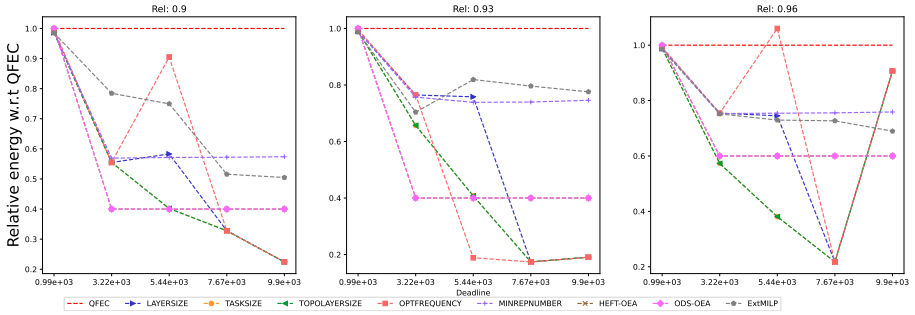

Figure 1929: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

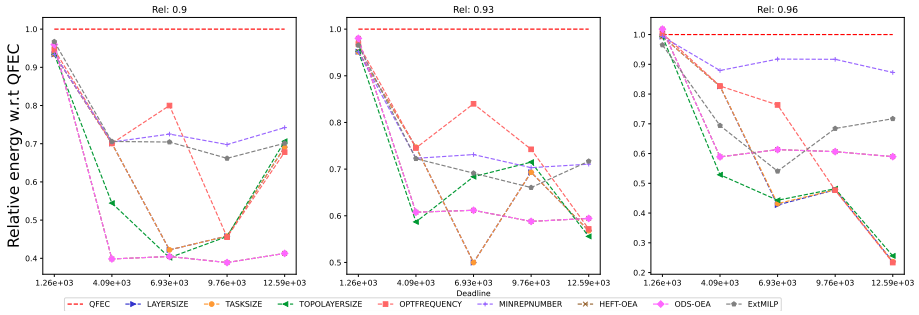

Figure 1930: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

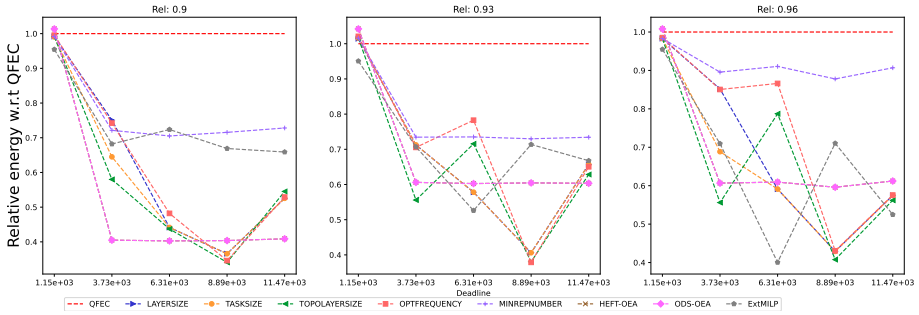

Figure 1931: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

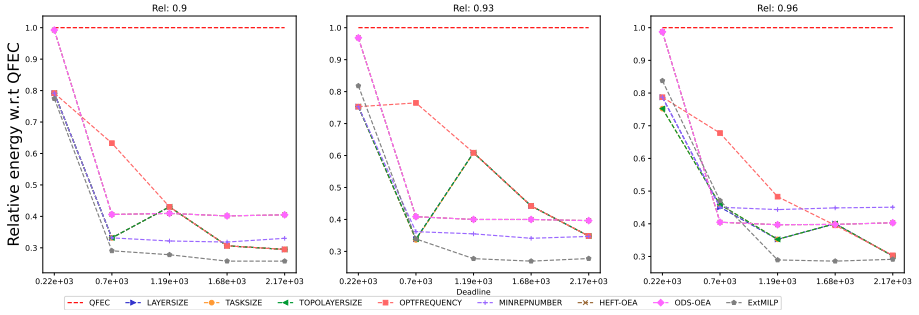

Figure 1932: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

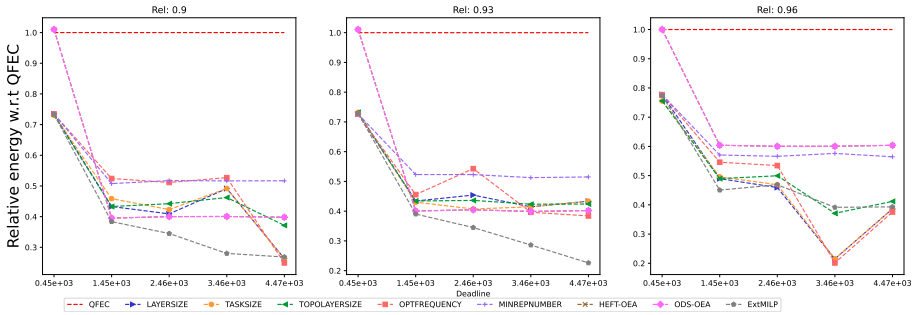

Figure 1933: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

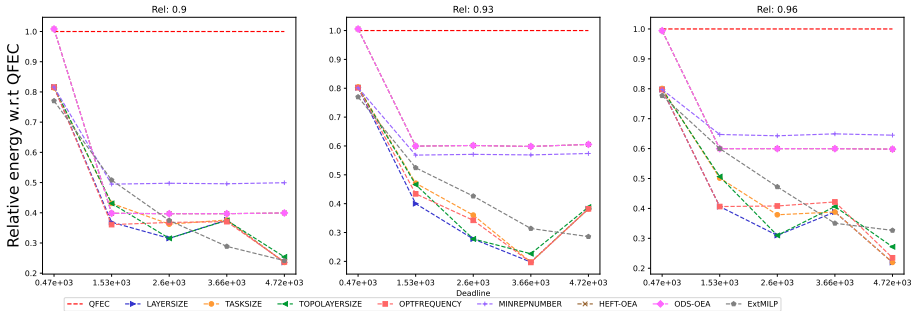

Figure 1934: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

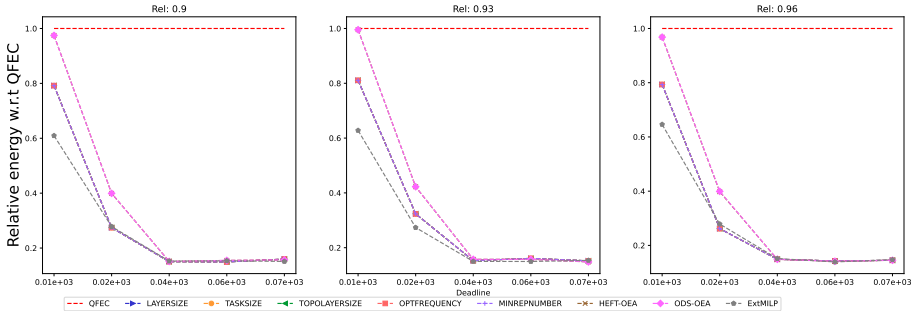

Figure 1935: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

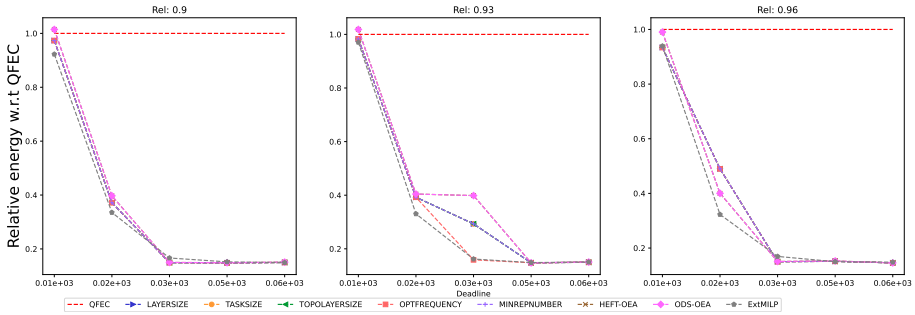

Figure 1936: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

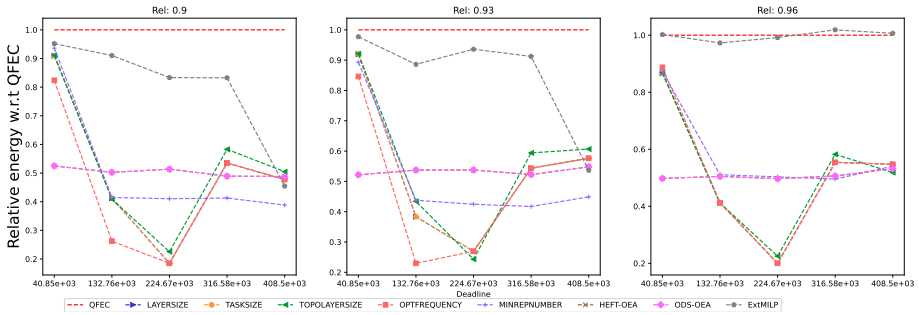

Figure 1937: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

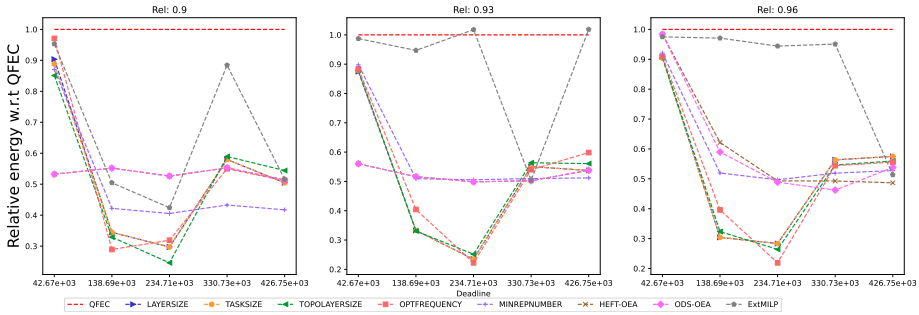

Figure 1938: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

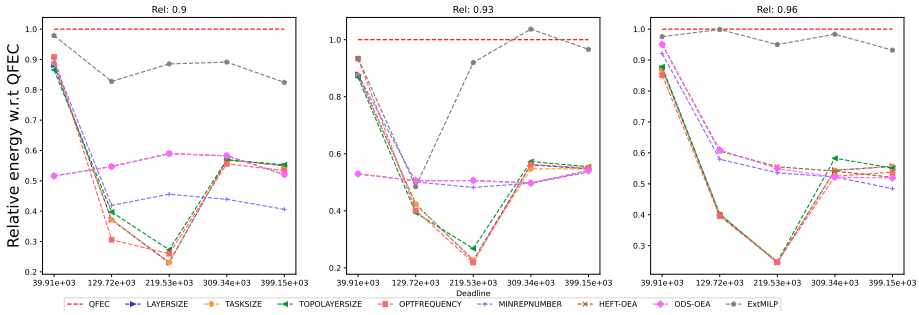

Figure 1939: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

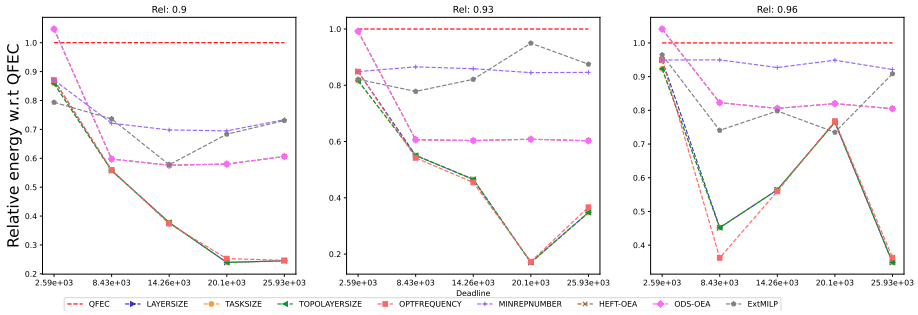

Figure 1940: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

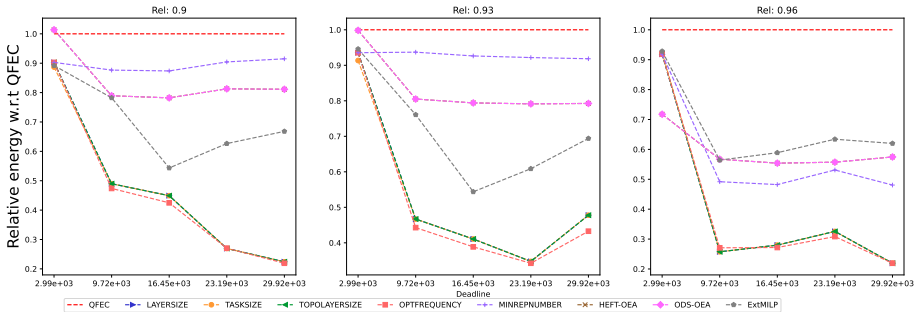

Figure 1941: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.6.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

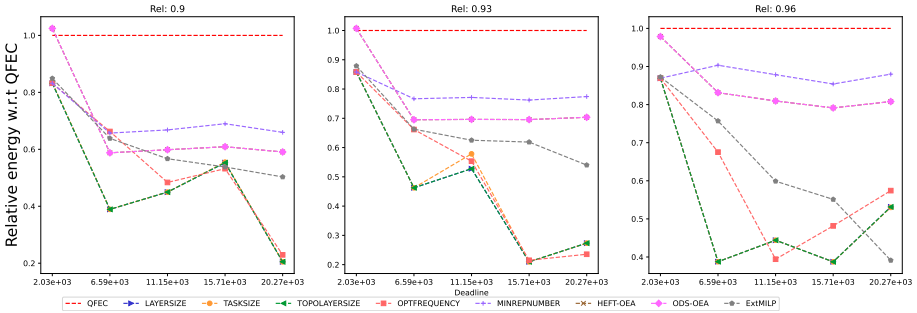

Figure 1942: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

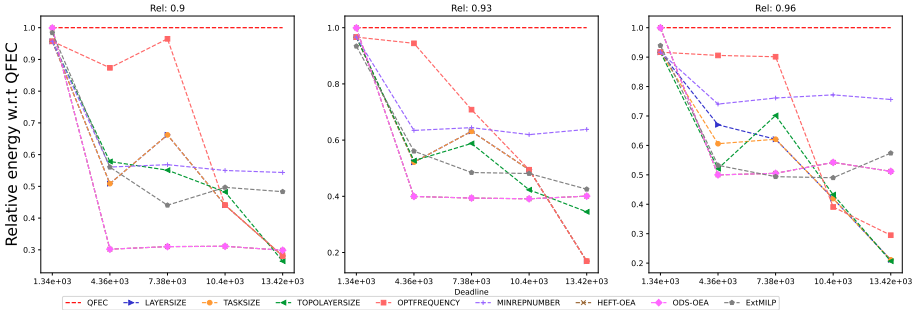

Figure 1943: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

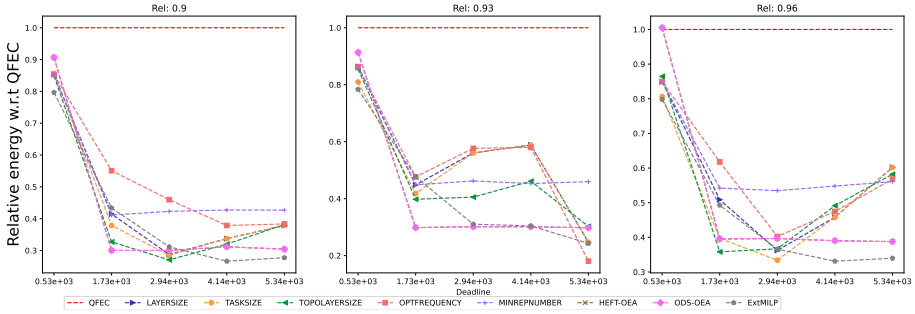

Figure 1944: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

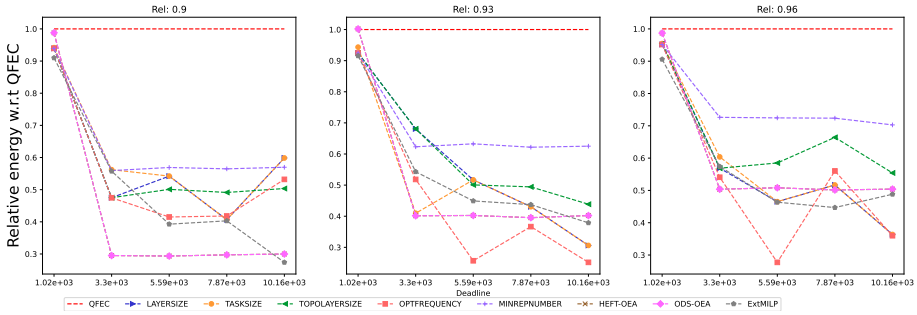

Figure 1945: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

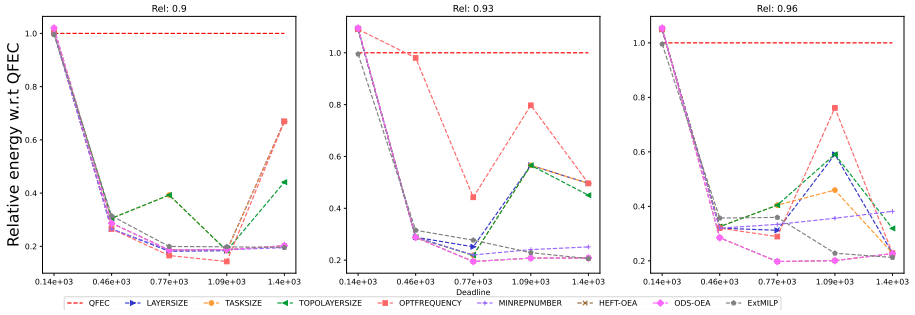

Figure 1946: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

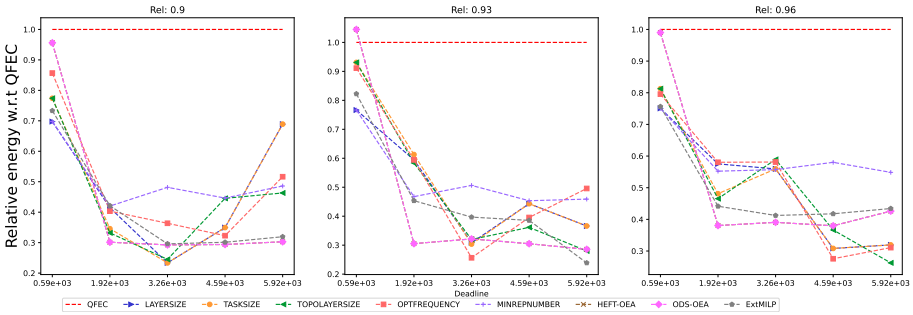

Figure 1947: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

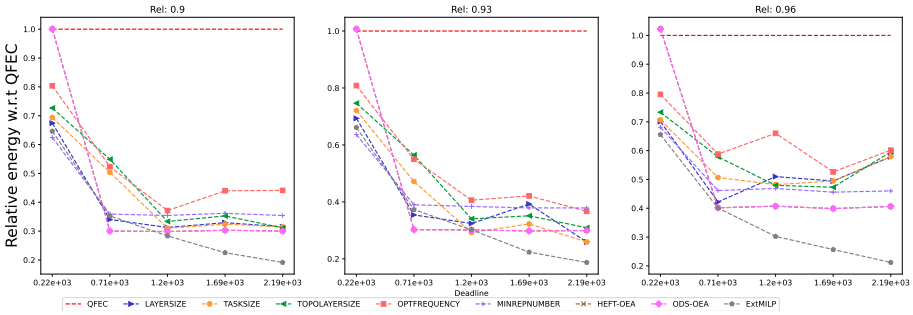

Figure 1948: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

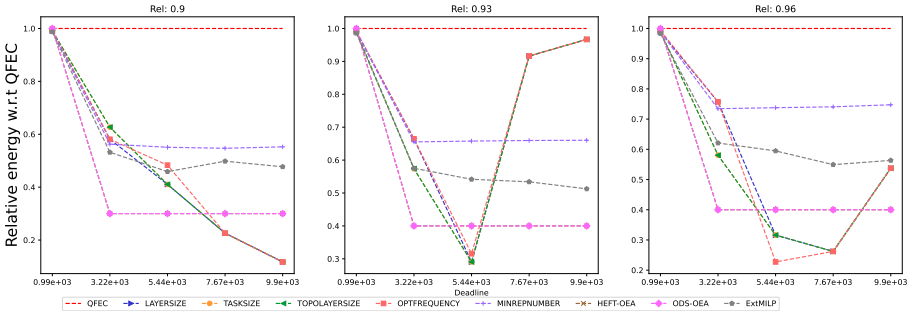

Figure 1949: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

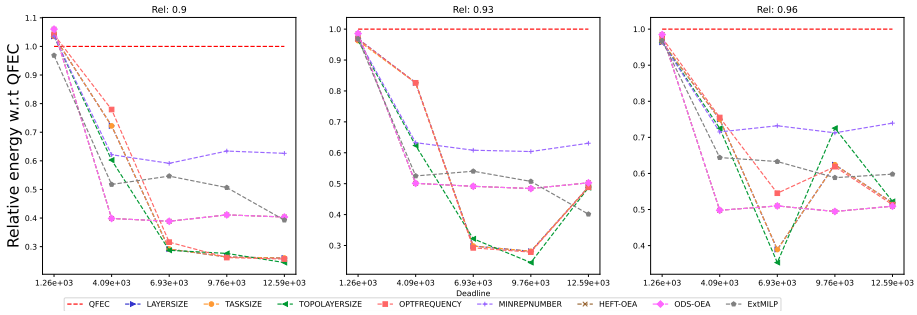

Figure 1950: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

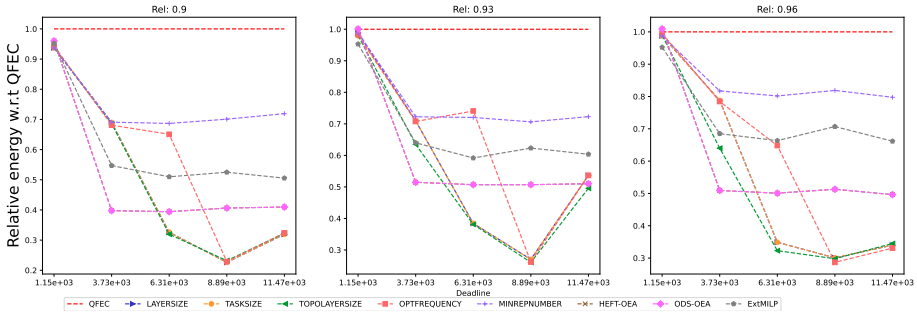

Figure 1951: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

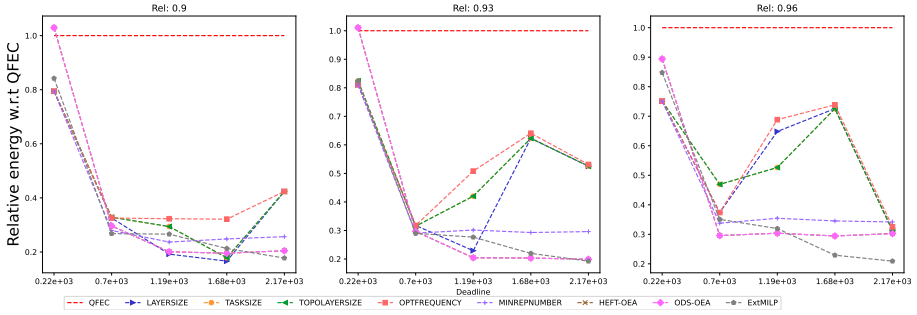

Figure 1952: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

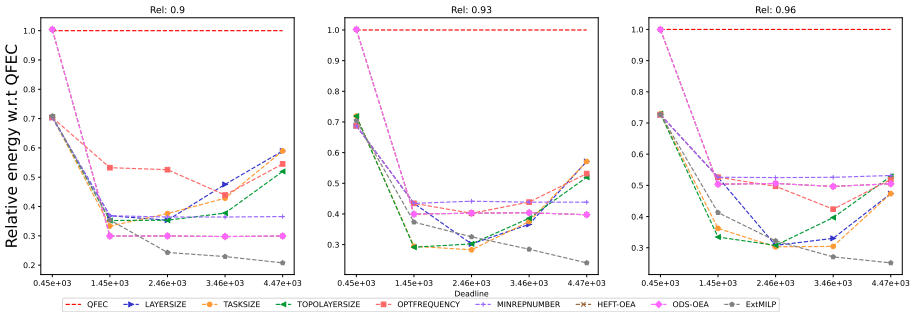

Figure 1953: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

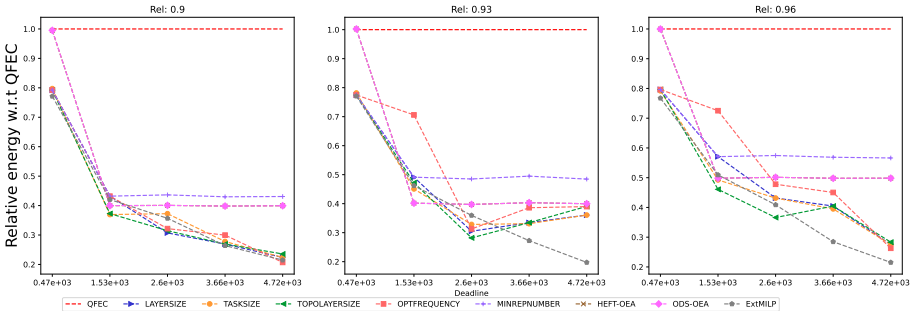

Figure 1954: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

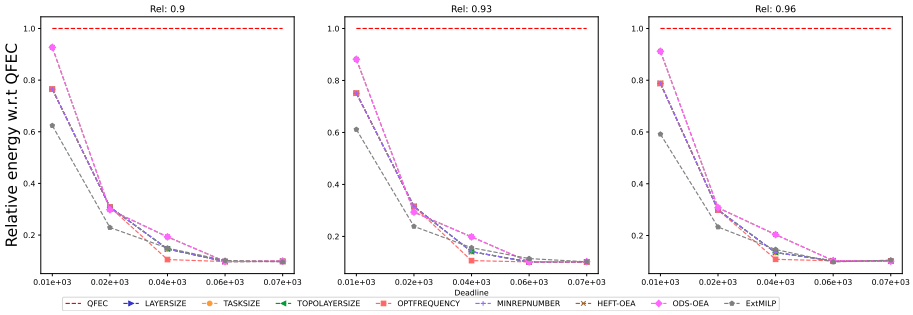

Figure 1955: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

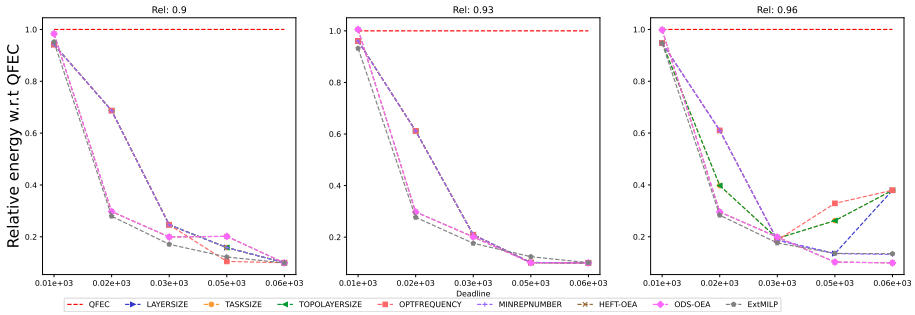

Figure 1956: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

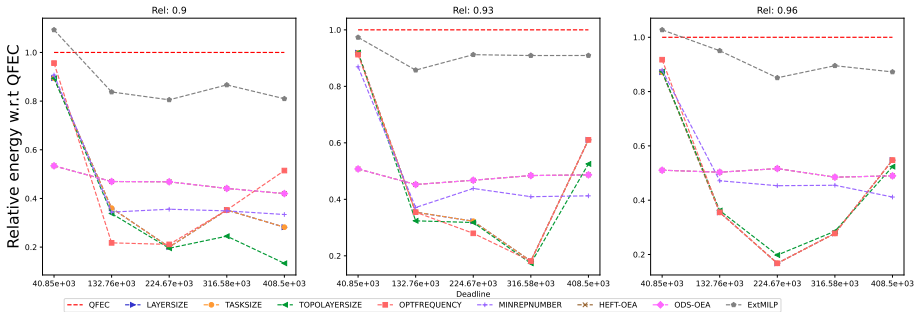

Figure 1957: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

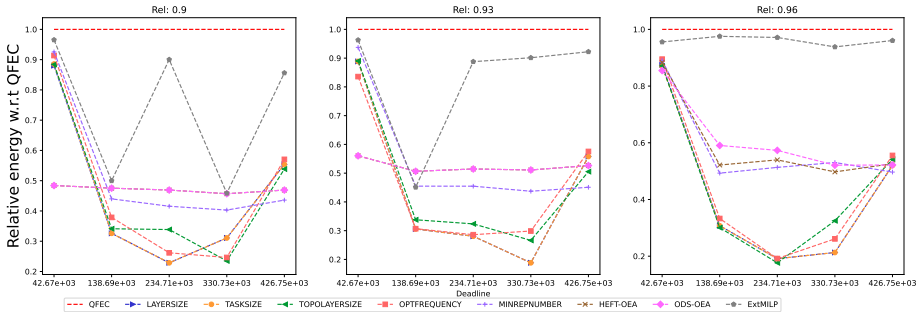

Figure 1958: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

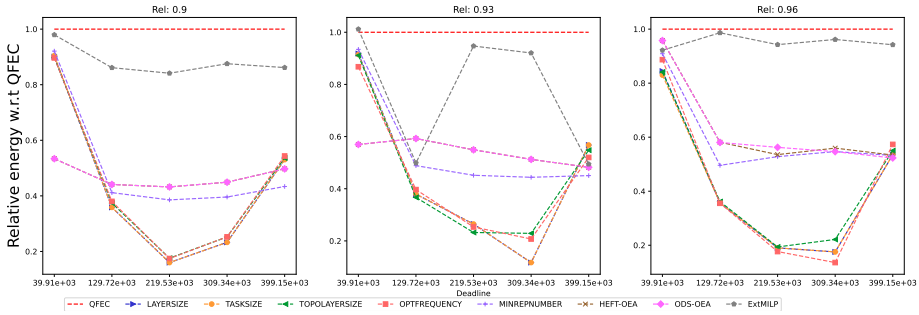

Figure 1959: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

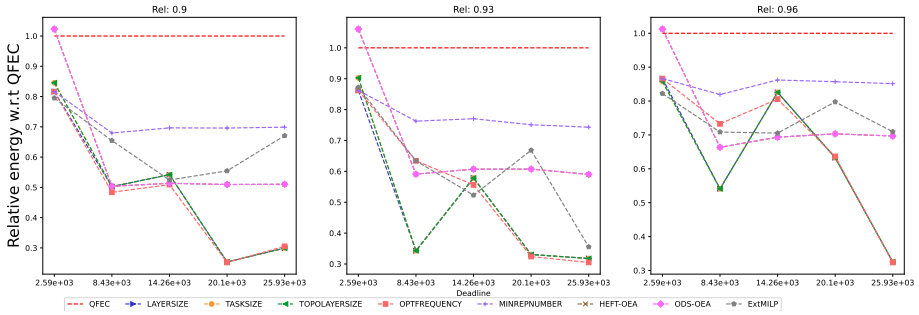

Figure 1960: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

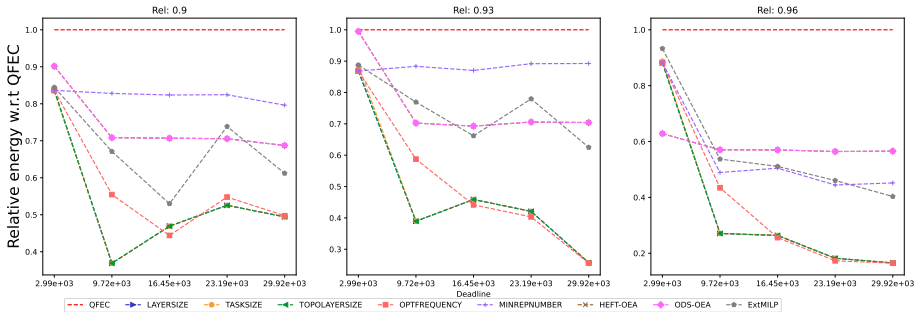

Figure 1961: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.6.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

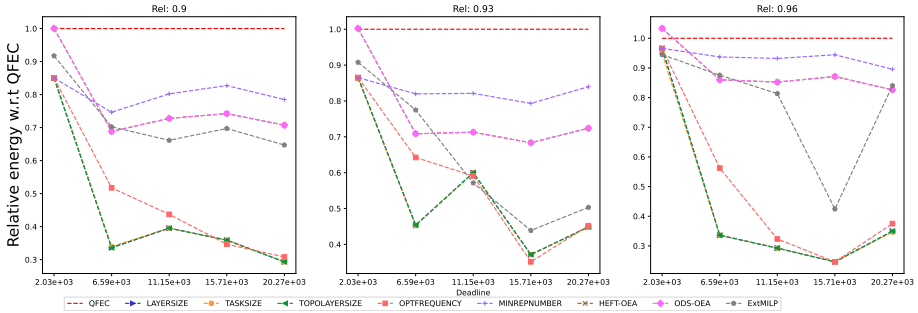

Figure 1962: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

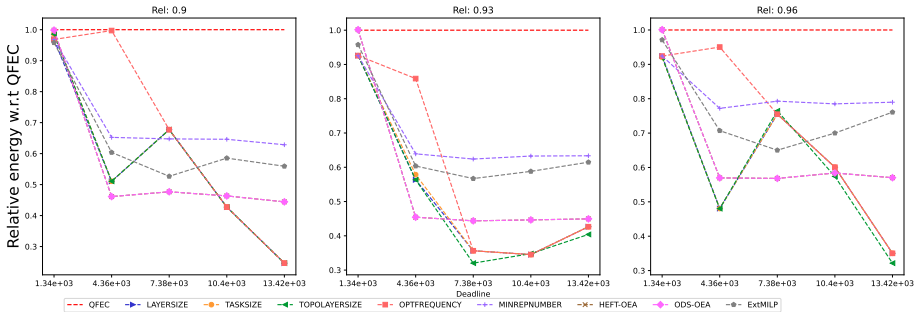

Figure 1963: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

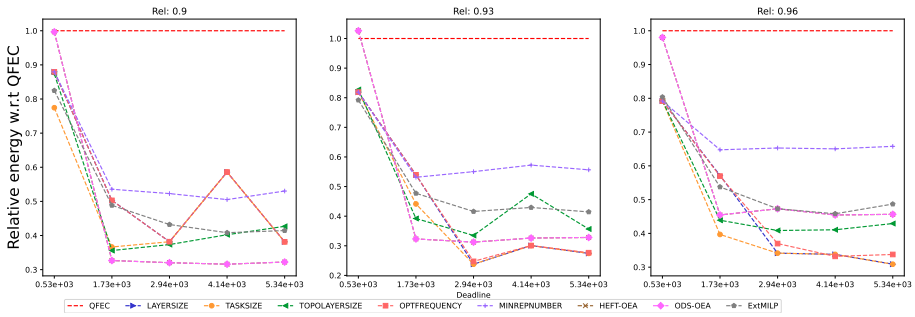

Figure 1964: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

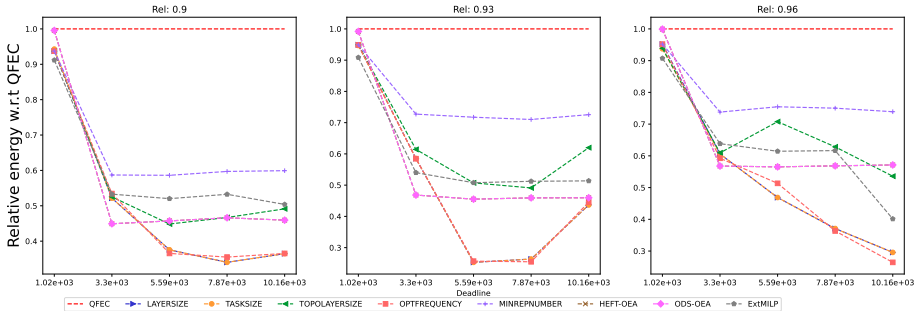

Figure 1965: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

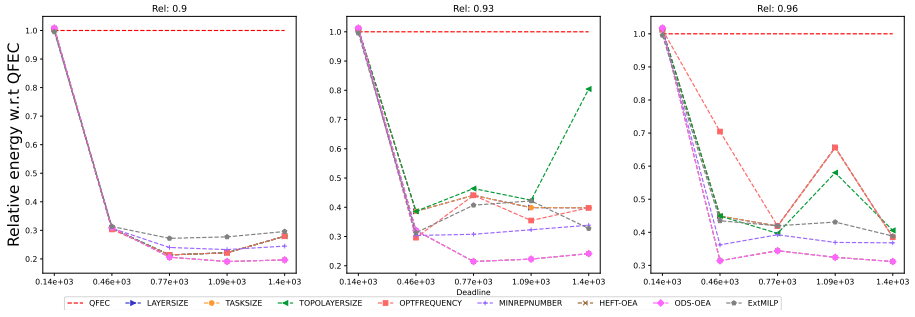

Figure 1966: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

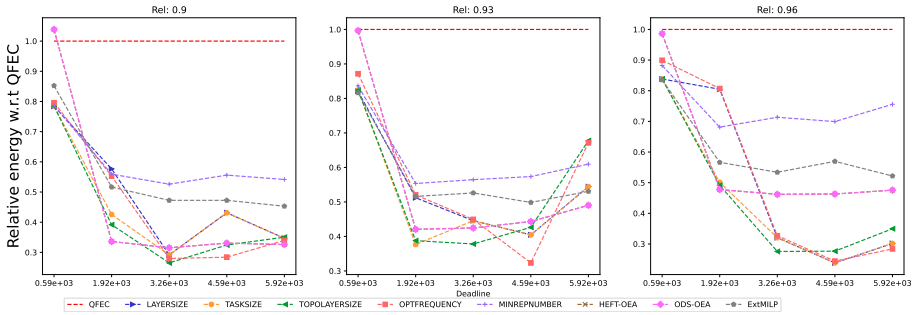

Figure 1967: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

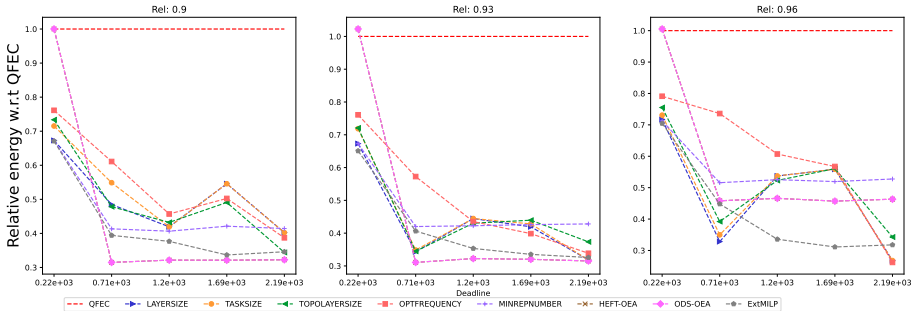

Figure 1968: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

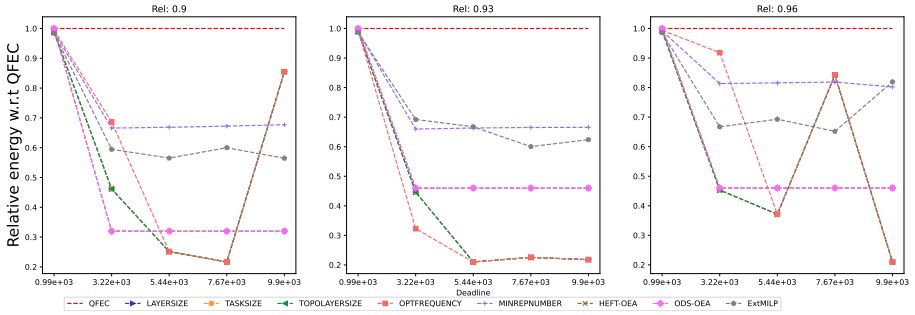

Figure 1969: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

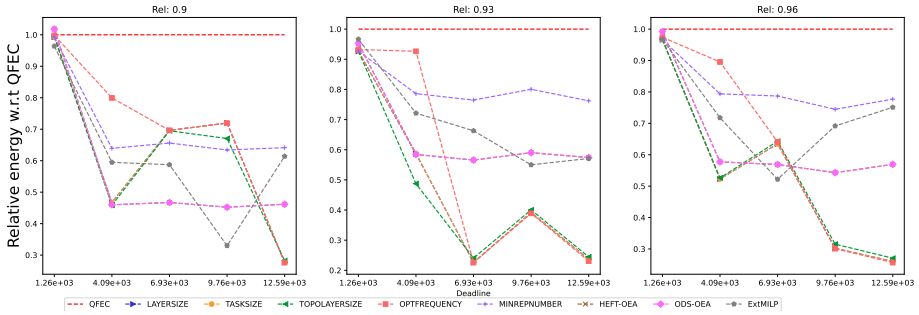

Figure 1970: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

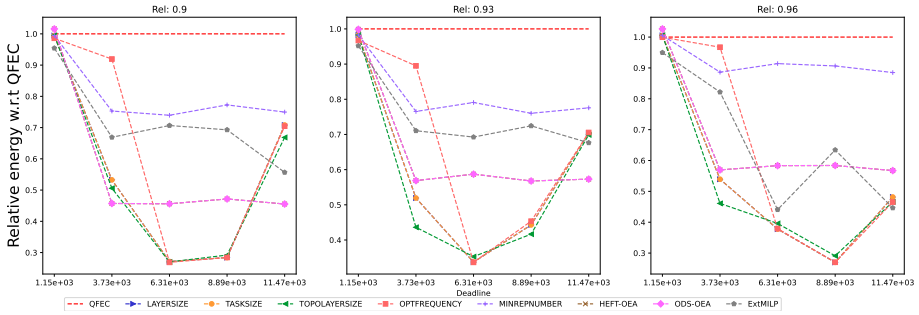

Figure 1971: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

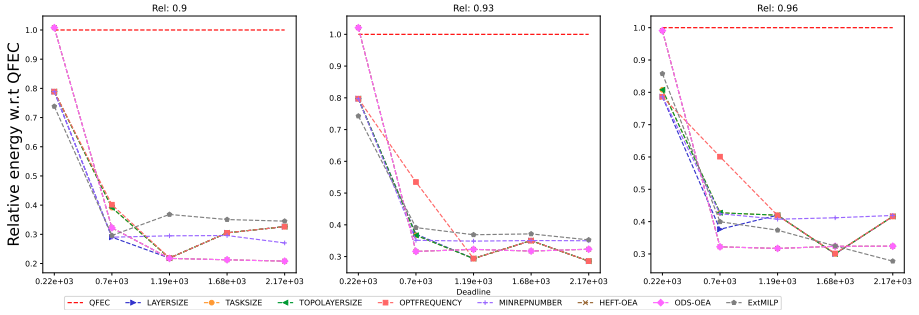

Figure 1972: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

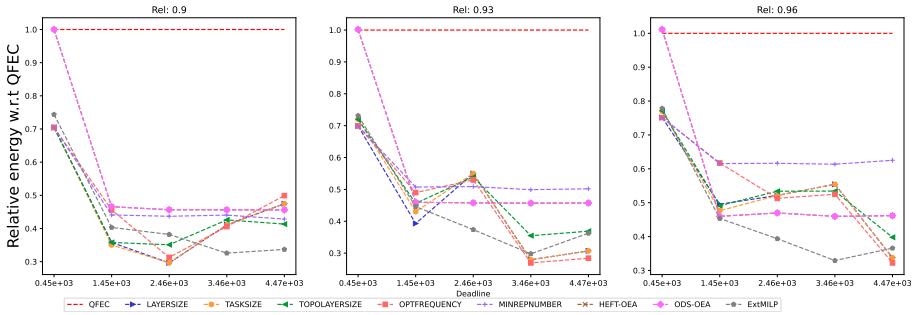

Figure 1973: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

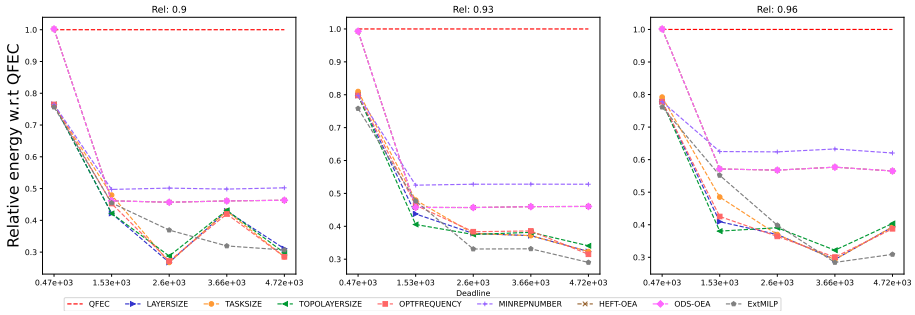

Figure 1974: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

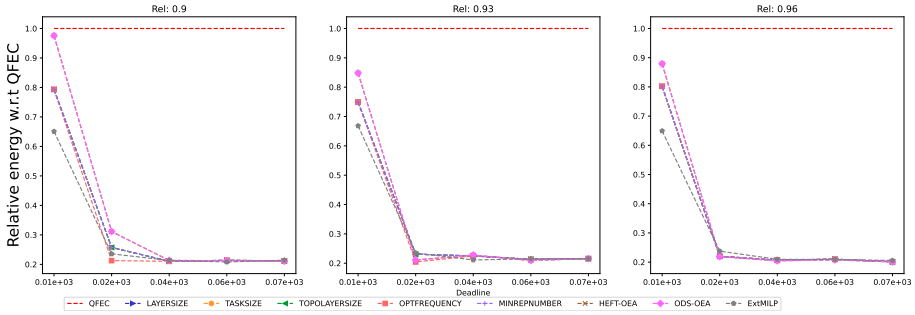

Figure 1975: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

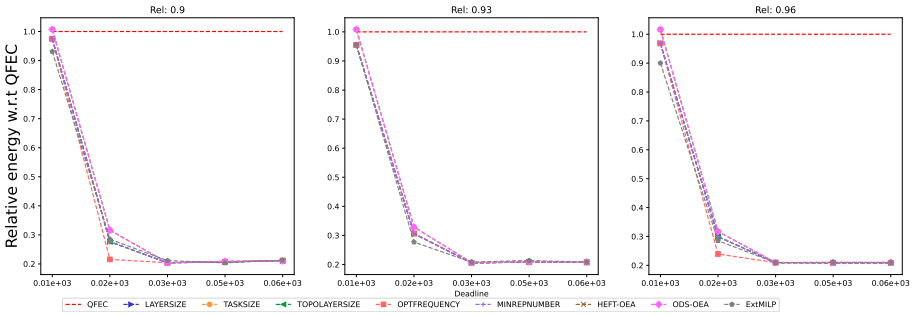

Figure 1976: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

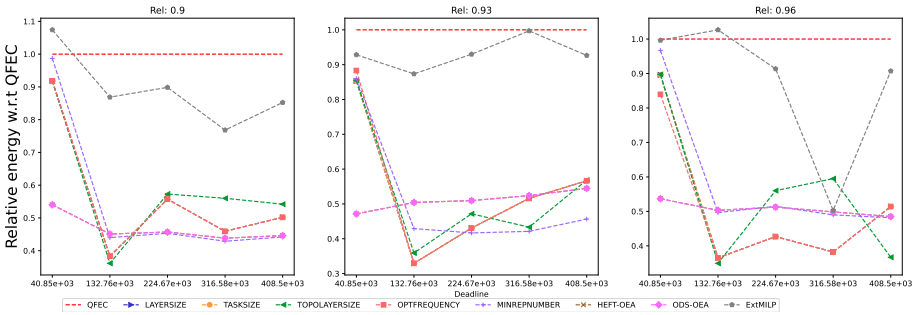

Figure 1977: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

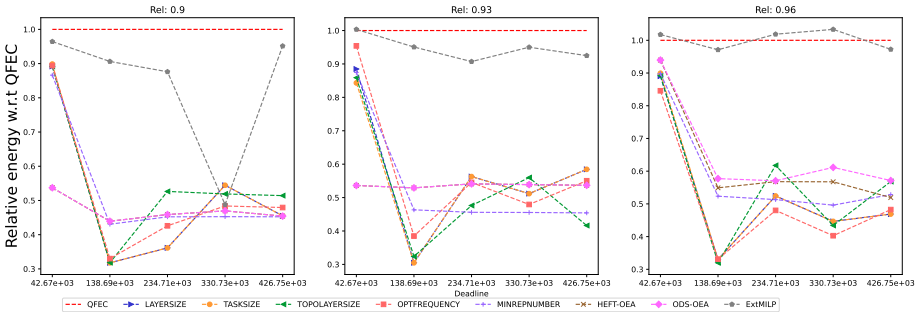

Figure 1978: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

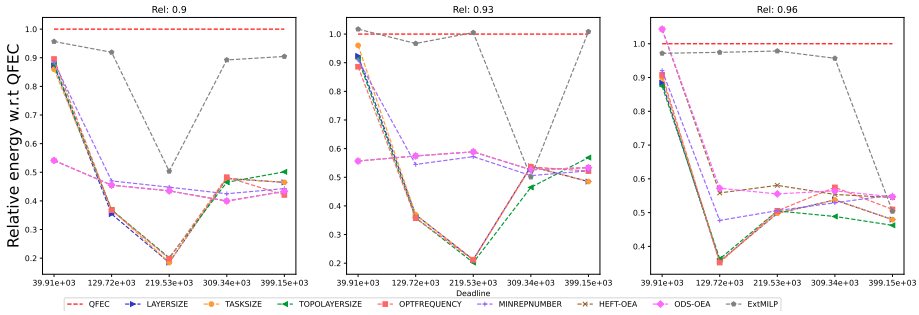

Figure 1979: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

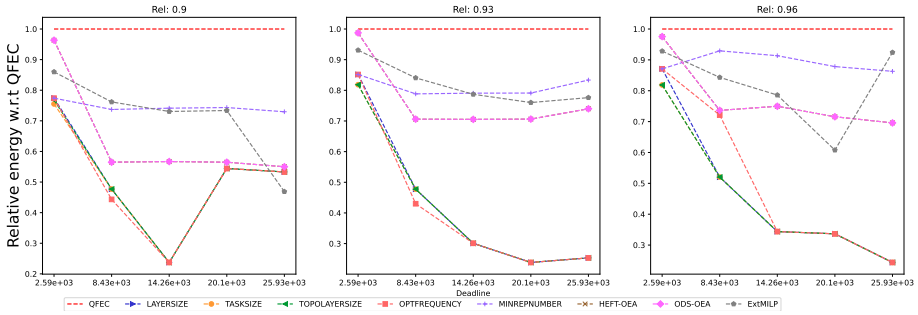

Figure 1980: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

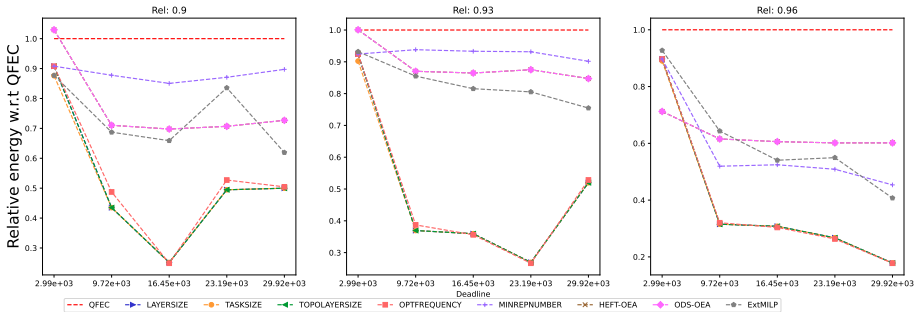

Figure 1981: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.6.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

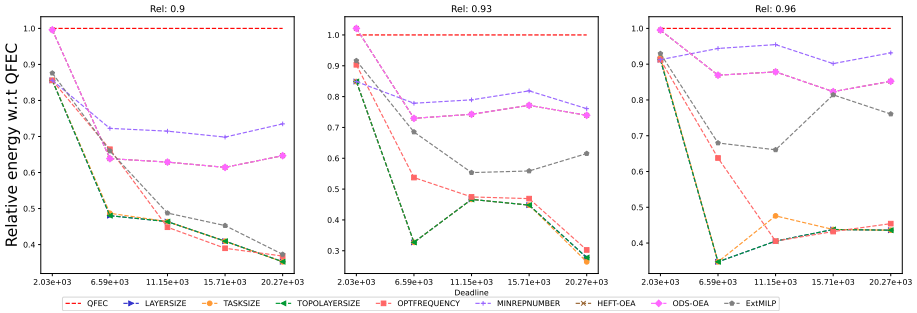

Figure 1982: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

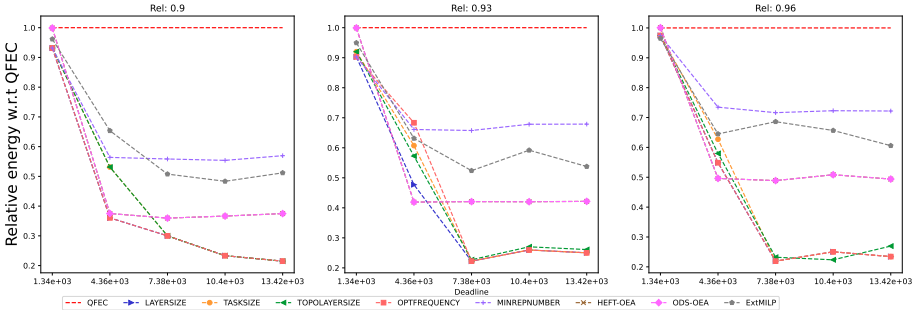

Figure 1983: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

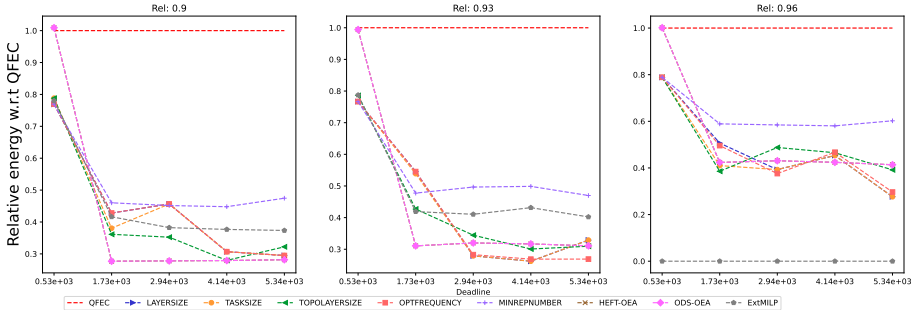

Figure 1984: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

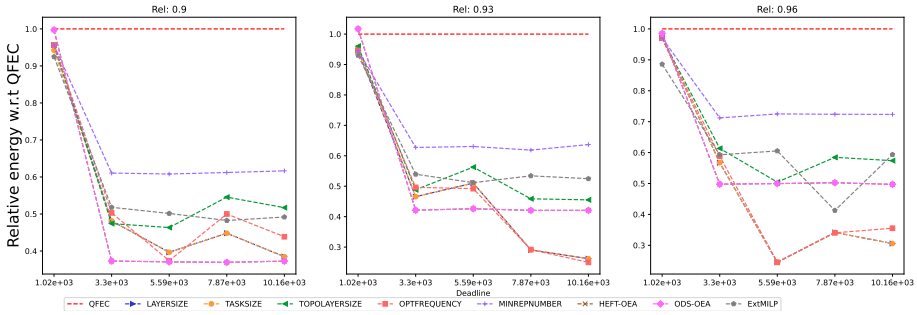

Figure 1985: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

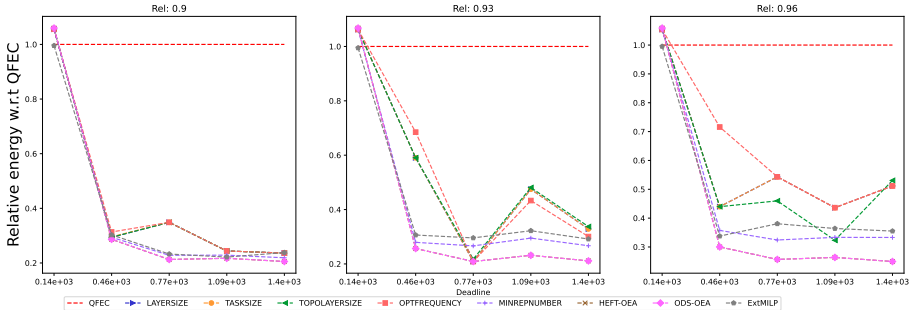

Figure 1986: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

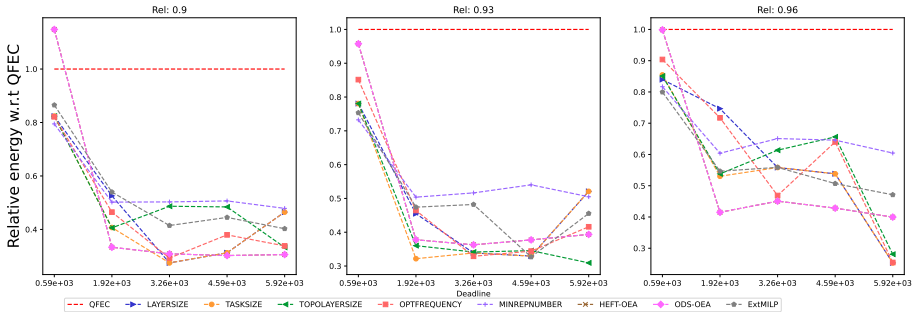

Figure 1987: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

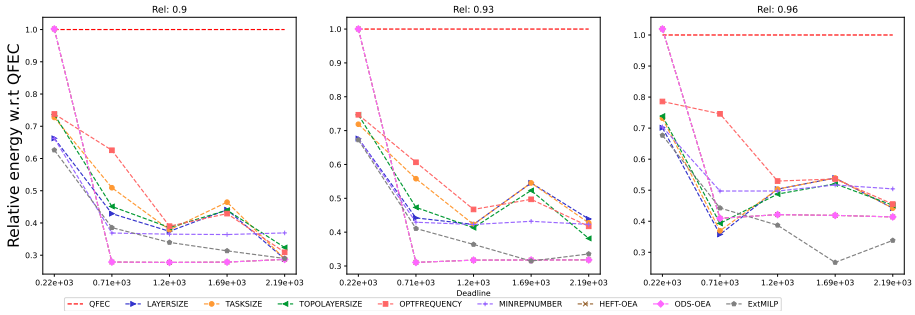

Figure 1988: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

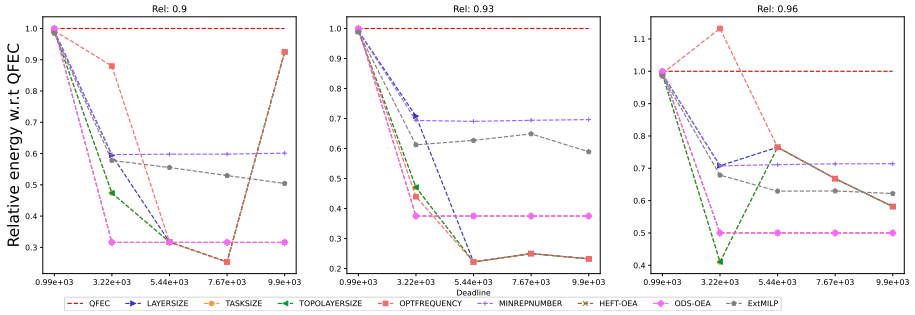

Figure 1989: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

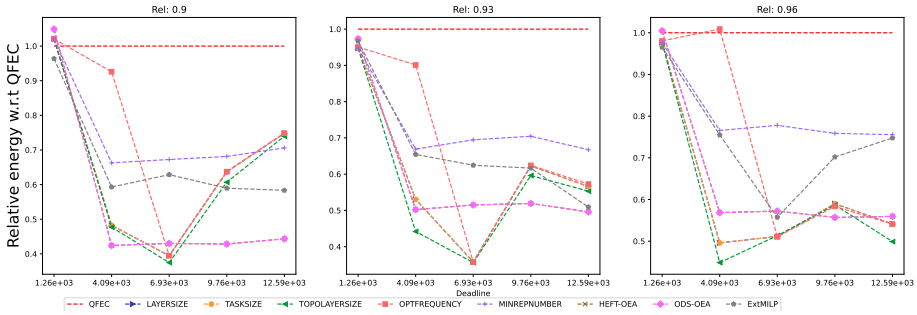

Figure 1990: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

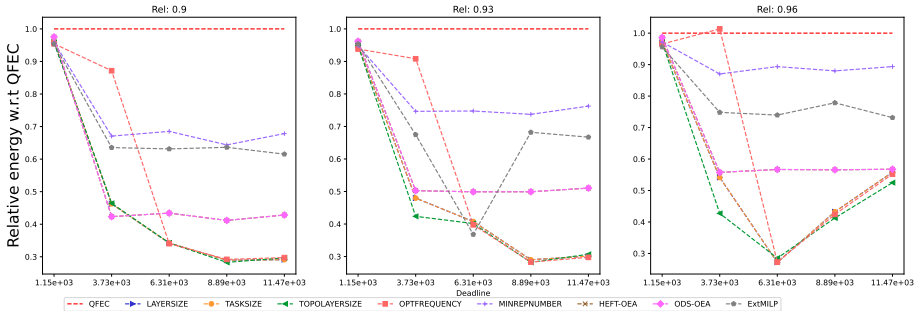

Figure 1991: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

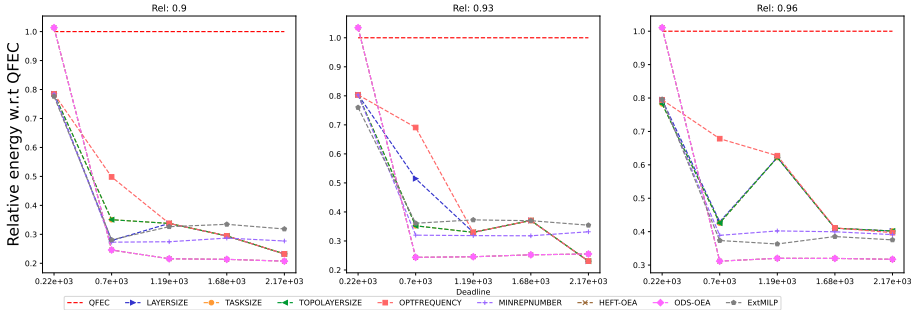

Figure 1992: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

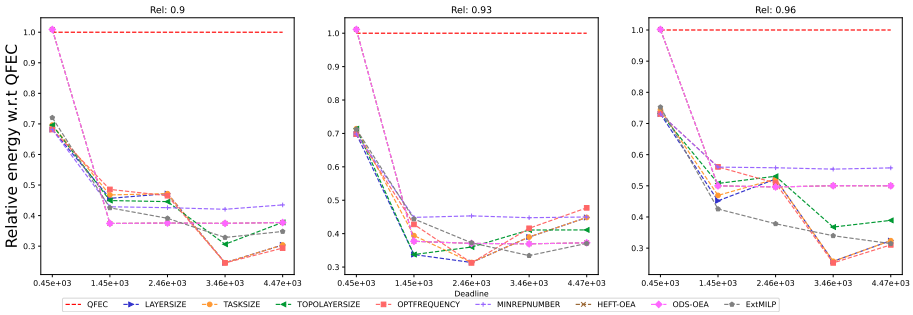

Figure 1993: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

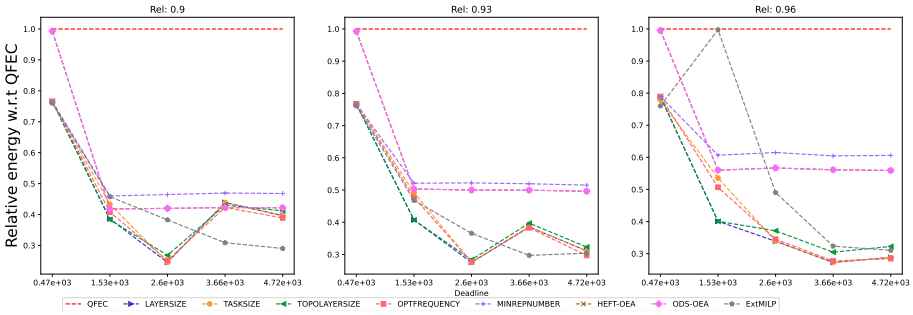

Figure 1994: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

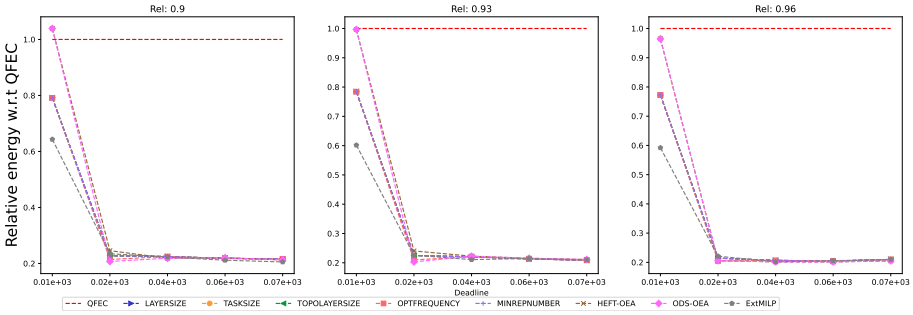

Figure 1995: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

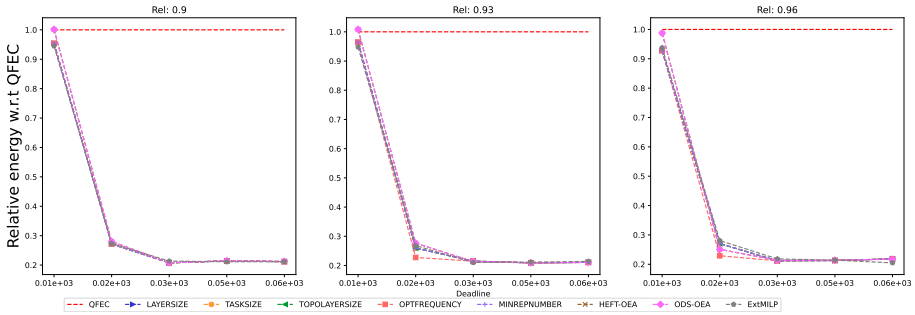

Figure 1996: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

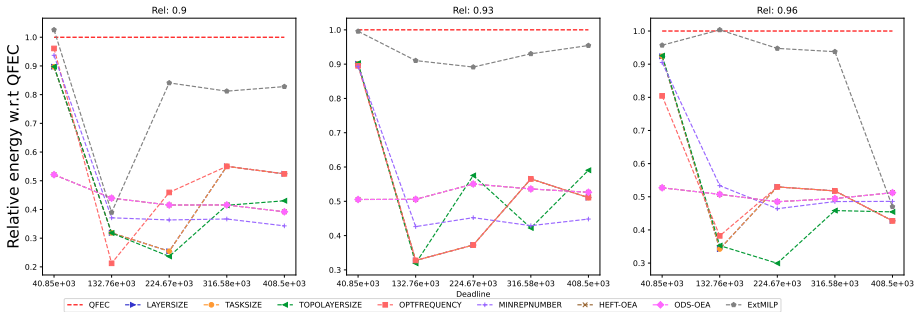

Figure 1997: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

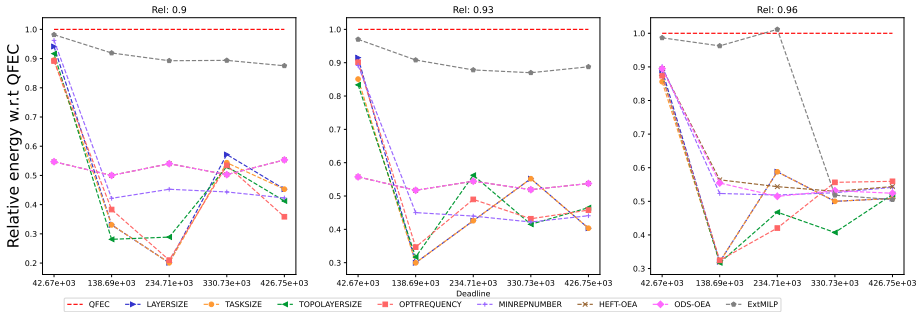

Figure 1998: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

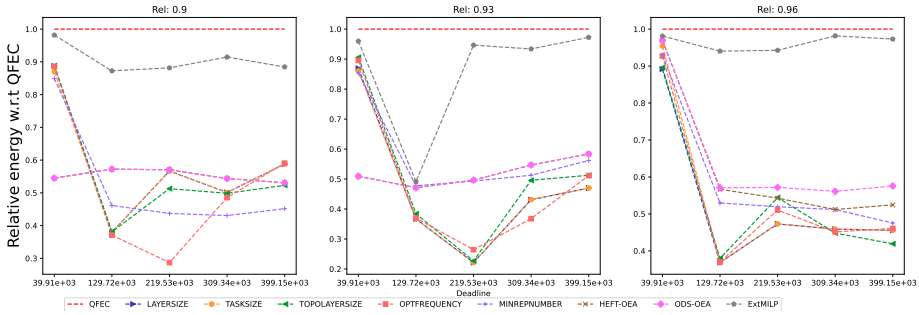

Figure 1999: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

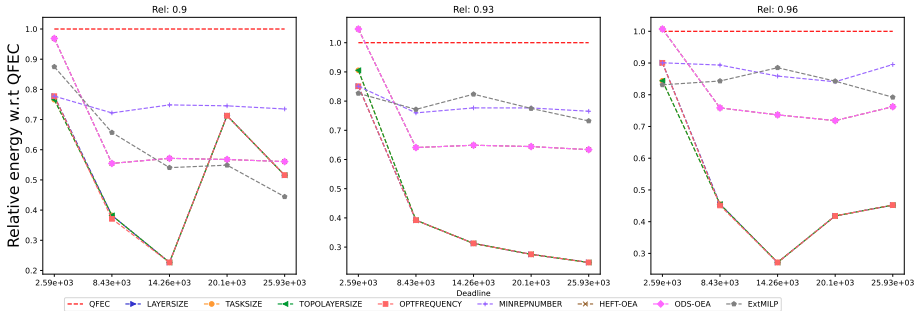

Figure 2000: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

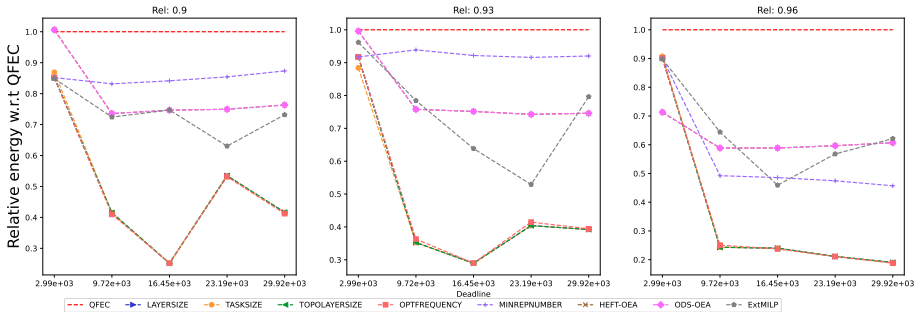

Figure 2001: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.7 $BC/WC = 0.7$

G.7.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

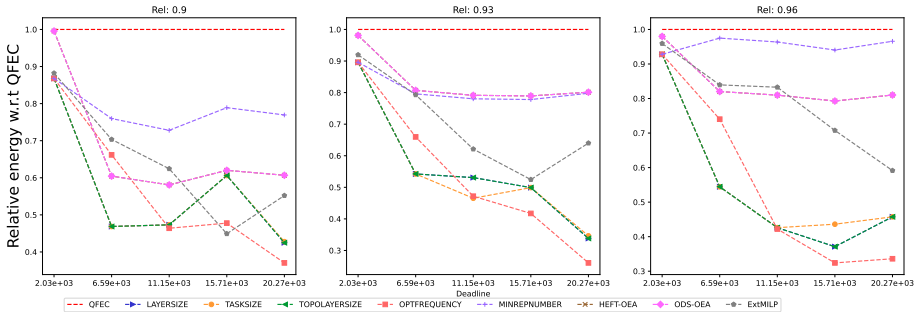

Figure 2002: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

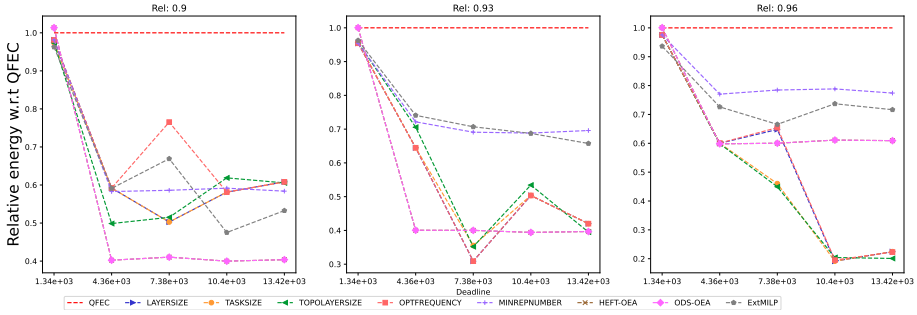

Figure 2003: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

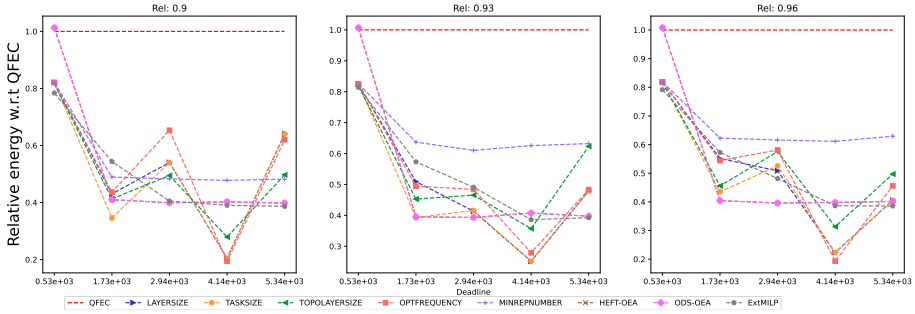

Figure 2004: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

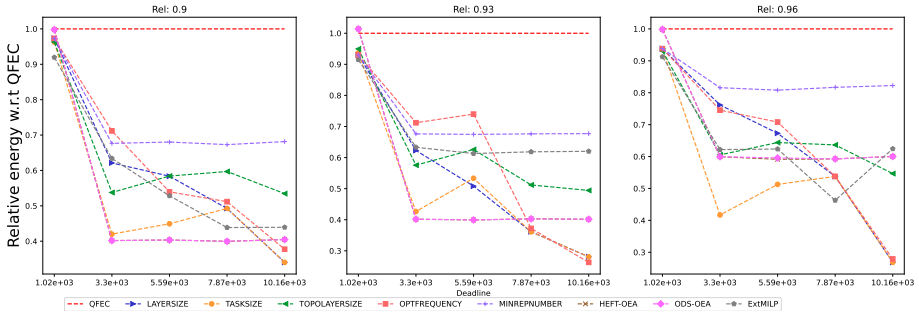

Figure 2005: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

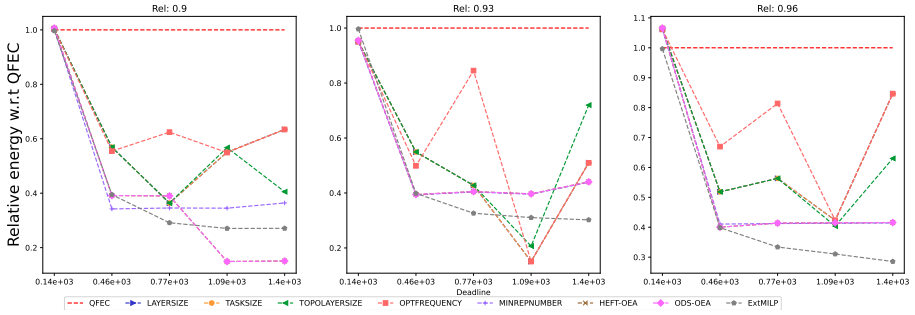

Figure 2006: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

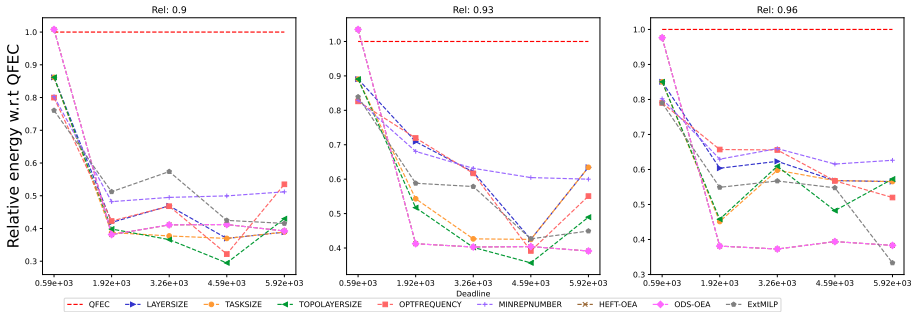

Figure 2007: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

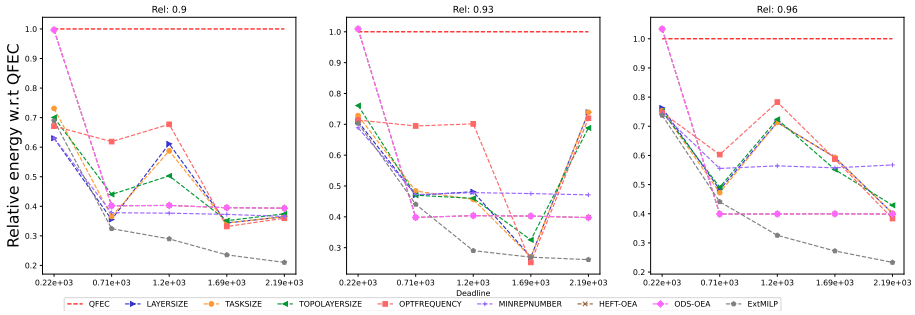

Figure 2008: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

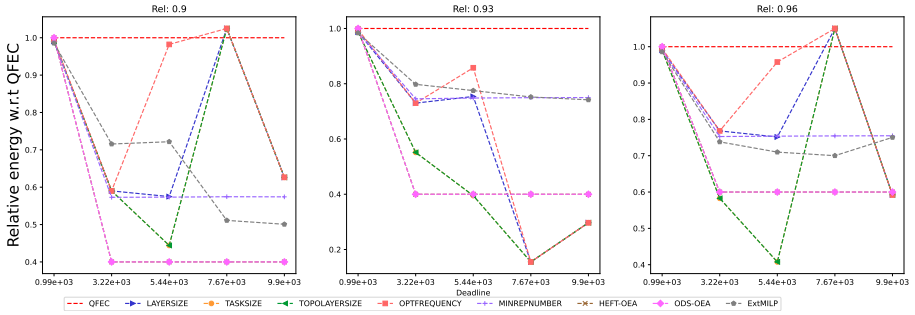

Figure 2009: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

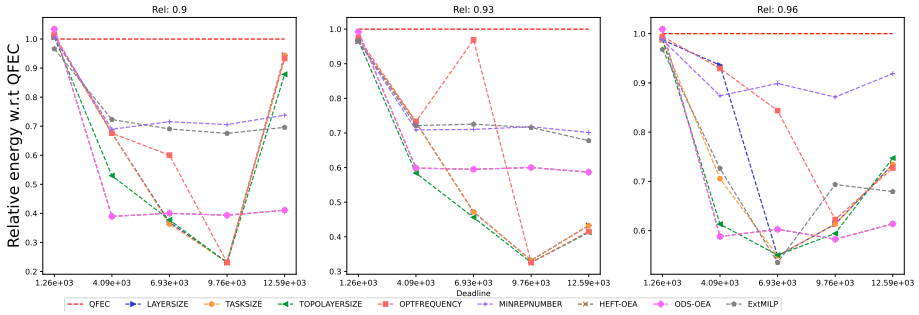

Figure 2010: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

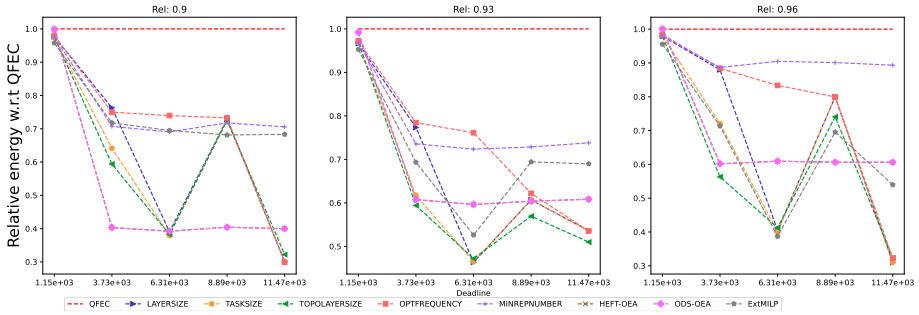

Figure 2011: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

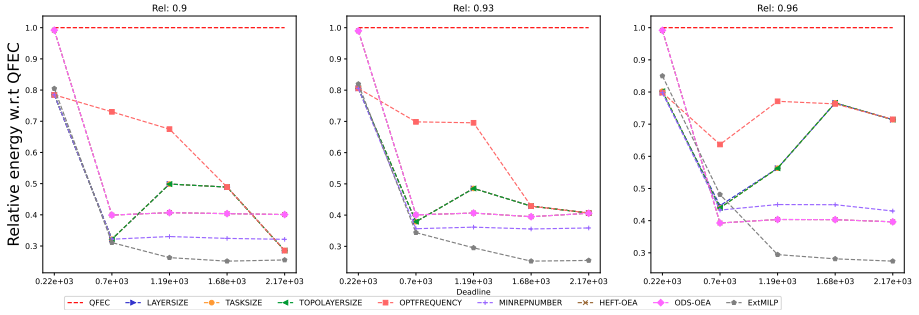

Figure 2012: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

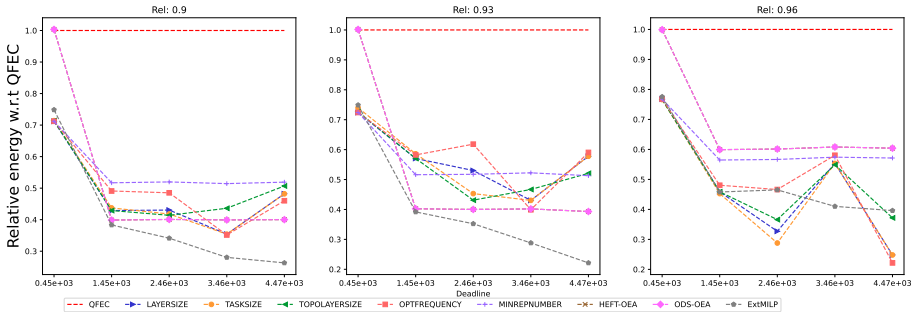

Figure 2013: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

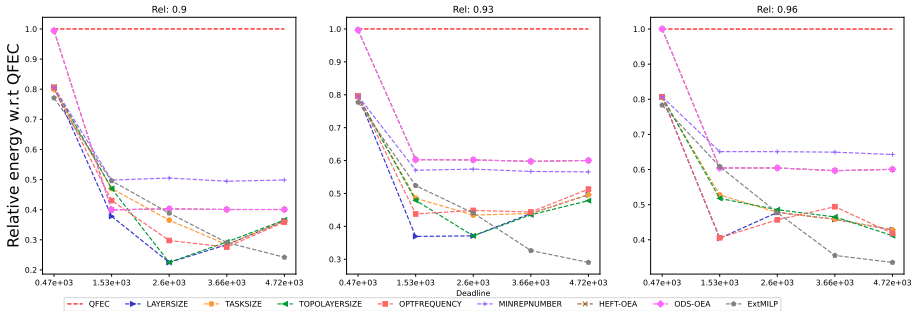

Figure 2014: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

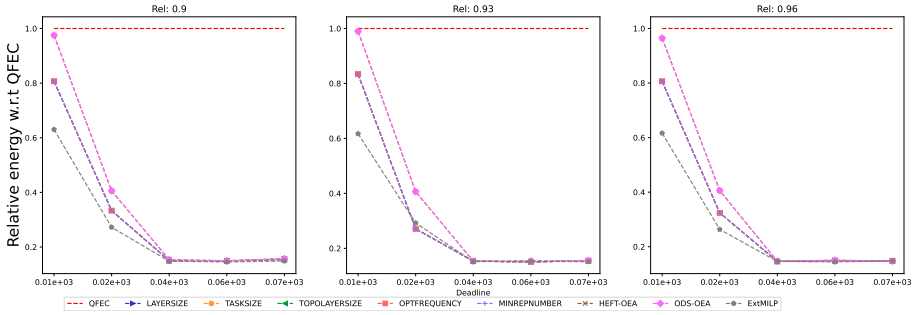

Figure 2015: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

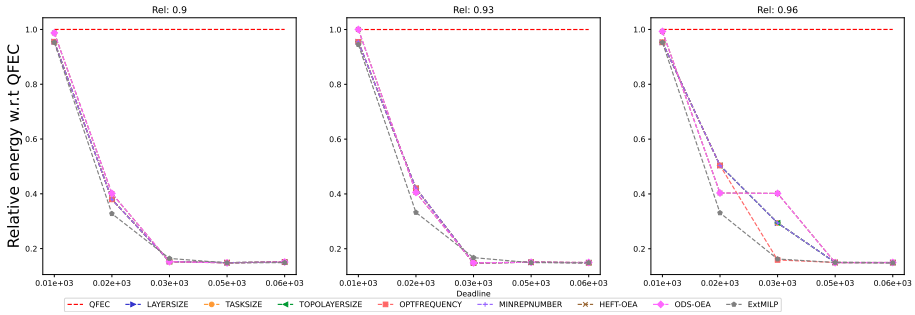

Figure 2016: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

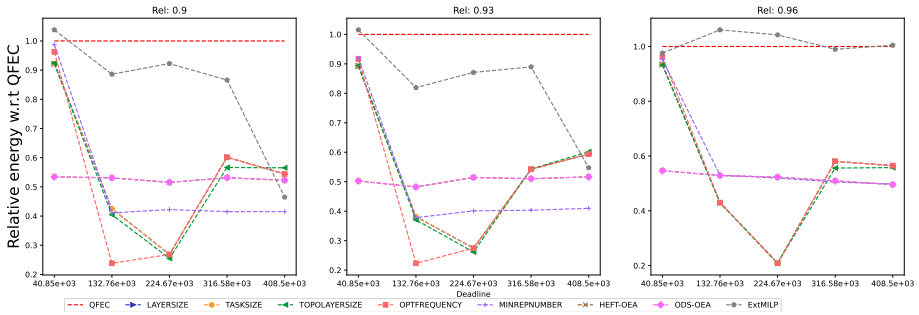

Figure 2017: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

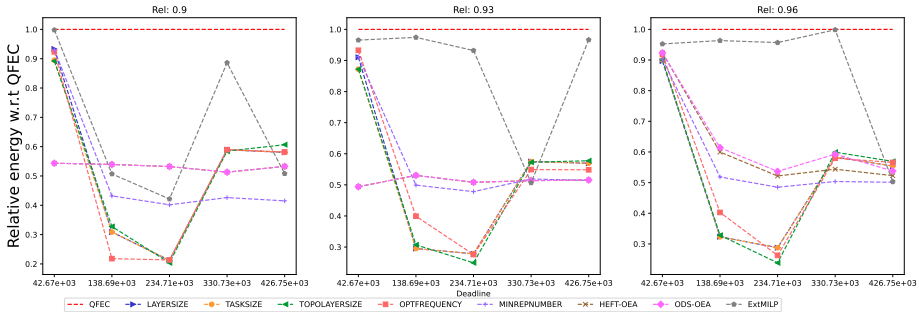

Figure 2018: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

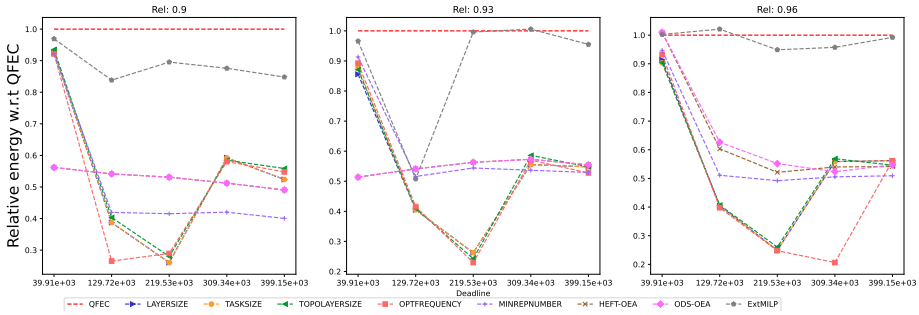

Figure 2019: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

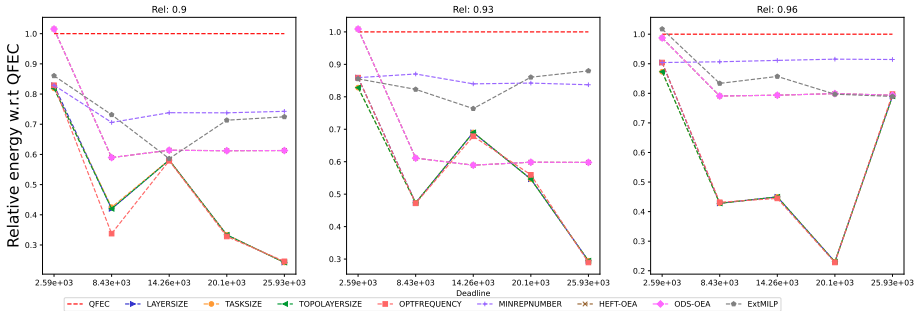

Figure 2020: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

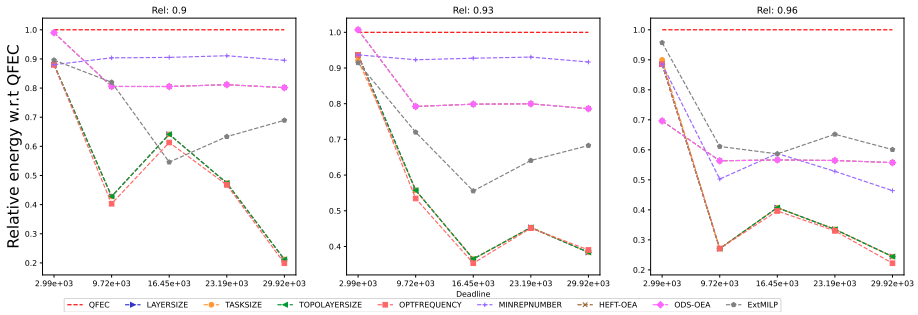

Figure 2021: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.7.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

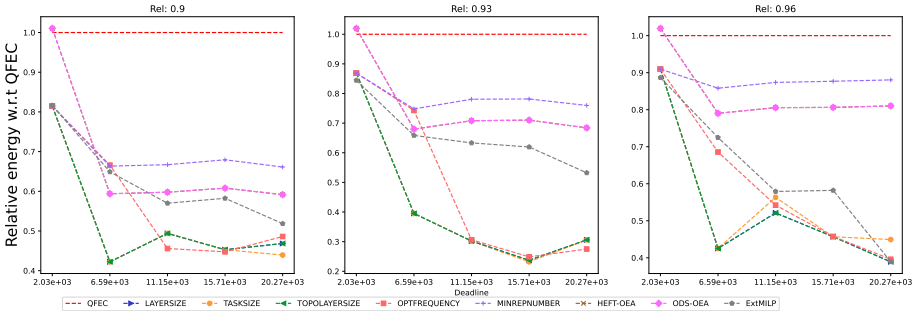

Figure 2022: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

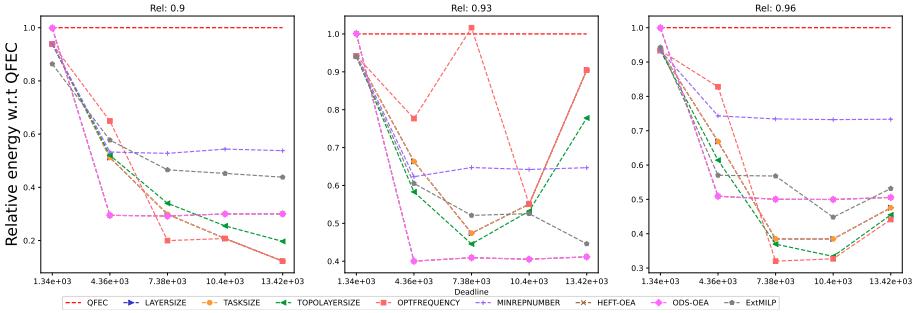

Figure 2023: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

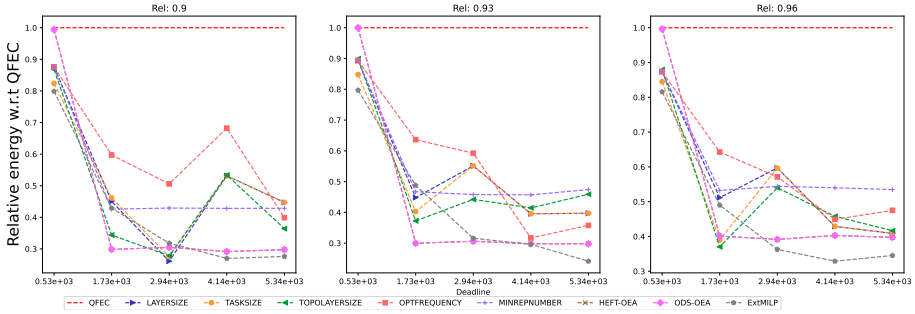

Figure 2024: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

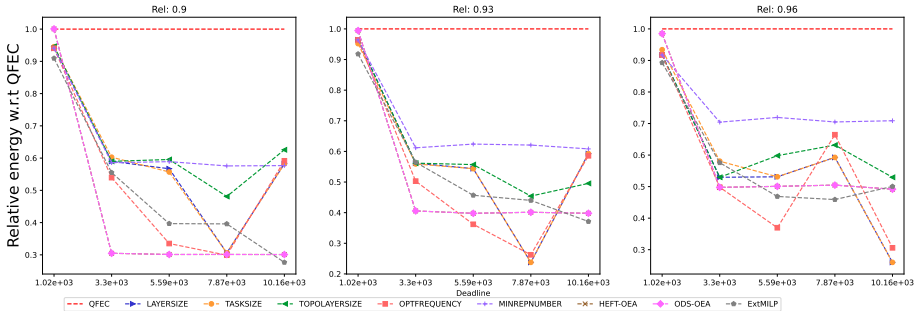

Figure 2025: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

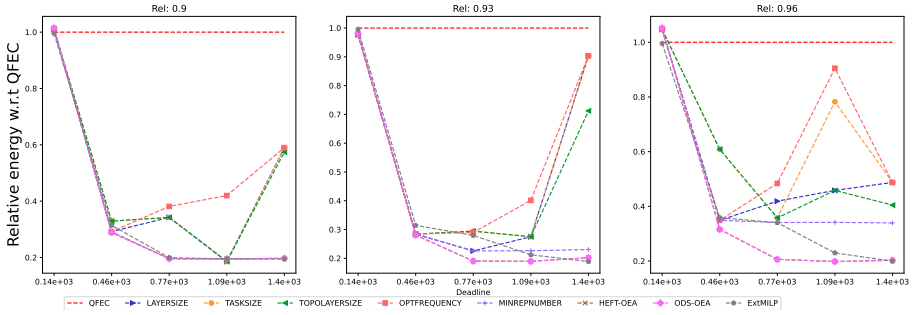

Figure 2026: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

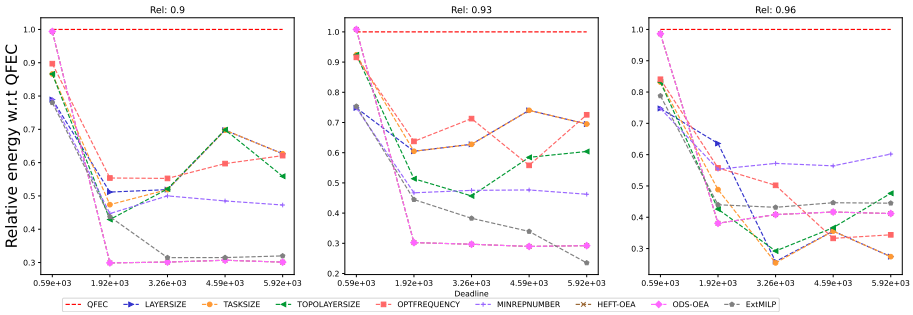

Figure 2027: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

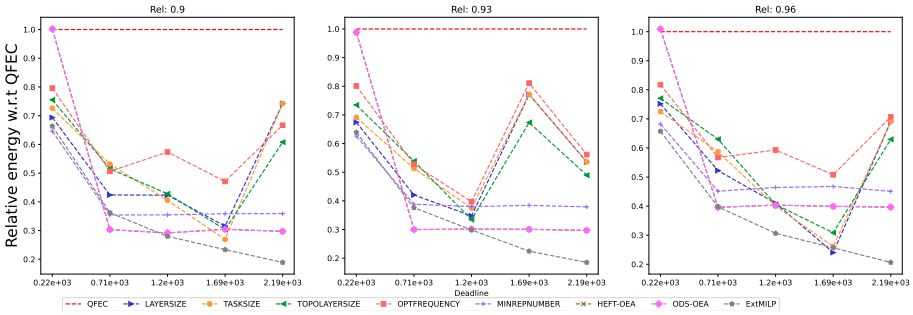

Figure 2028: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

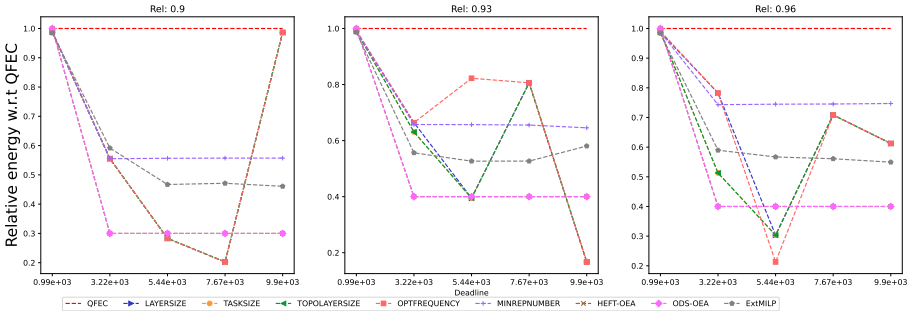

Figure 2029: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

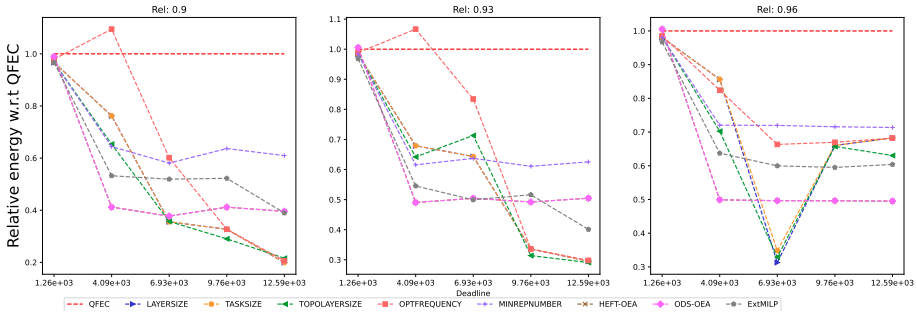

Figure 2030: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

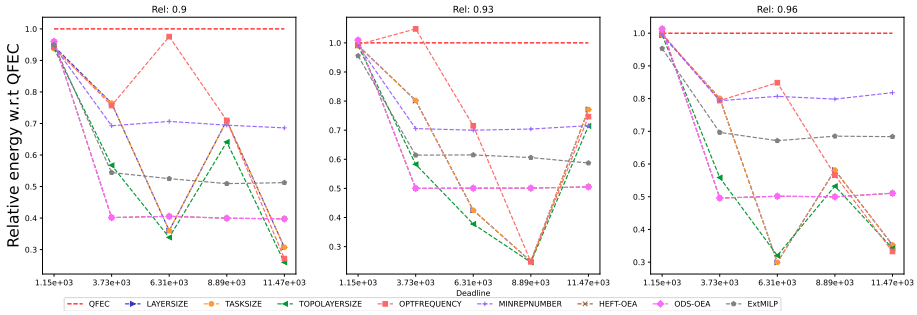

Figure 2031: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

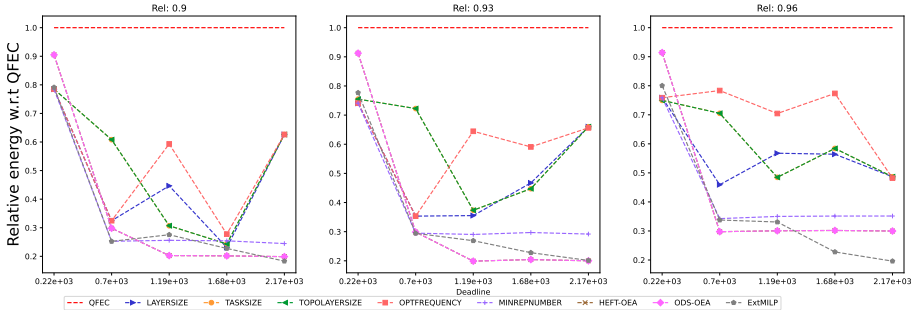

Figure 2032: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

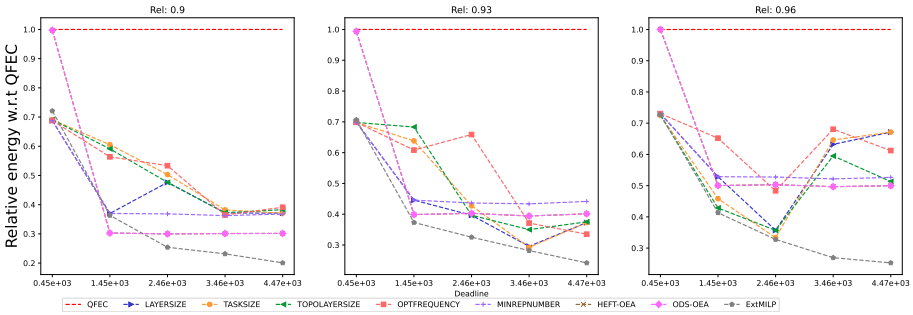

Figure 2033: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

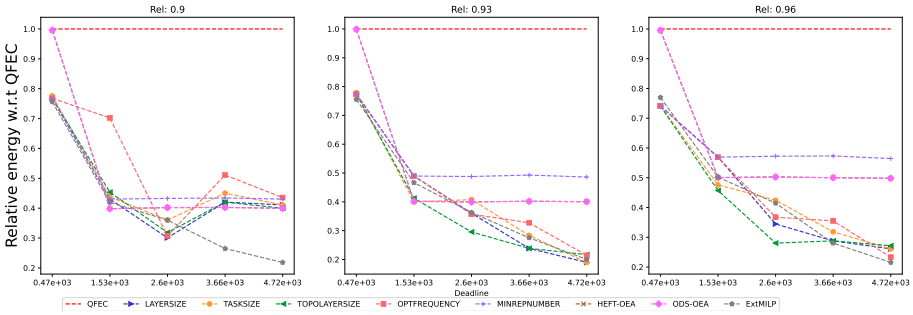

Figure 2034: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

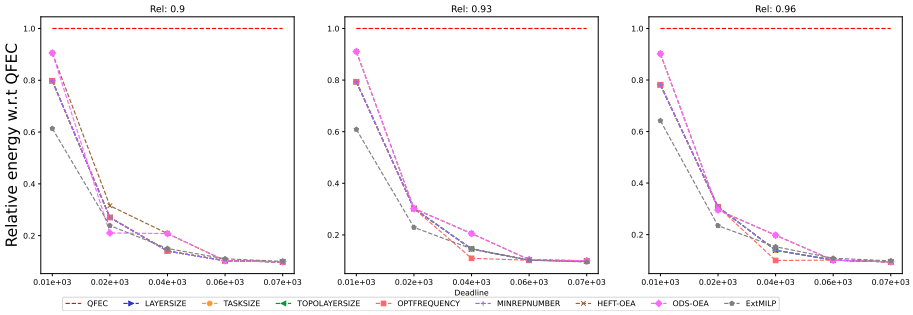

Figure 2035: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

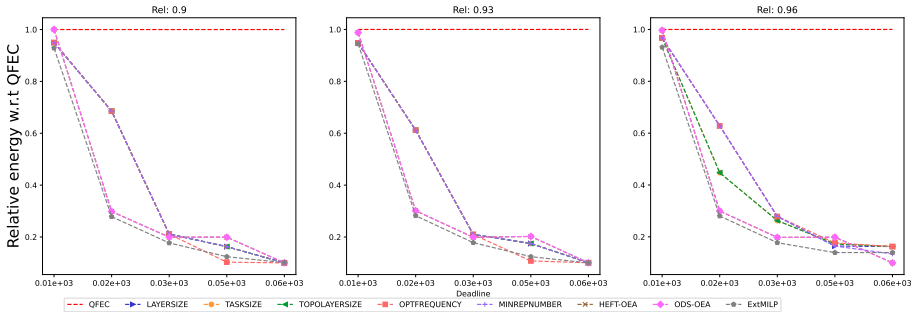

Figure 2036: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

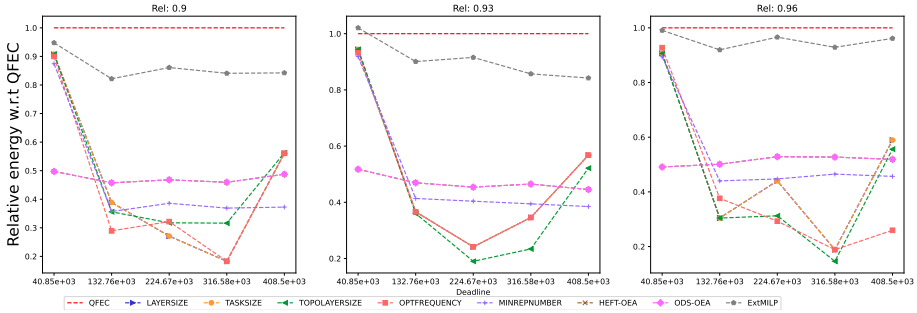

Figure 2037: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

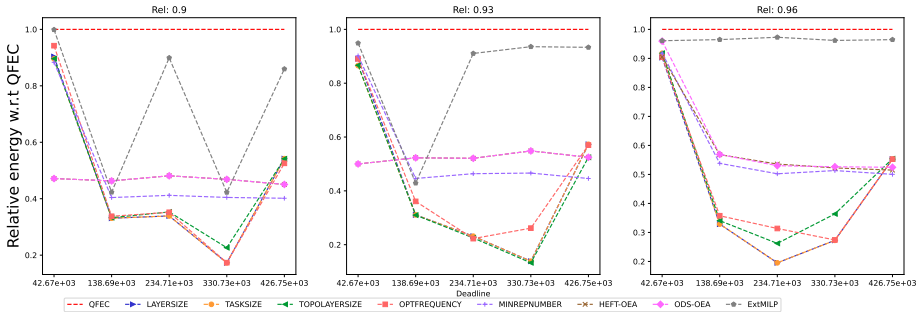

Figure 2038: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

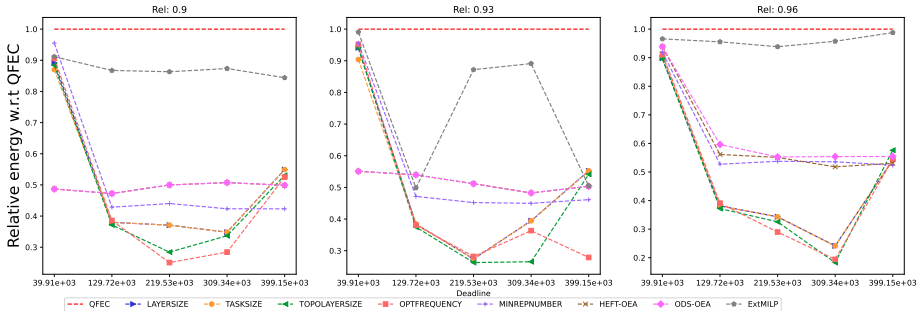

Figure 2039: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

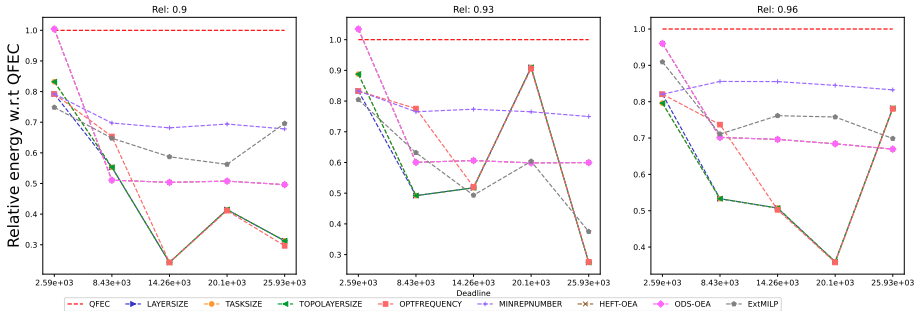

Figure 2040: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

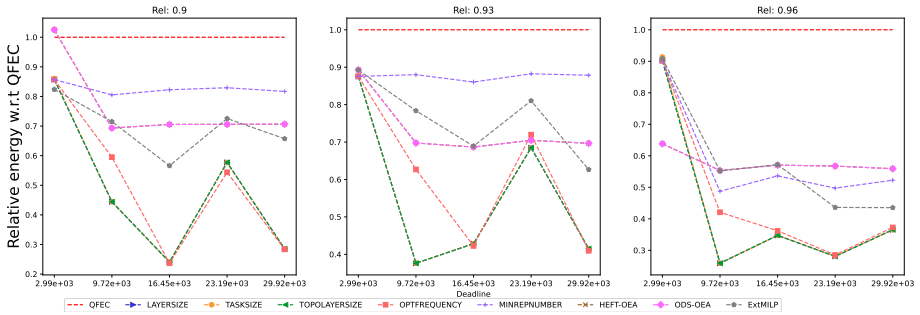

Figure 2041: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.7.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

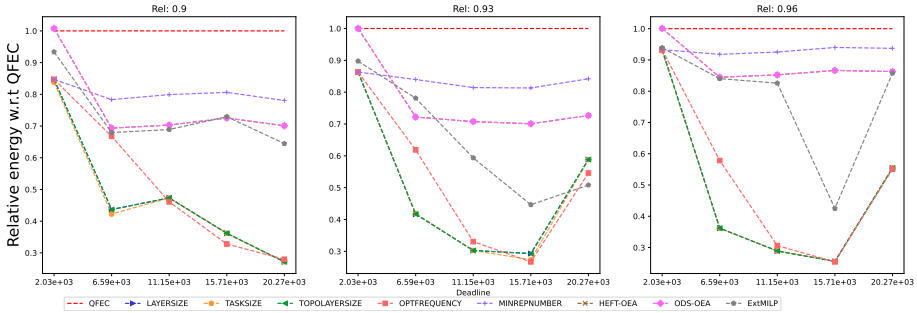

Figure 2042: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

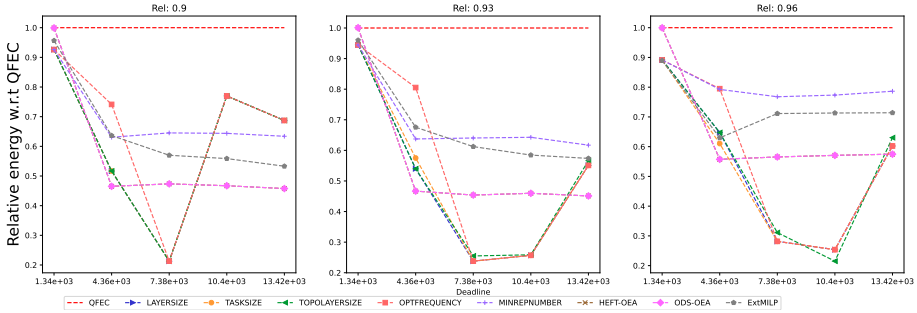

Figure 2043: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

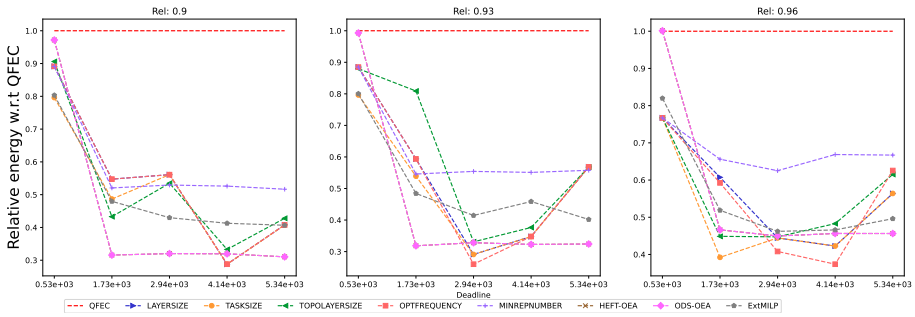

Figure 2044: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

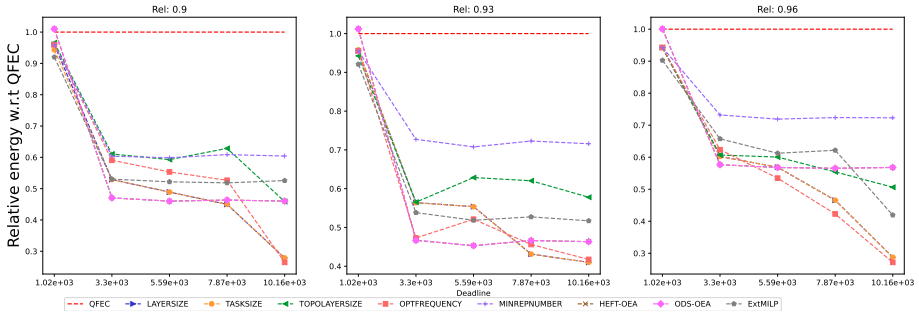

Figure 2045: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

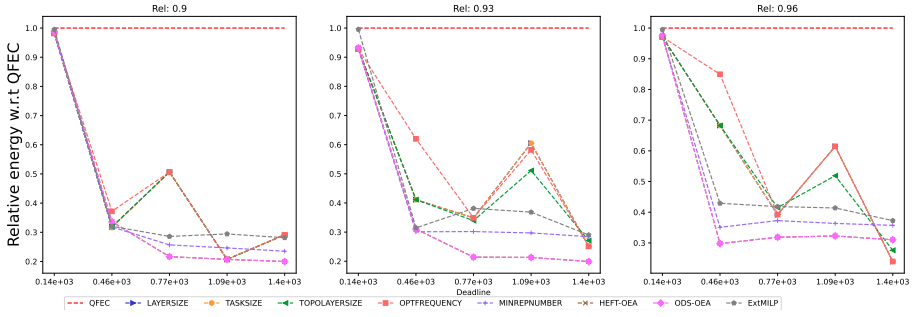

Figure 2046: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

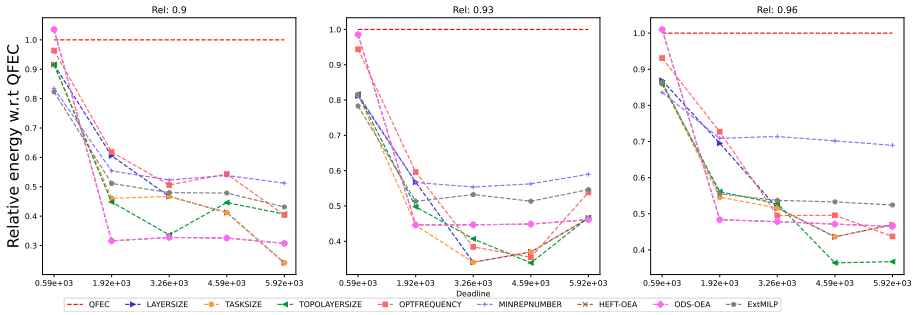

Figure 2047: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

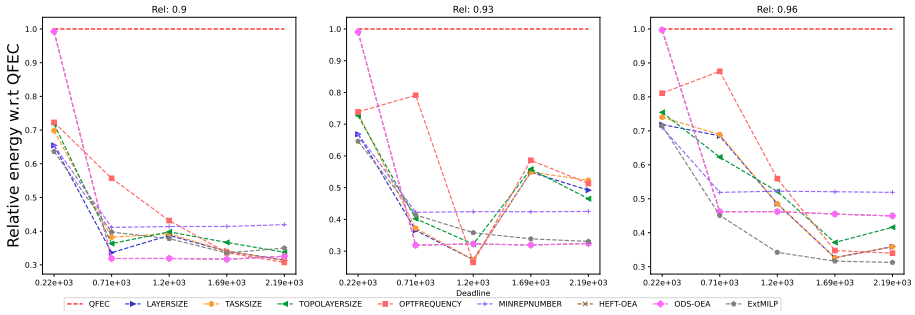

Figure 2048: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

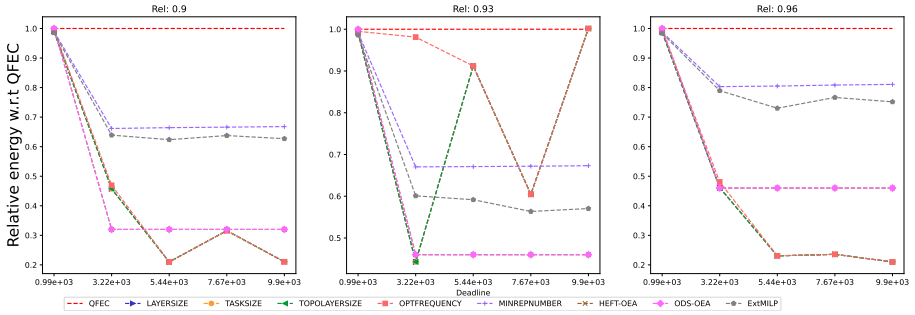

Figure 2049: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

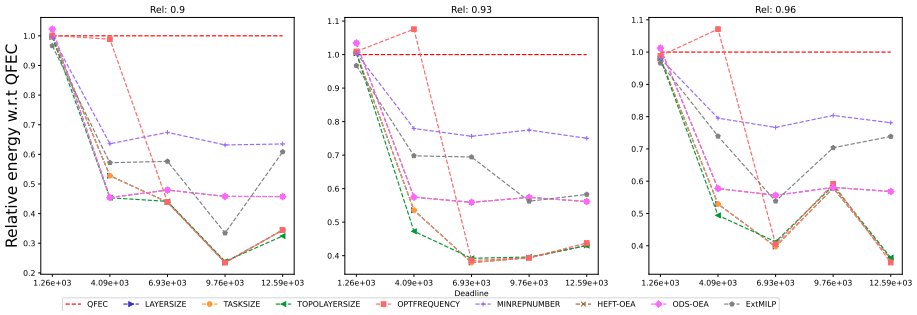

Figure 2050: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

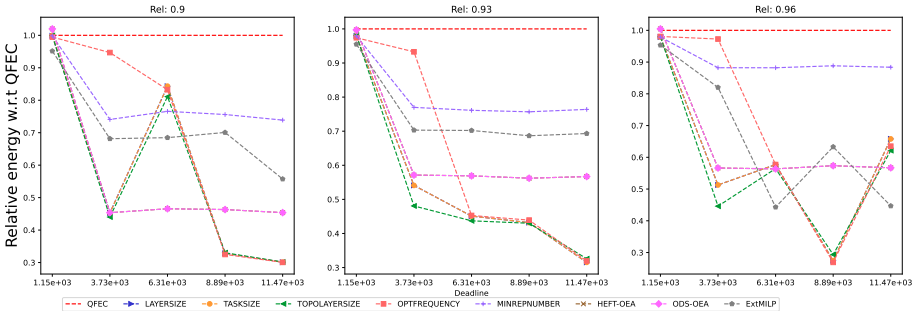

Figure 2051: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

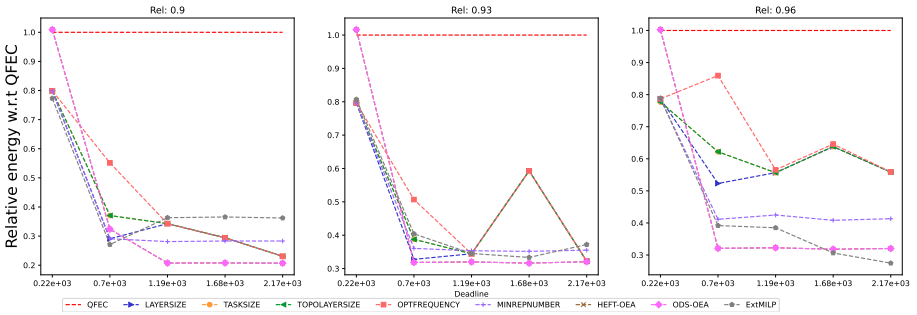

Figure 2052: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

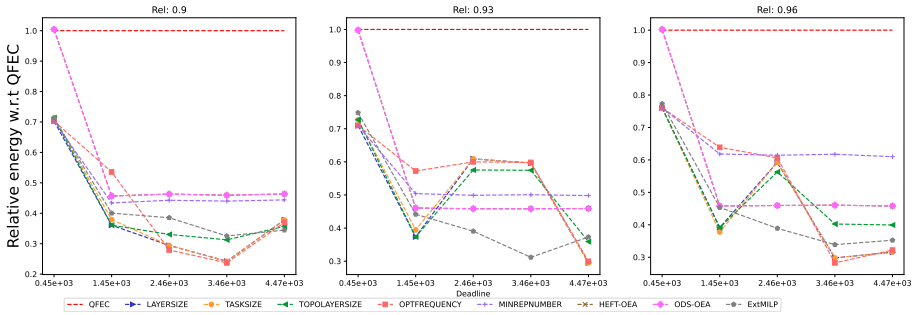

Figure 2053: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

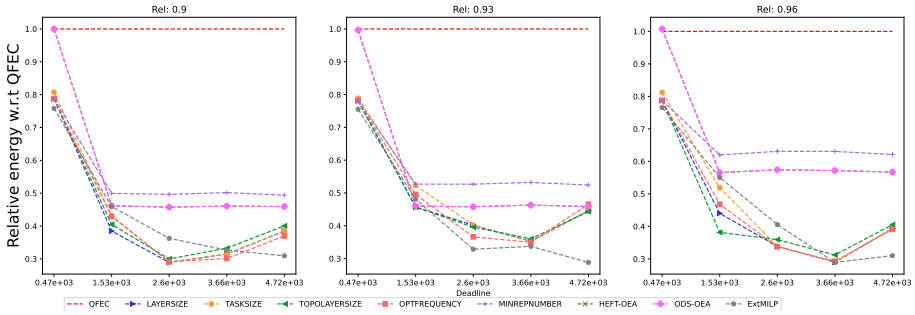

Figure 2054: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

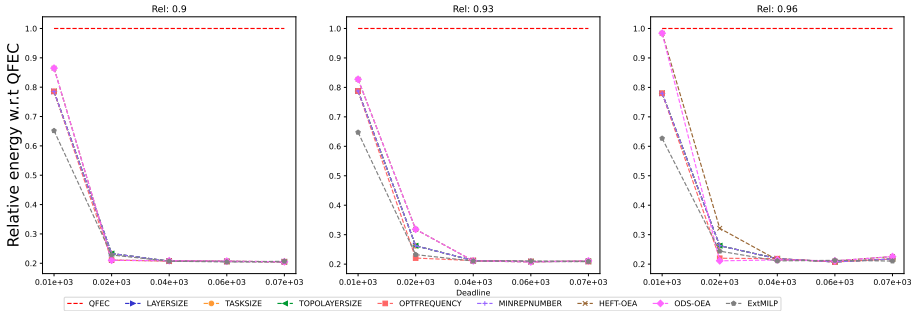

Figure 2055: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

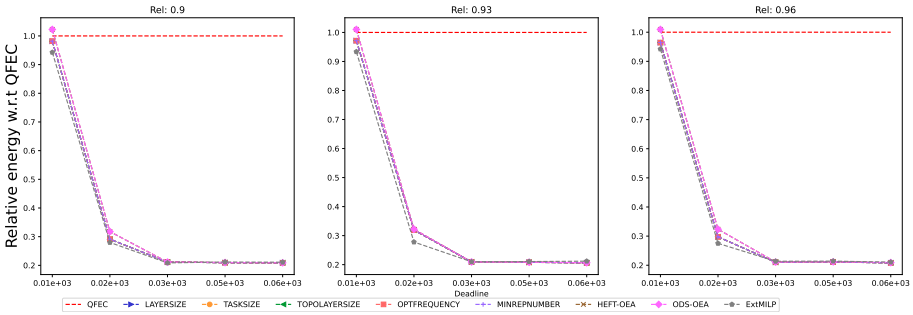

Figure 2056: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

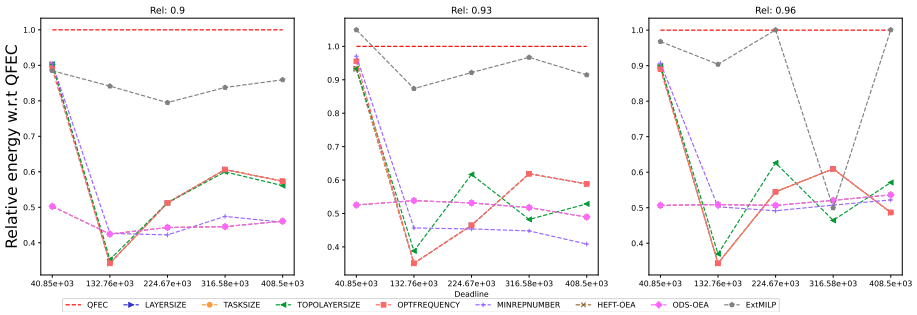

Figure 2057: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

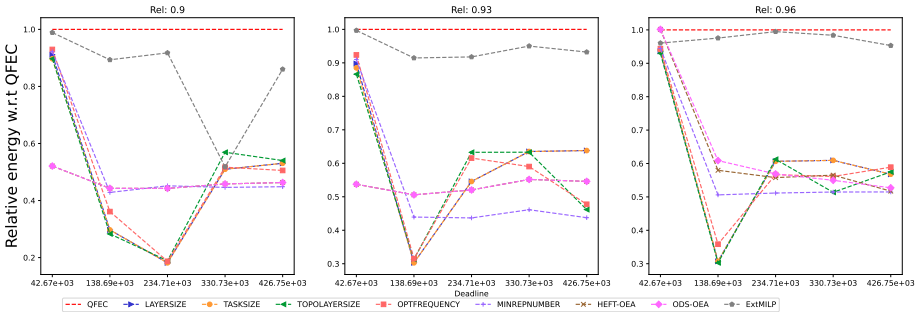

Figure 2058: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

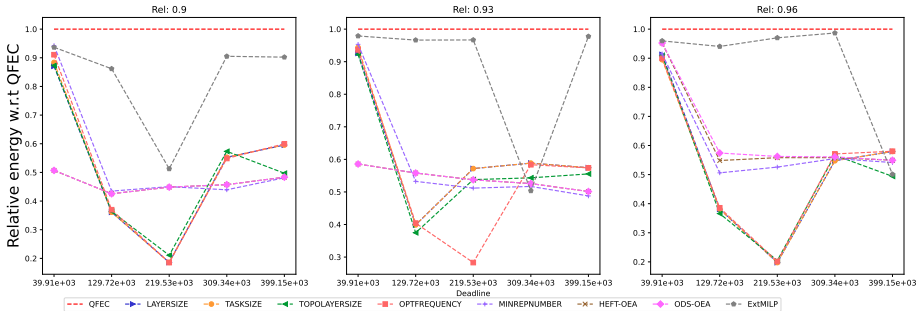

Figure 2059: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

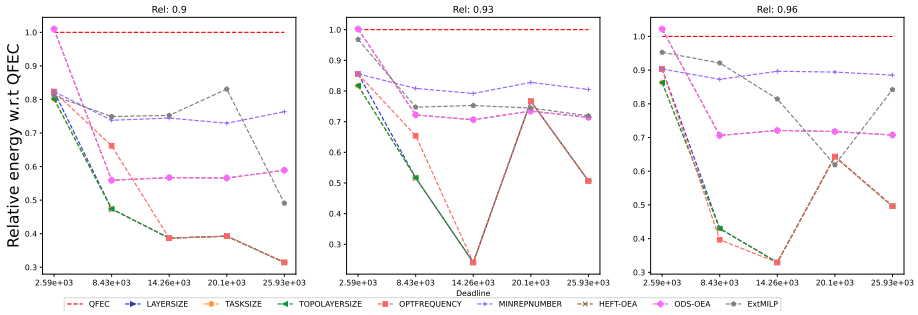

Figure 2060: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

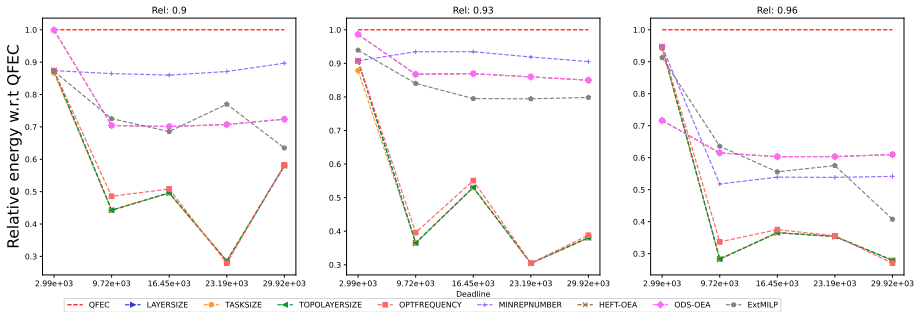

Figure 2061: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.7.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

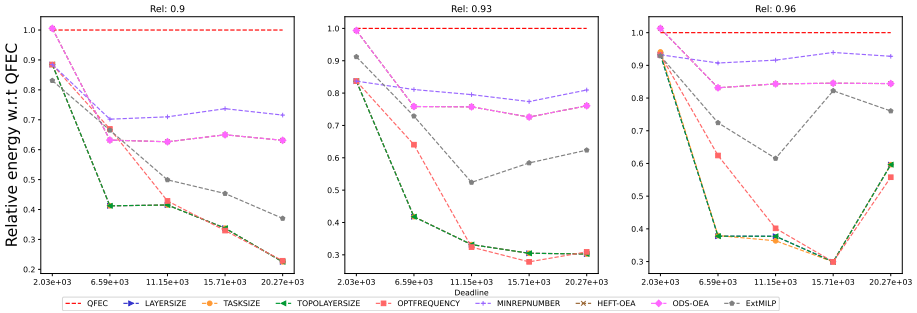

Figure 2062: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

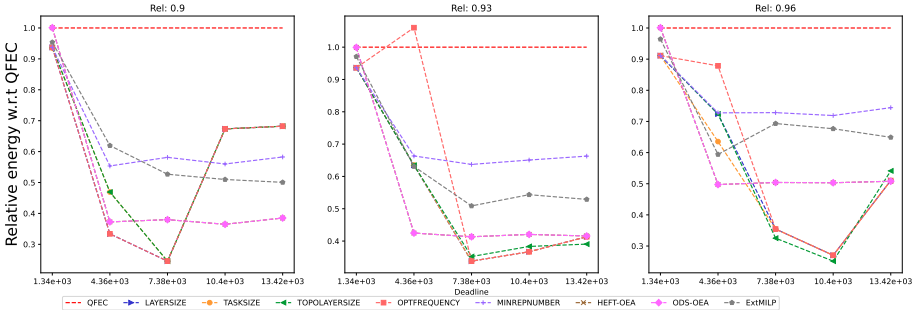

Figure 2063: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

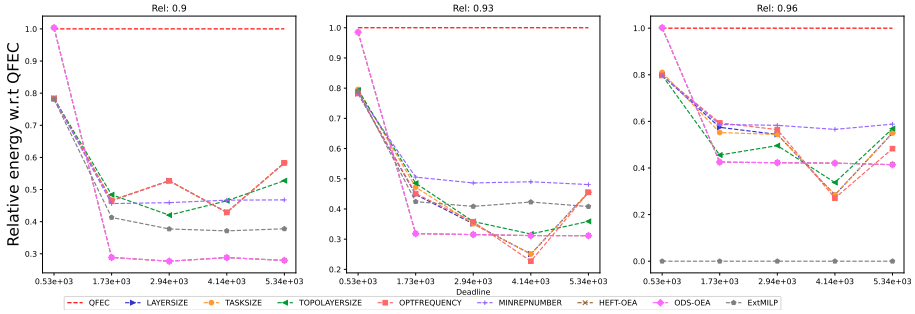

Figure 2064: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

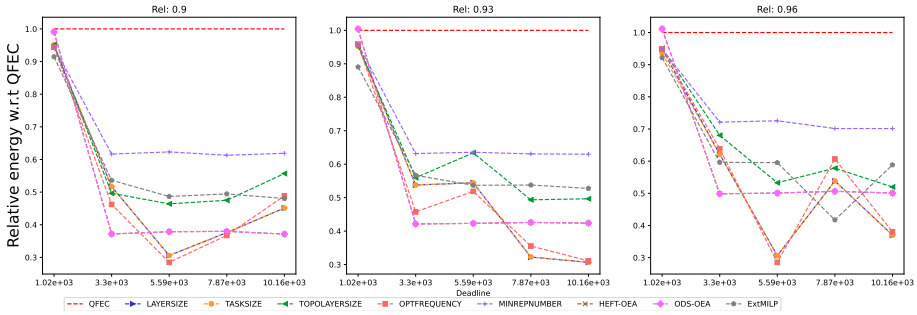

Figure 2065: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

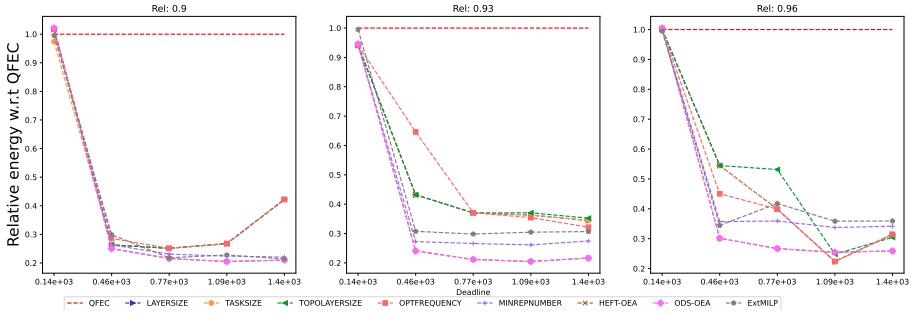

Figure 2066: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

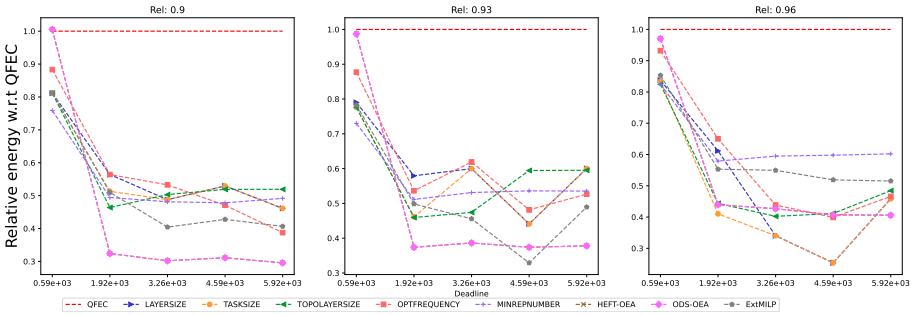

Figure 2067: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

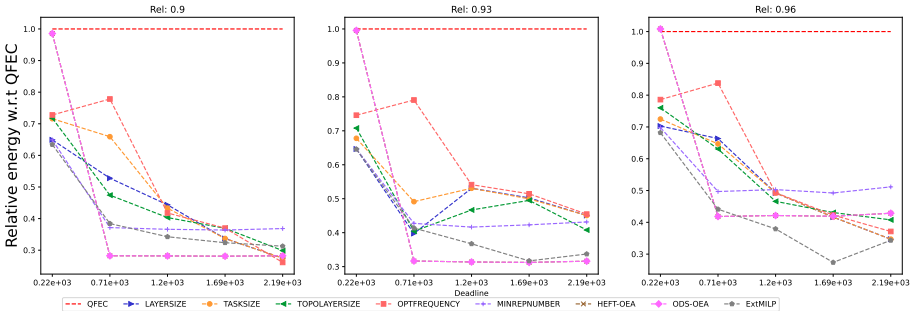

Figure 2068: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

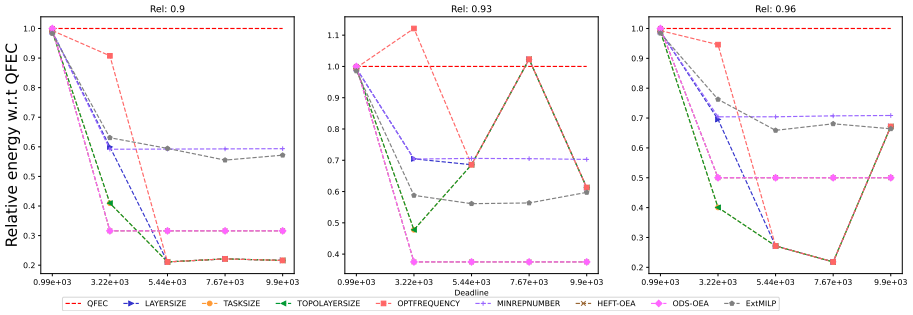

Figure 2069: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

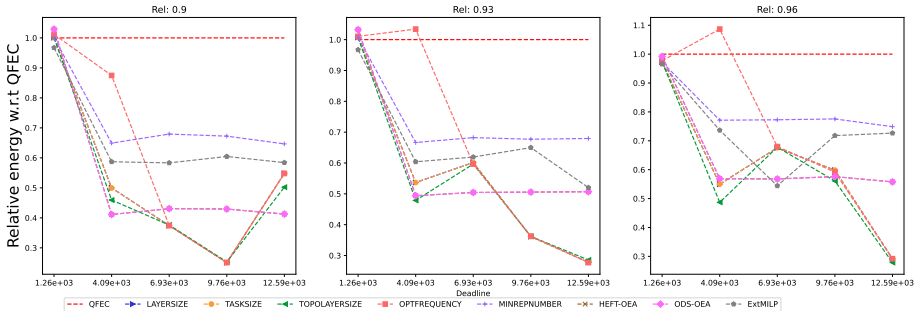

Figure 2070: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

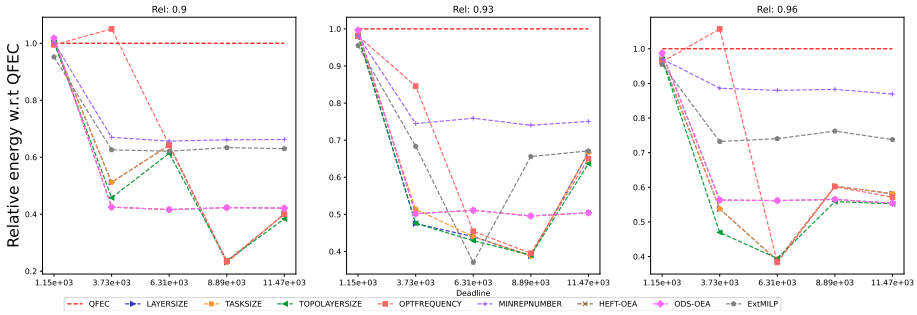

Figure 2071: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

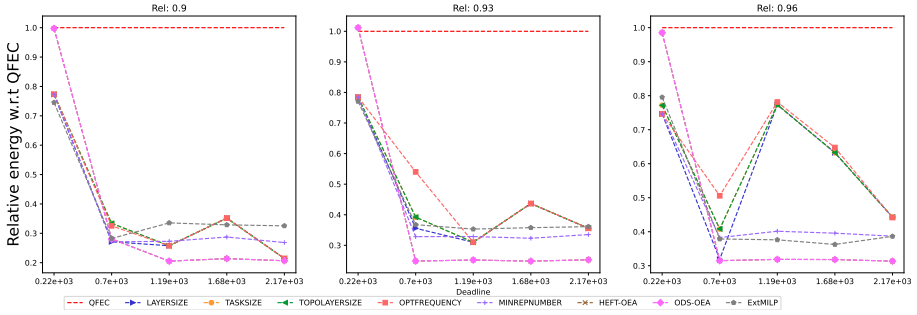

Figure 2072: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

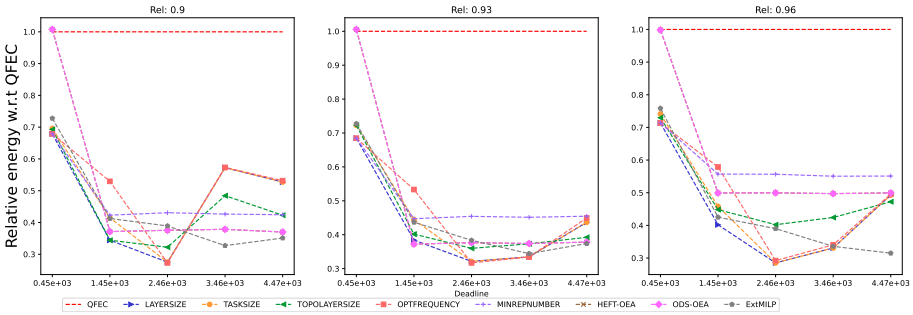

Figure 2073: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

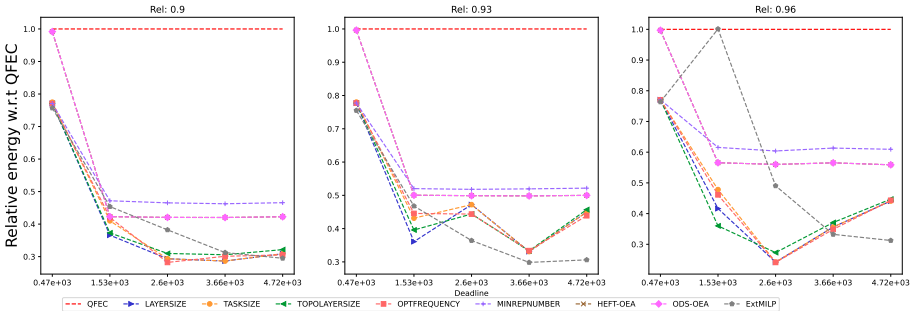

Figure 2074: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

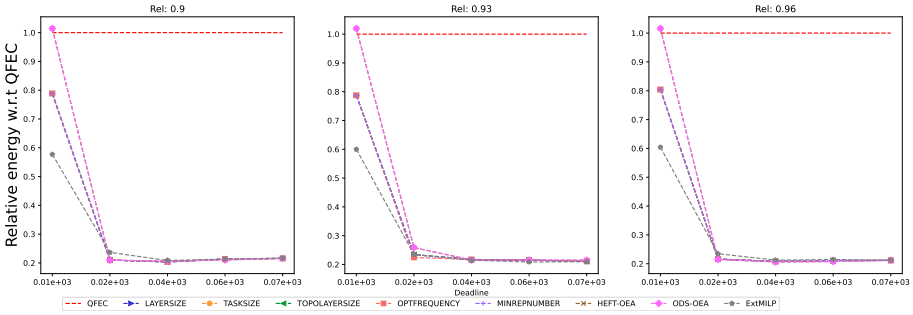

Figure 2075: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

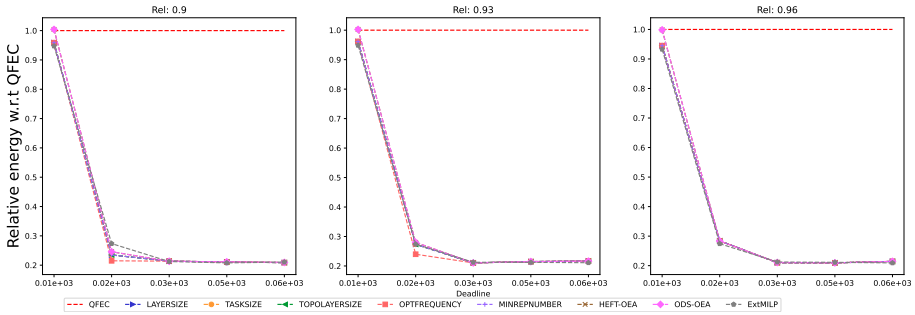

Figure 2076: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

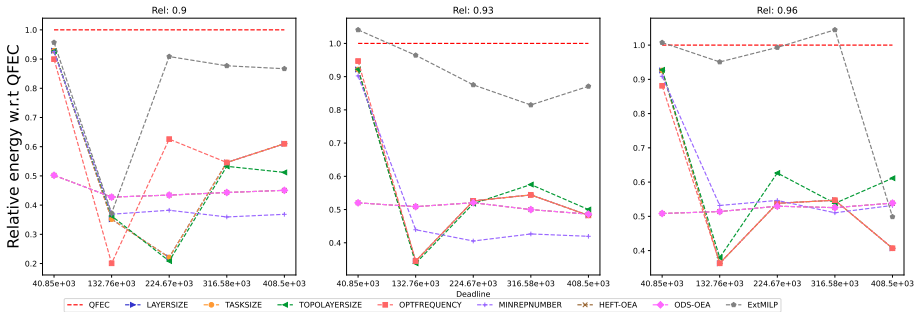

Figure 2077: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

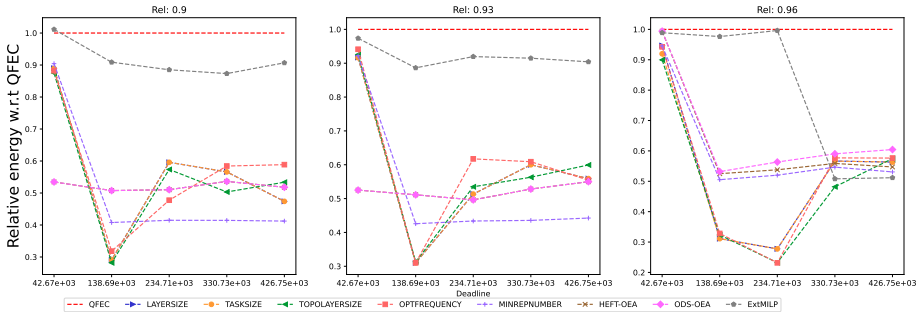

Figure 2078: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

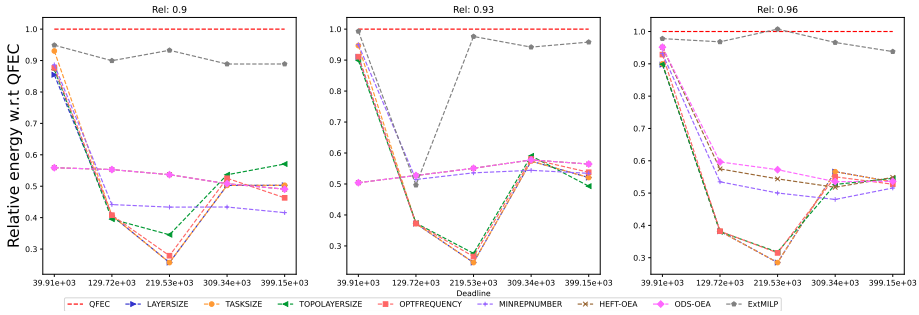

Figure 2079: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

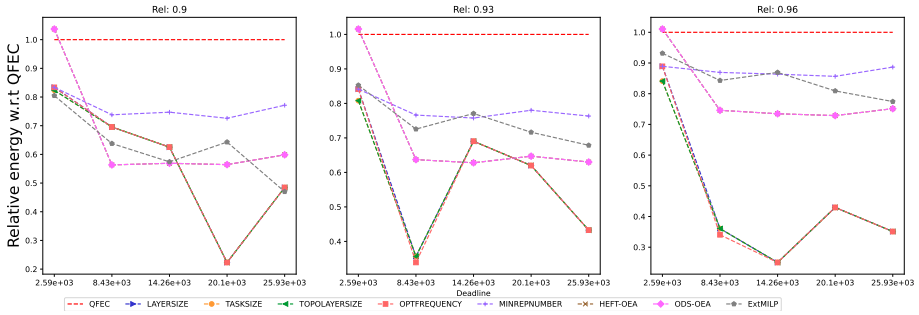

Figure 2080: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

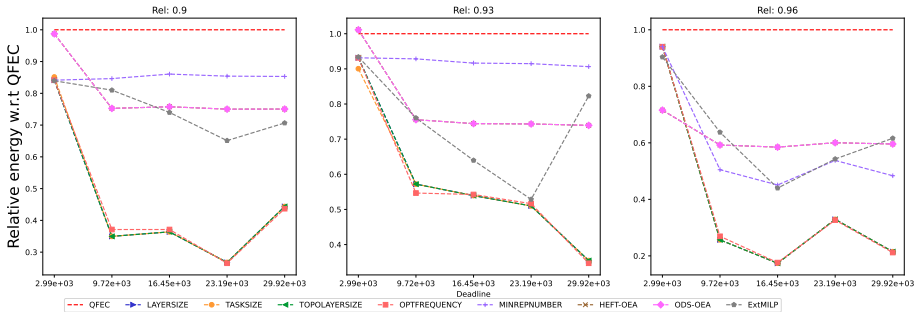

Figure 2081: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.8 $BC/WC = 0.8$

G.8.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

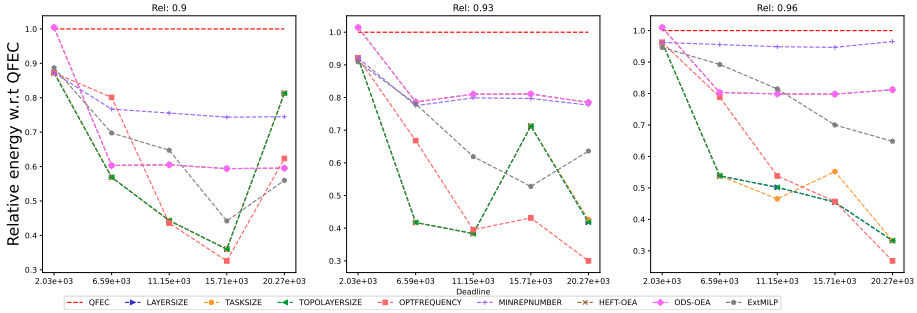

Figure 2082: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

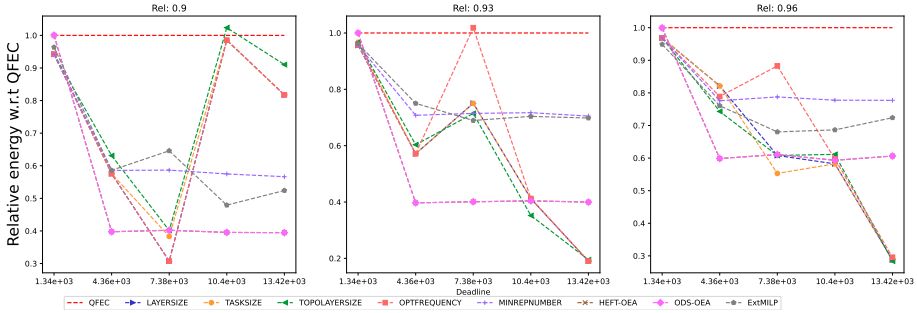

Figure 2083: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

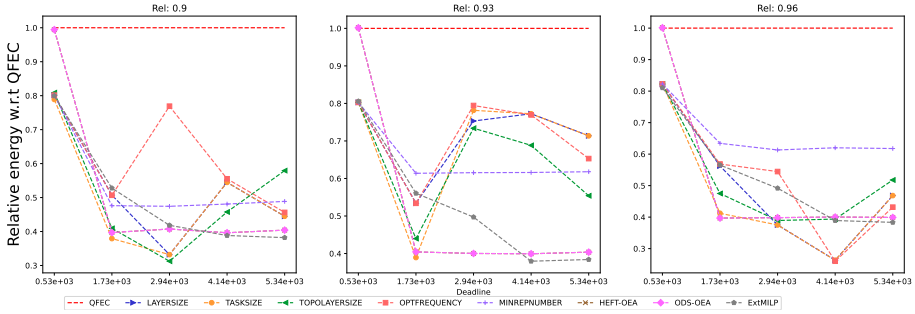

Figure 2084: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

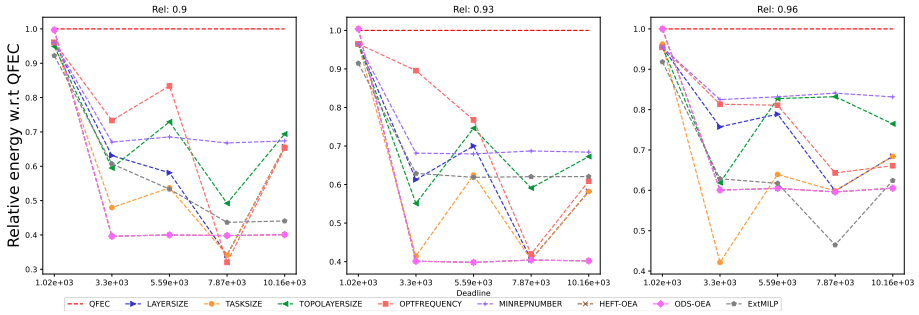

Figure 2085: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

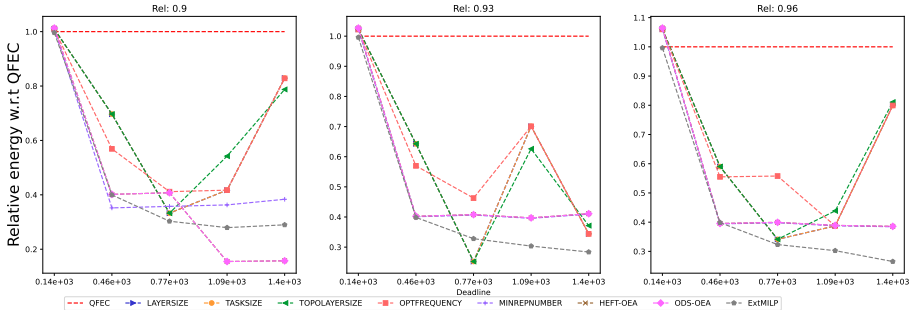

Figure 2086: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

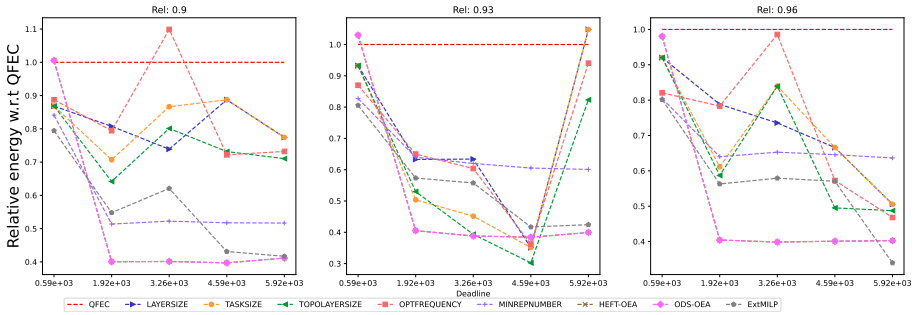

Figure 2087: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

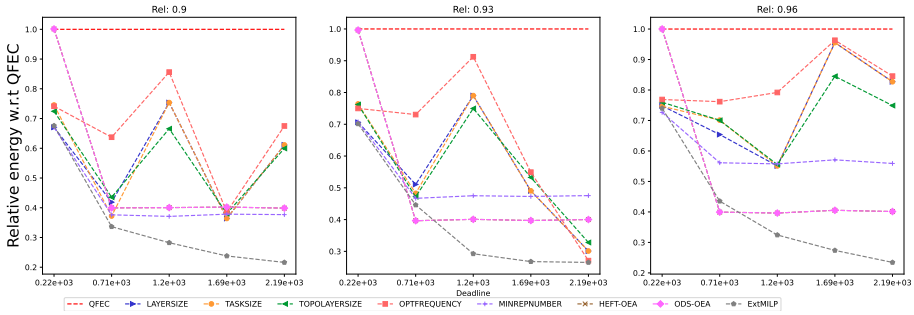

Figure 2088: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

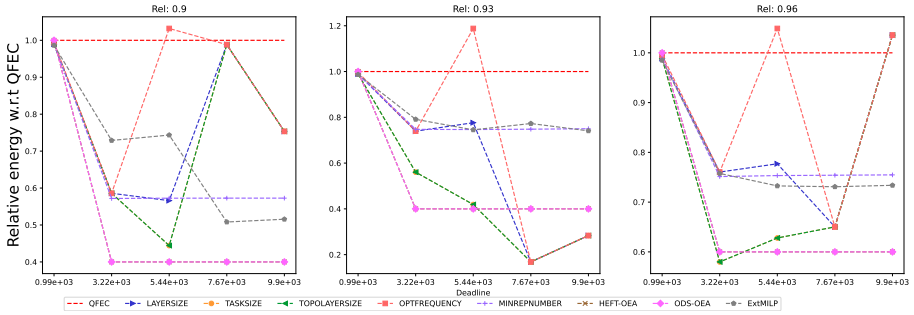

Figure 2089: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

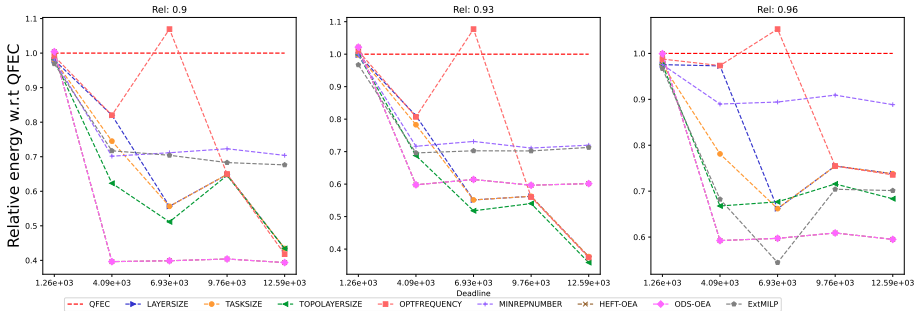

Figure 2090: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

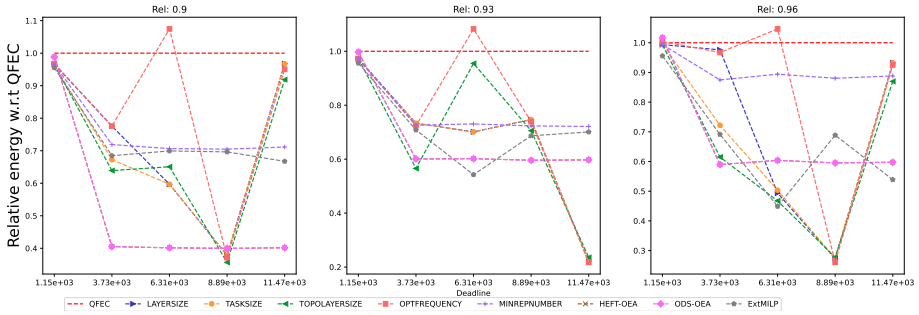

Figure 2091: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

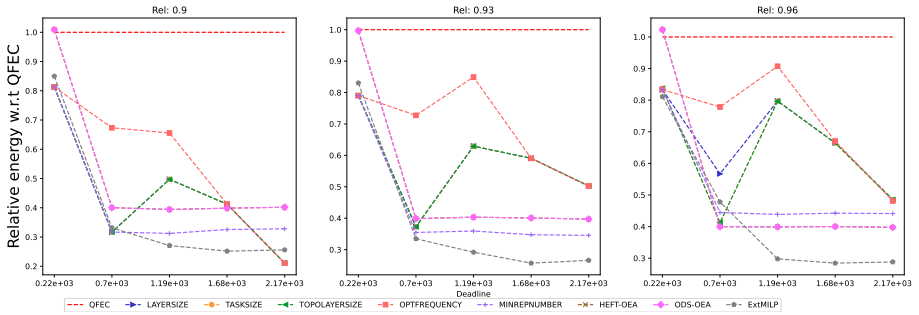

Figure 2092: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

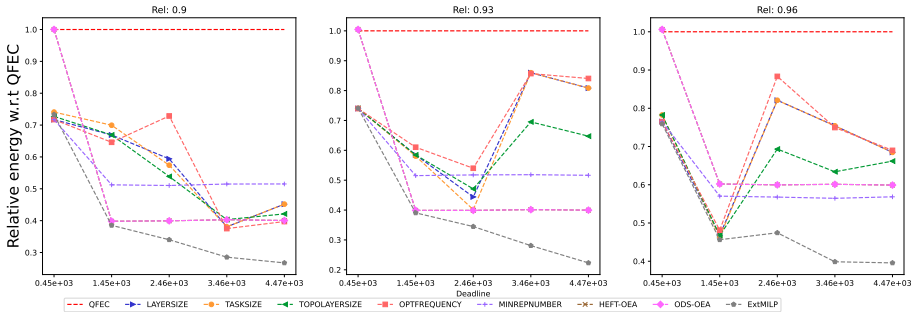

Figure 2093: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

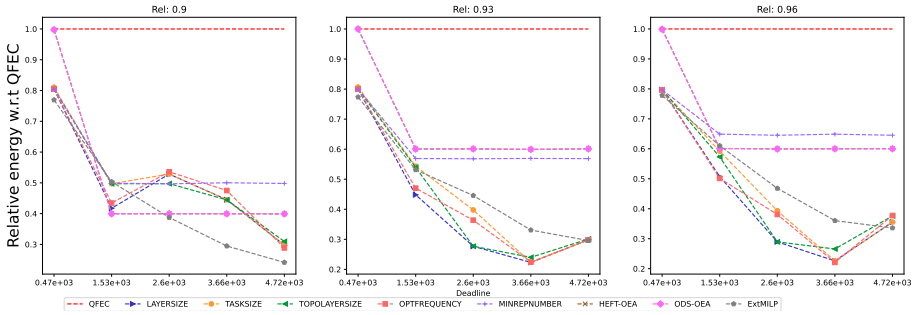

Figure 2094: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

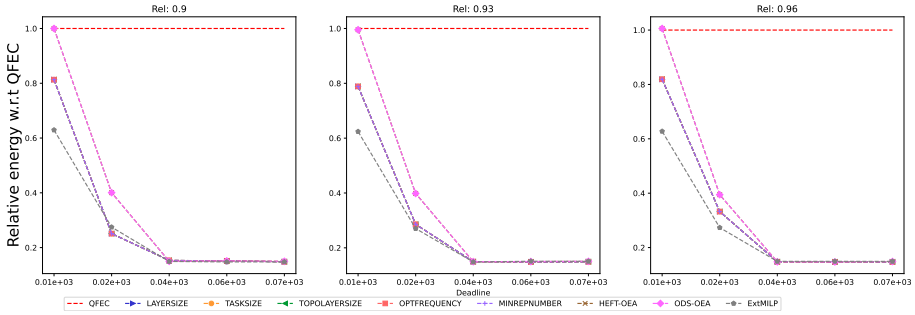

Figure 2095: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

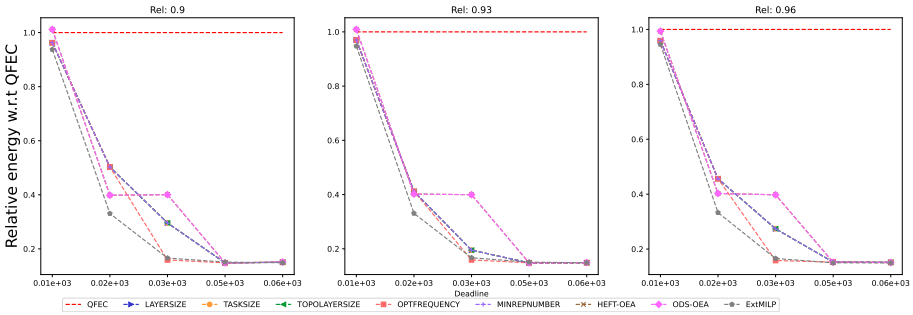

Figure 2096: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

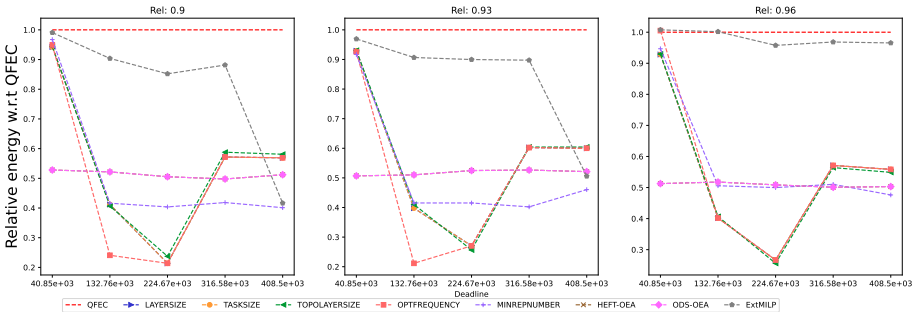

Figure 2097: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

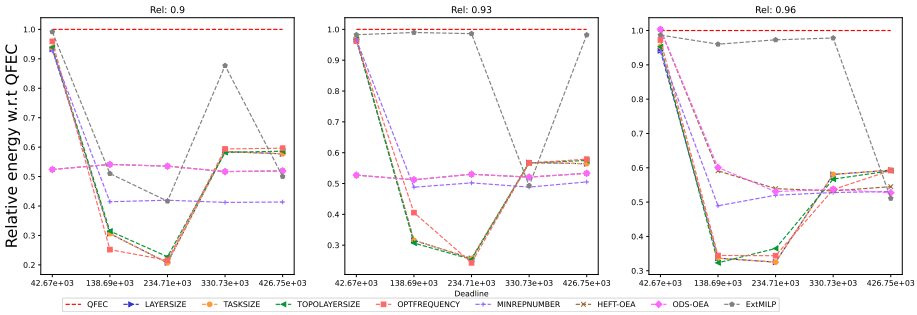

Figure 2098: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

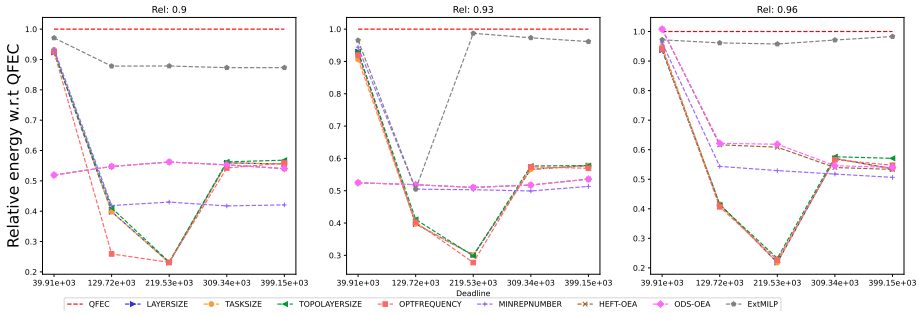

Figure 2099: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

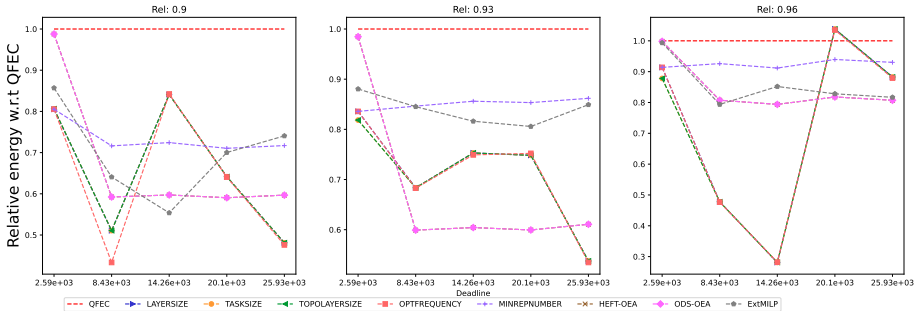

Figure 2100: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

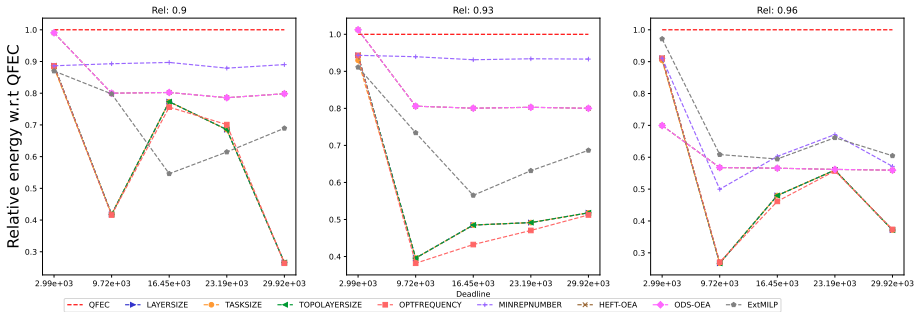

Figure 2101: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.8.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

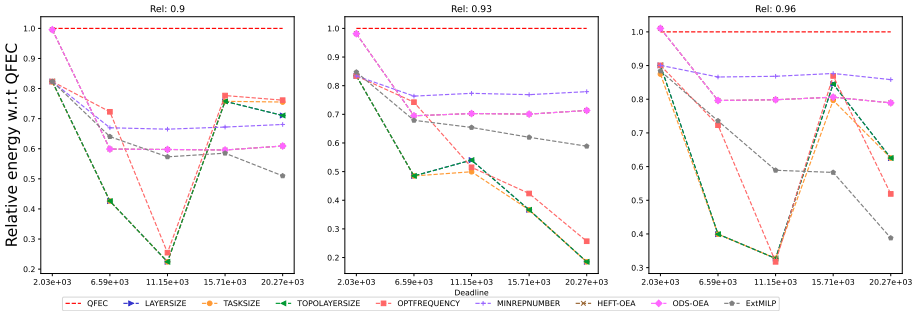

Figure 2102: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

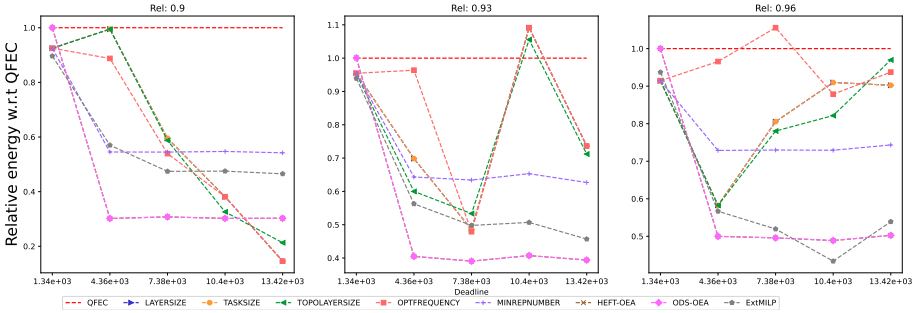

Figure 2103: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

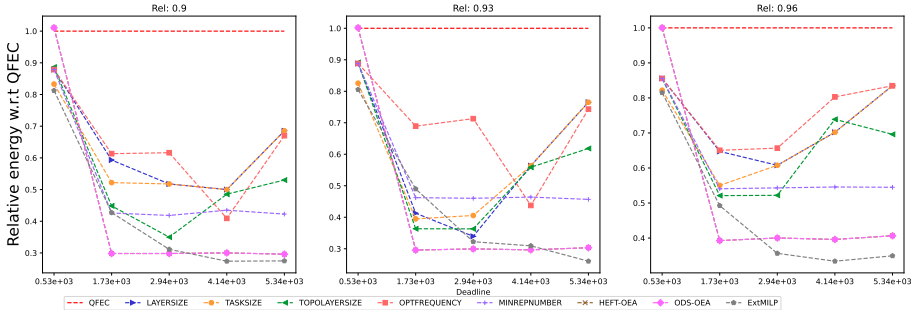

Figure 2104: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

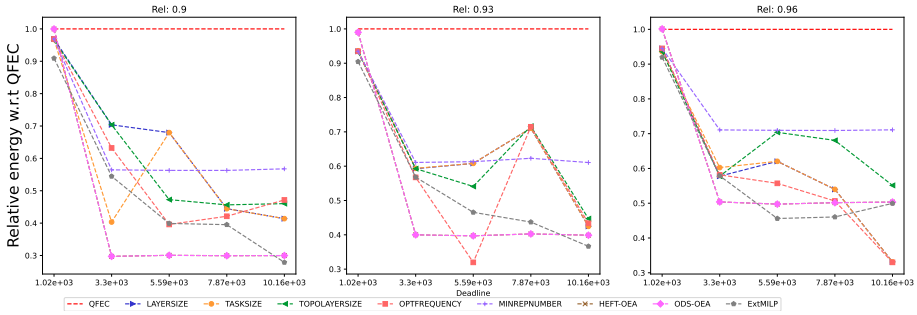

Figure 2105: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

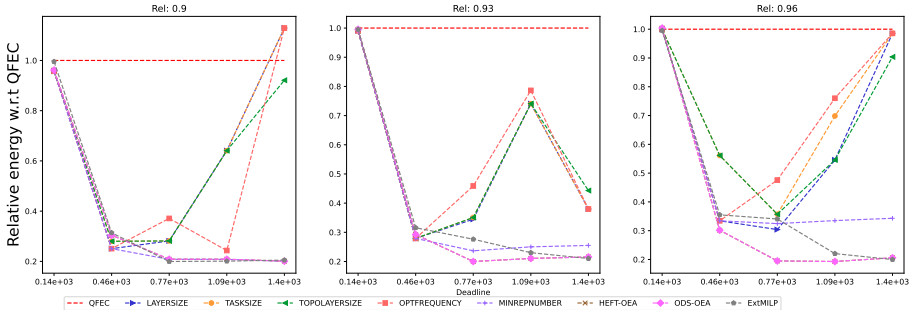

Figure 2106: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

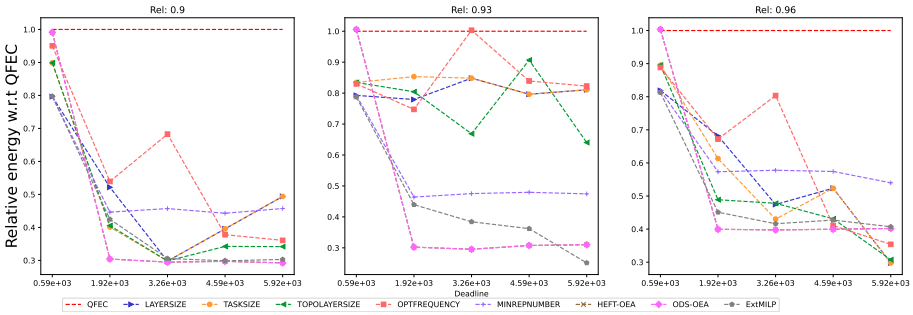

Figure 2107: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

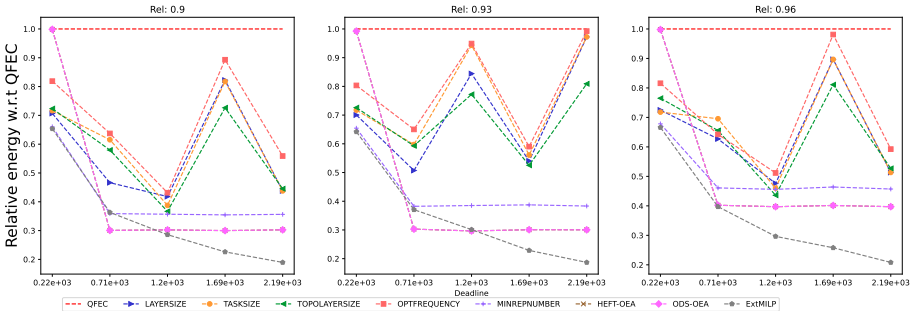

Figure 2108: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

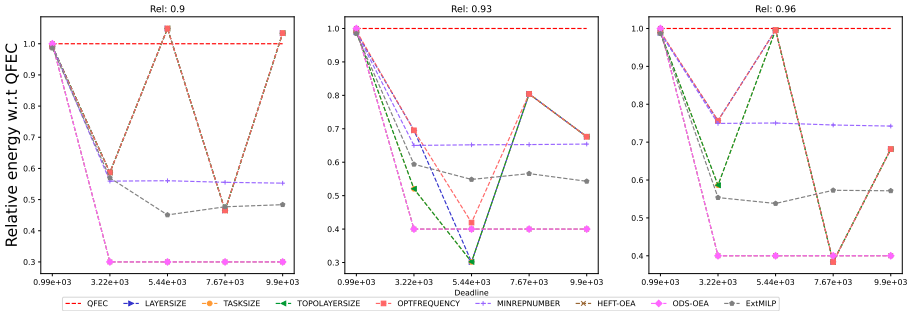

Figure 2109: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

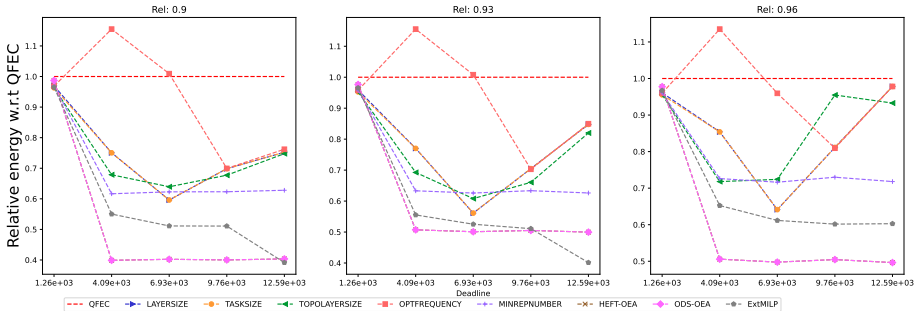

Figure 2110: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

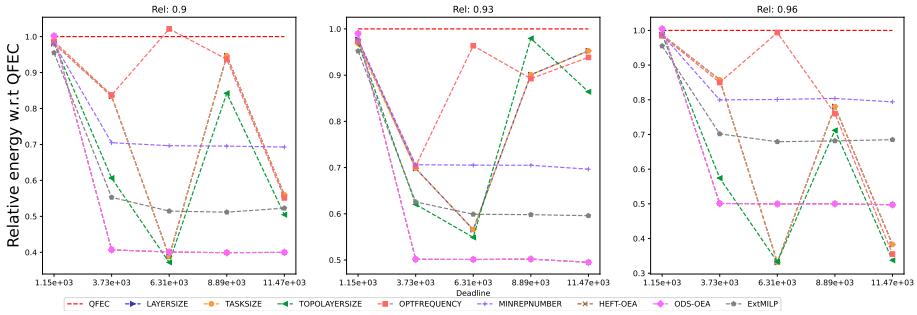

Figure 2111: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

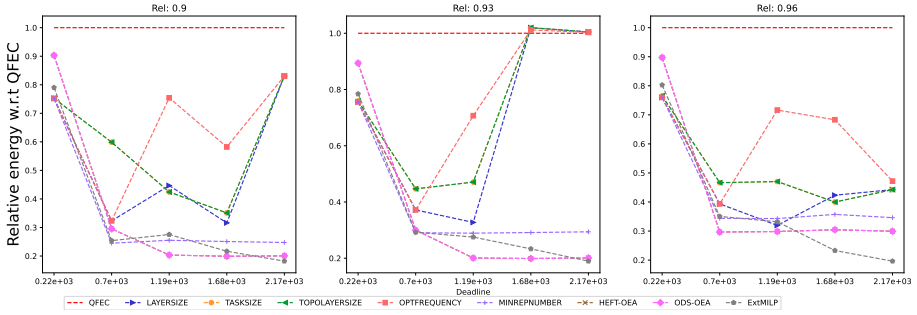

Figure 2112: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

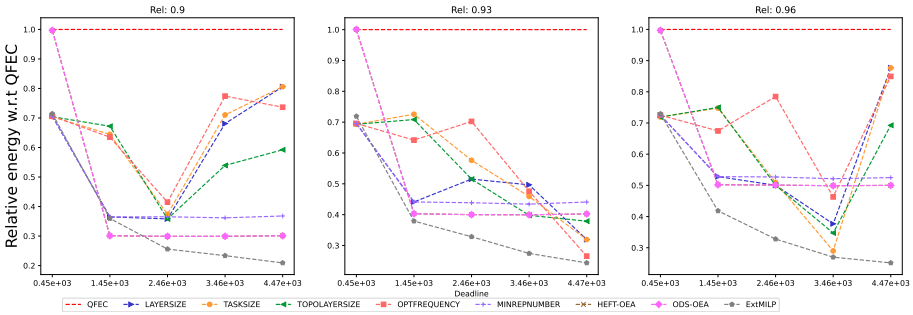

Figure 2113: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

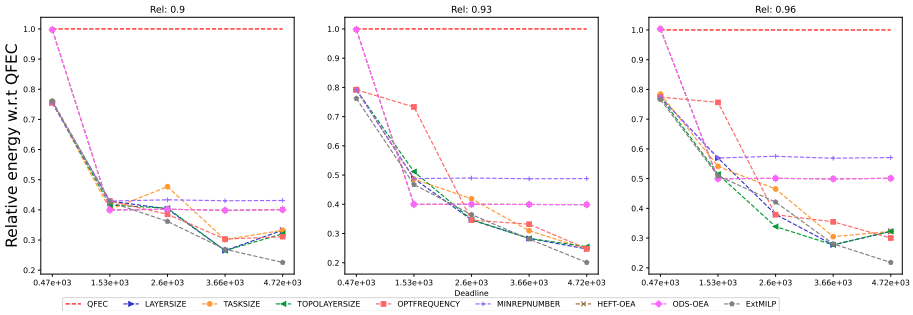

Figure 2114: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

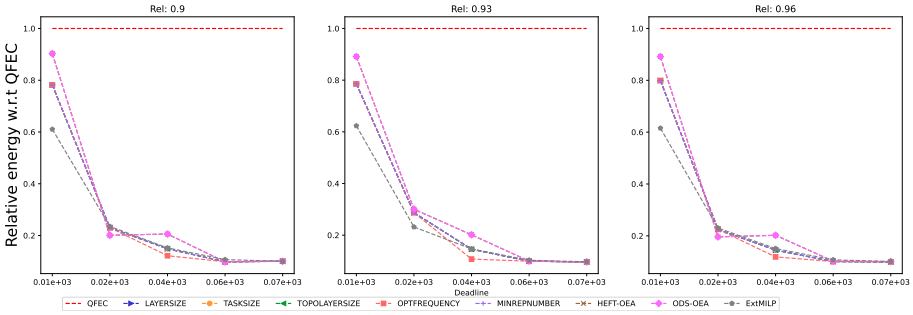

Figure 2115: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

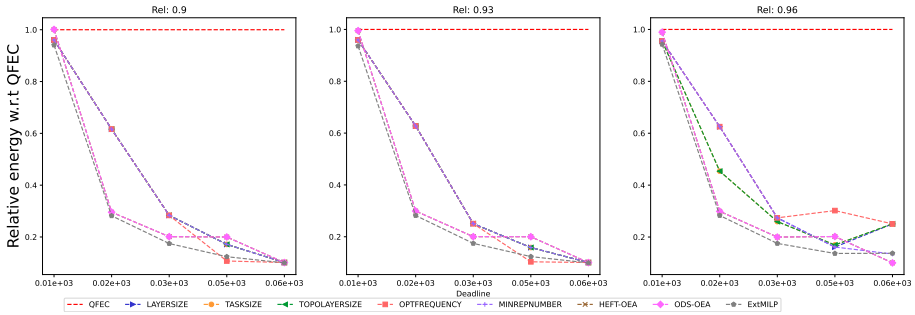

Figure 2116: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

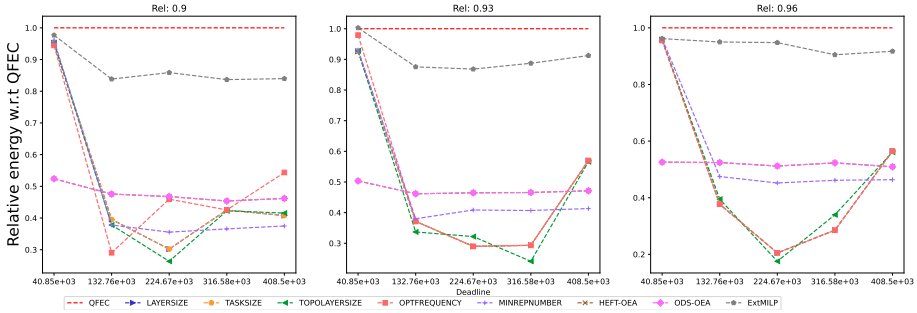

Figure 2117: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

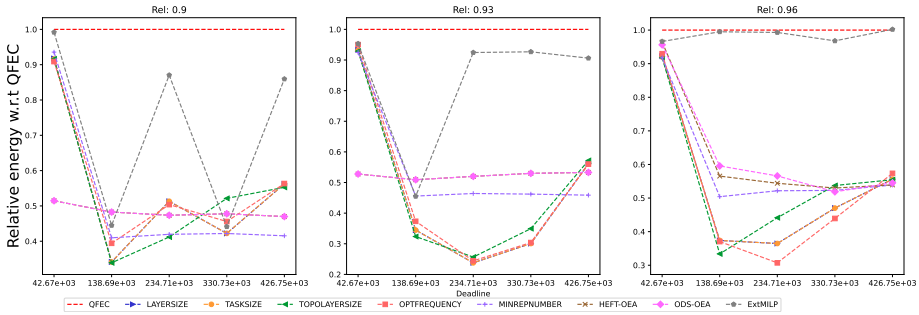

Figure 2118: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

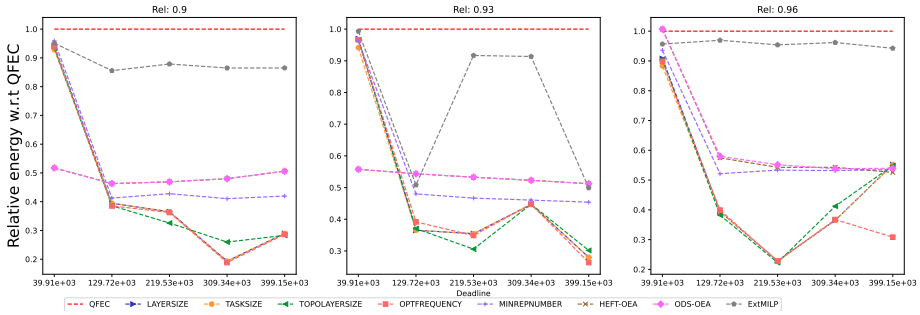

Figure 2119: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

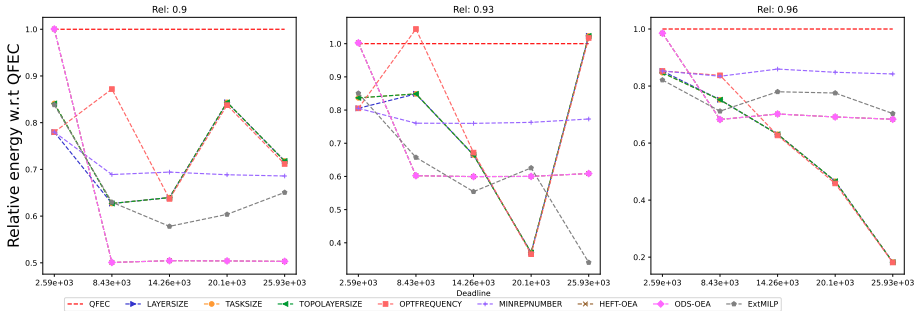

Figure 2120: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

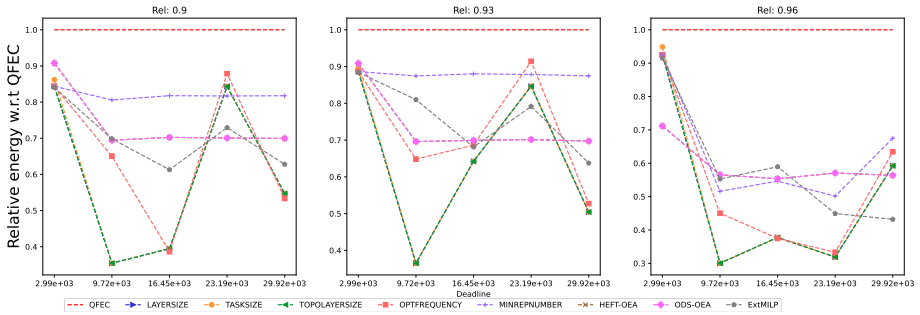

Figure 2121: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.8.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

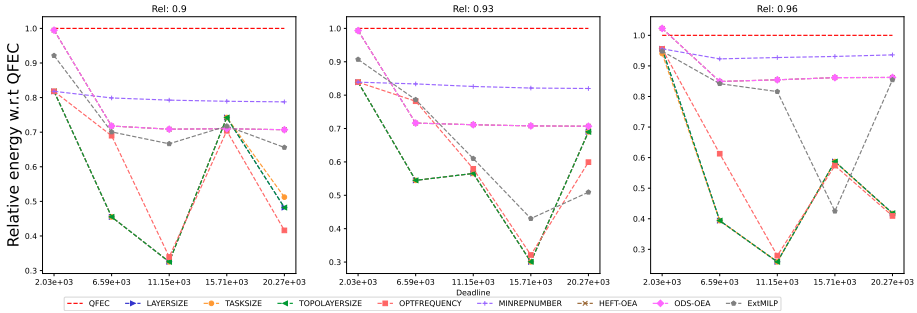

Figure 2122: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

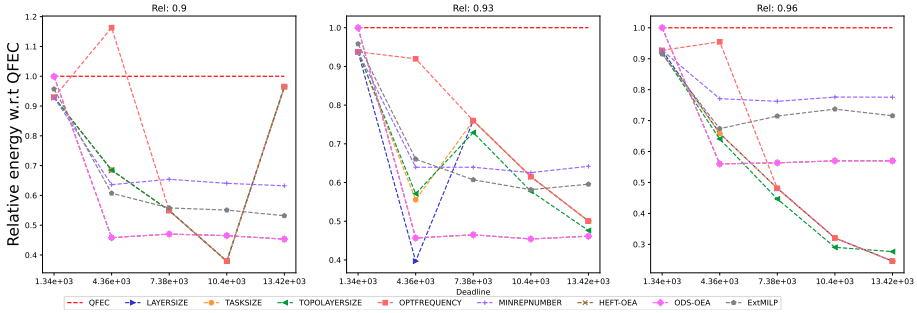

Figure 2123: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

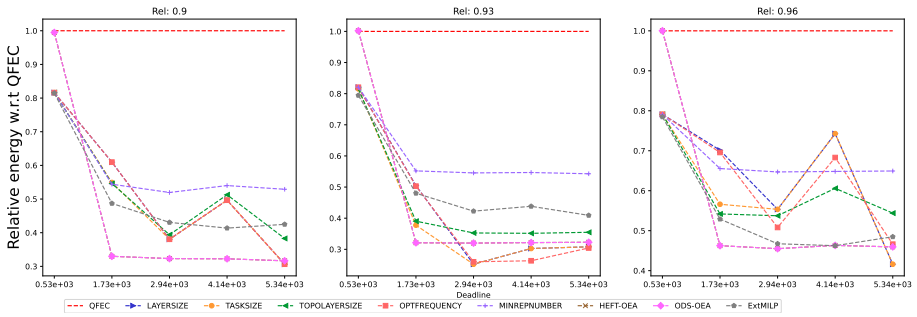

Figure 2124: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

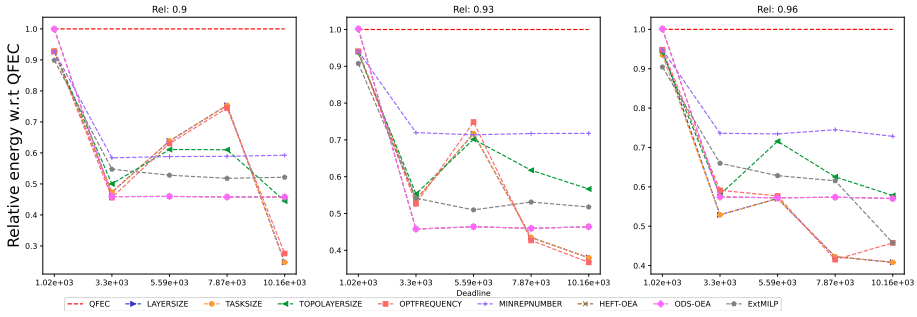

Figure 2125: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

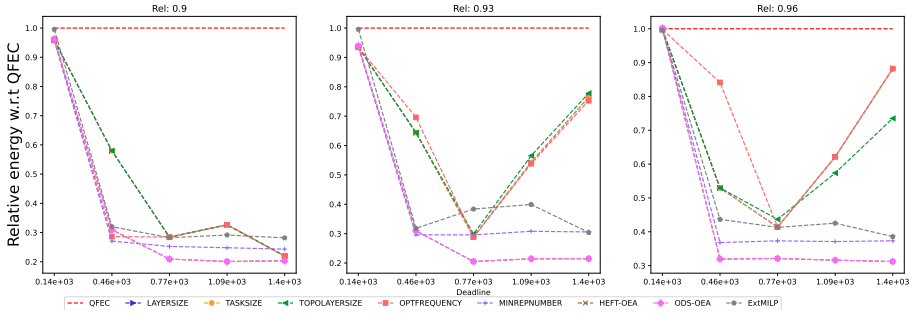

Figure 2126: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

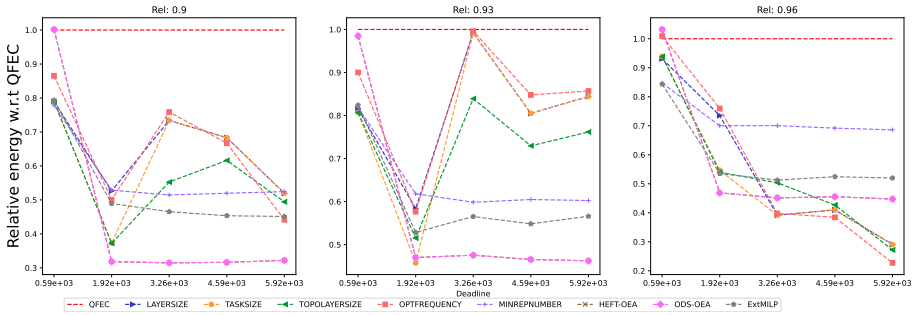

Figure 2127: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

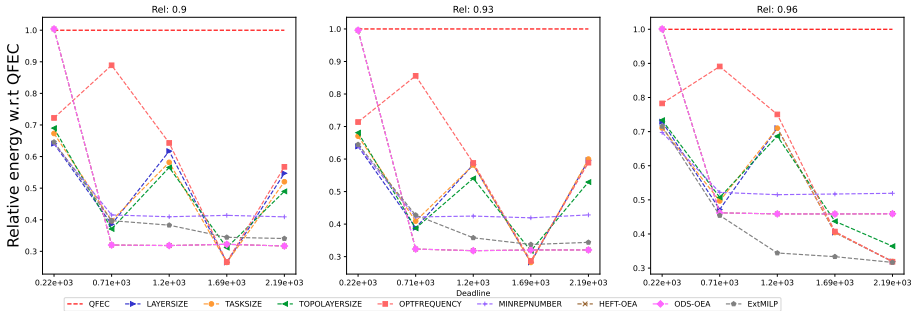

Figure 2128: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

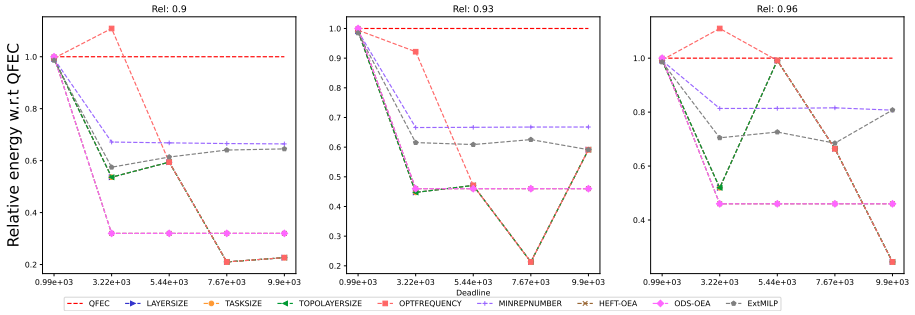

Figure 2129: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

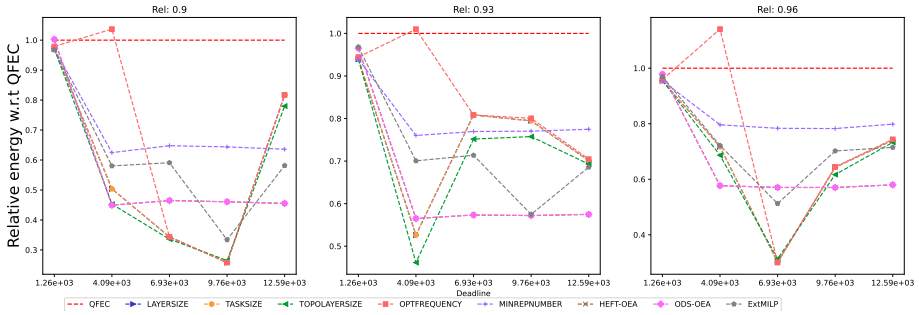

Figure 2130: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

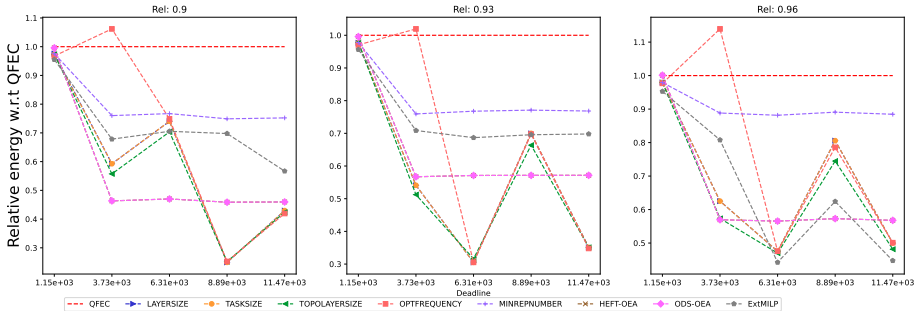

Figure 2131: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

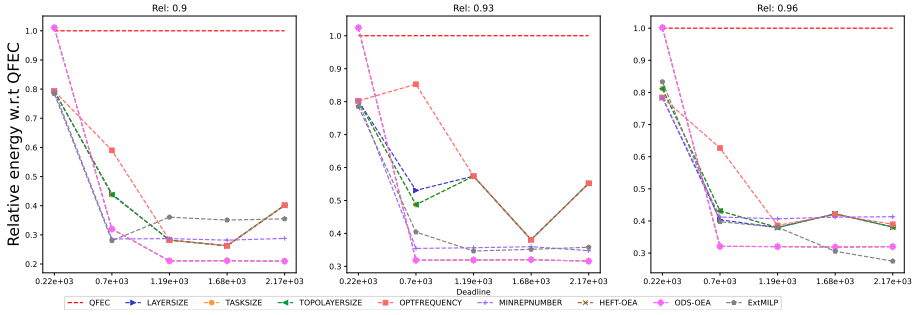

Figure 2132: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

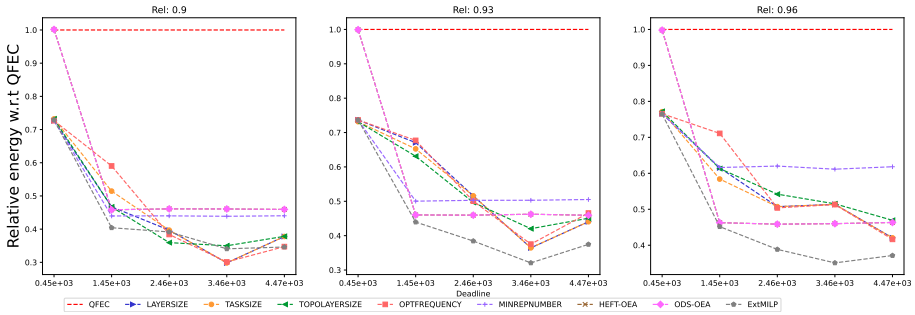

Figure 2133: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

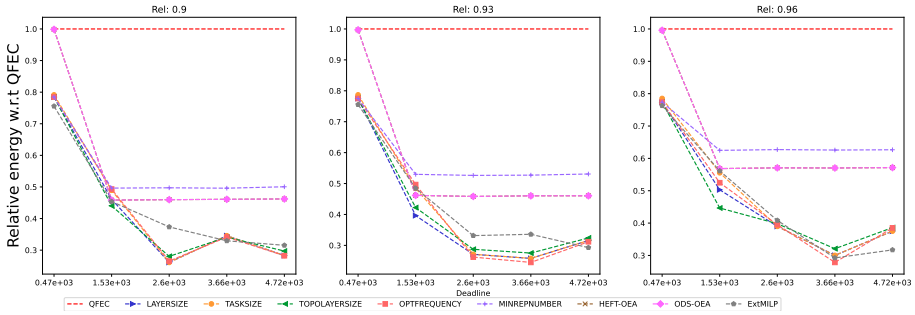

Figure 2134: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

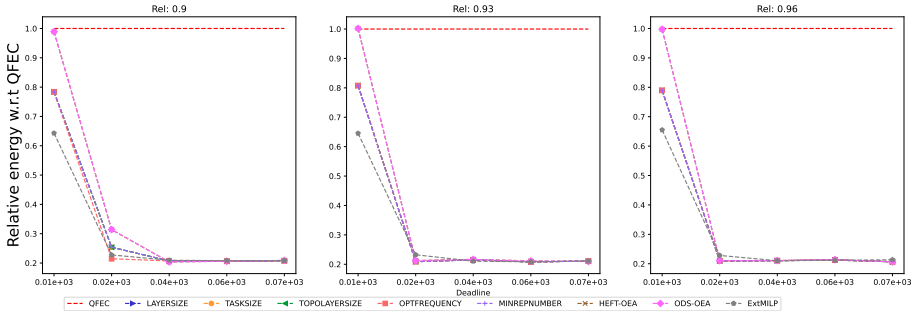

Figure 2135: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

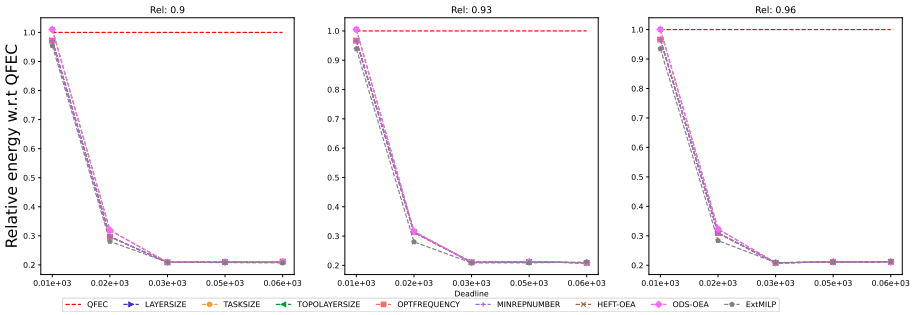

Figure 2136: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

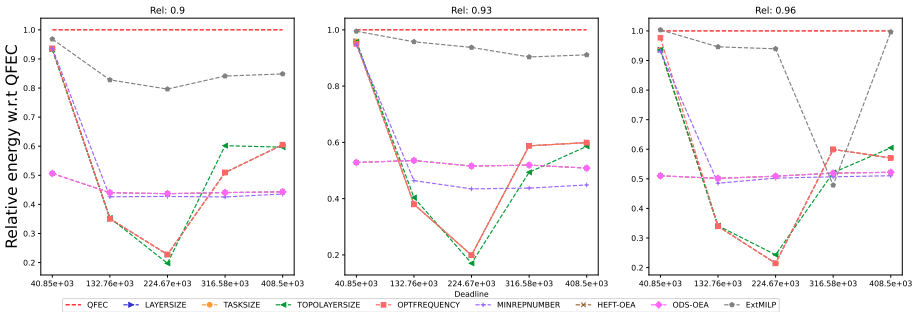

Figure 2137: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

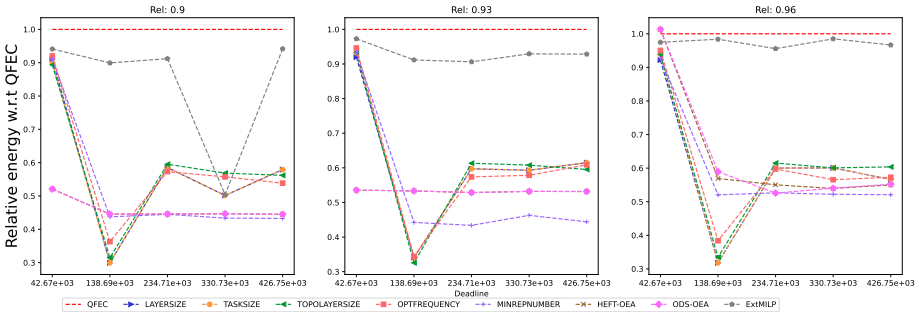

Figure 2138: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

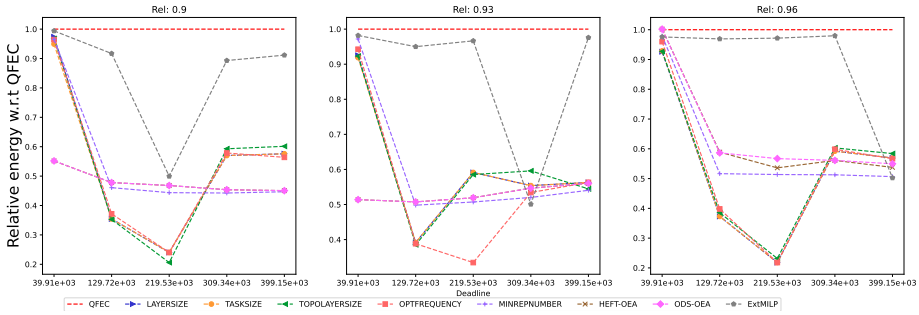

Figure 2139: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

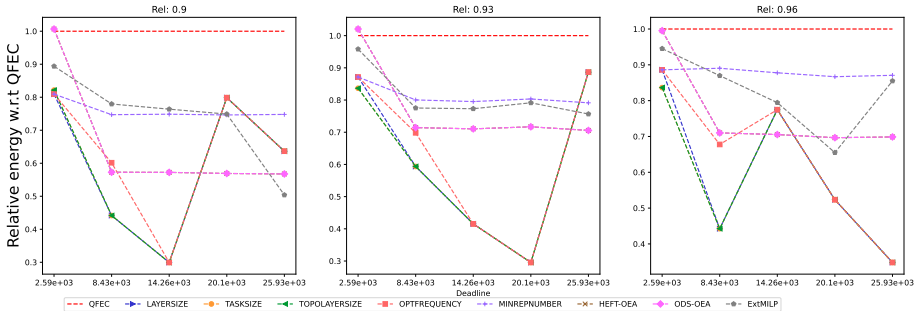

Figure 2140: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

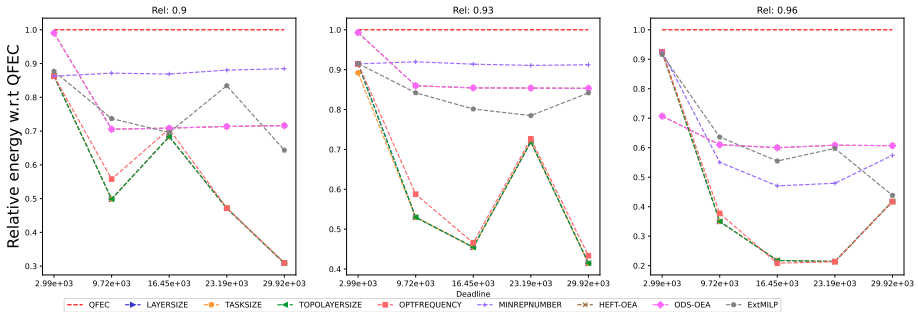

Figure 2141: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.8.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

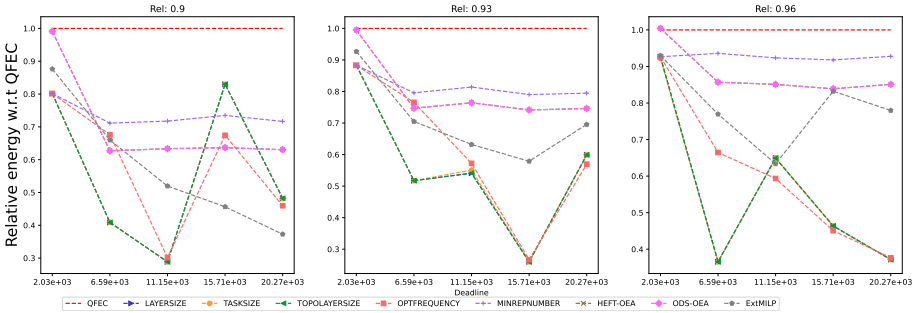

Figure 2142: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

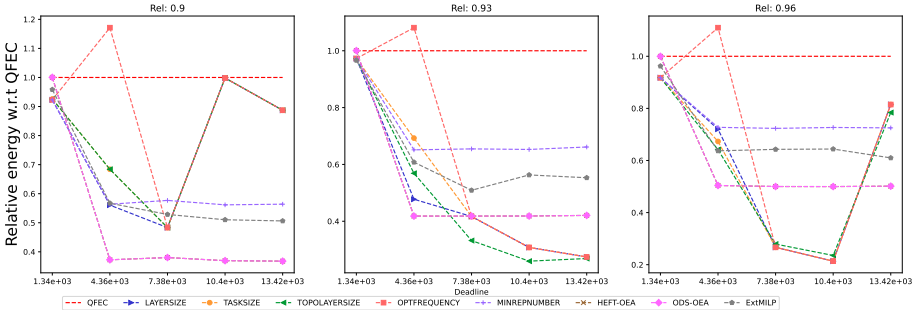

Figure 2143: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

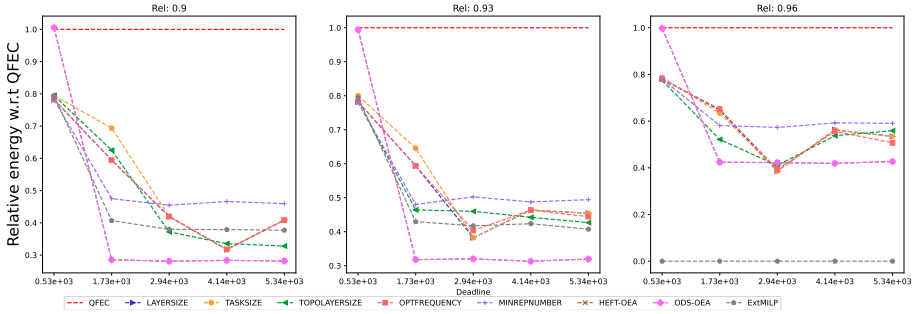

Figure 2144: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

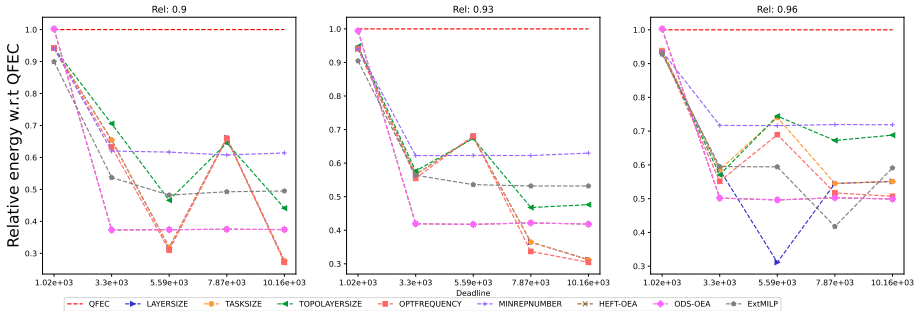

Figure 2145: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

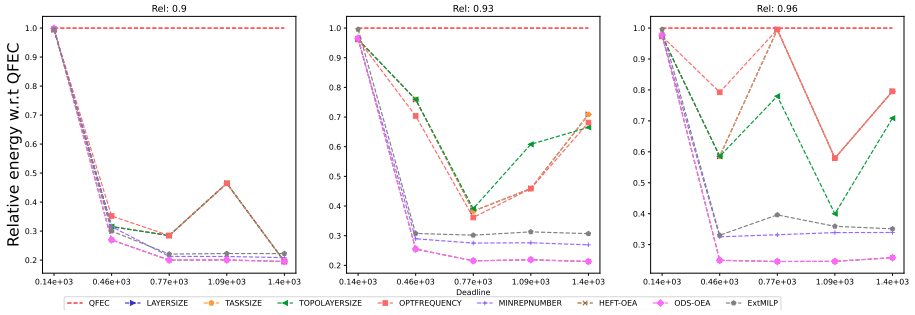

Figure 2146: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

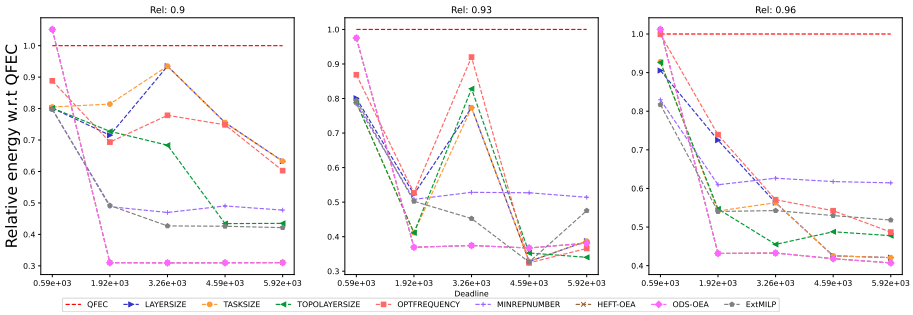

Figure 2147: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

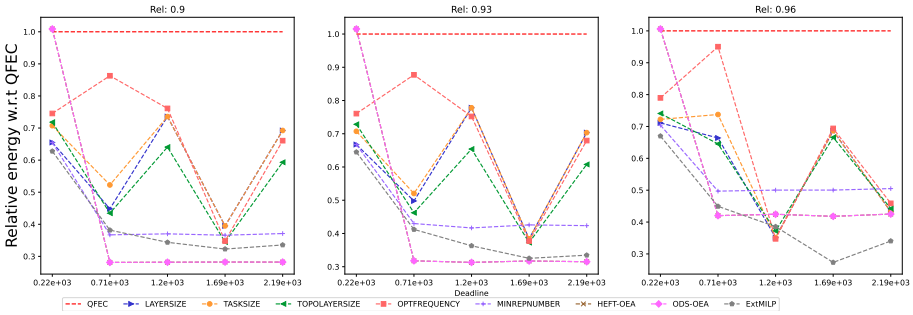

Figure 2148: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

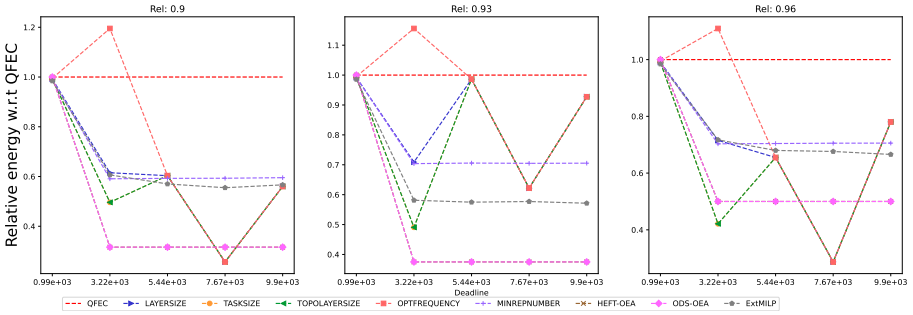

Figure 2149: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

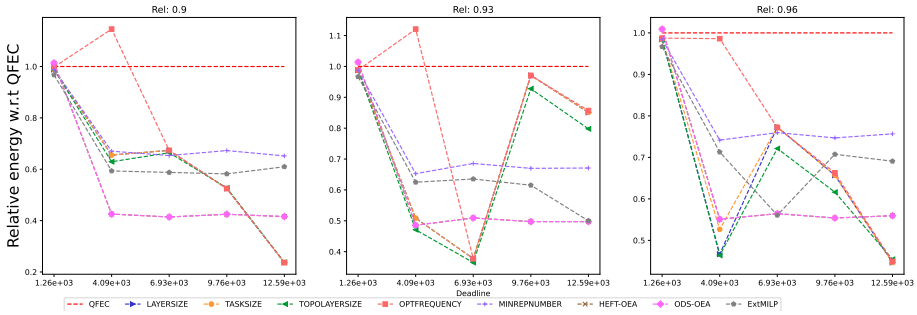

Figure 2150: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

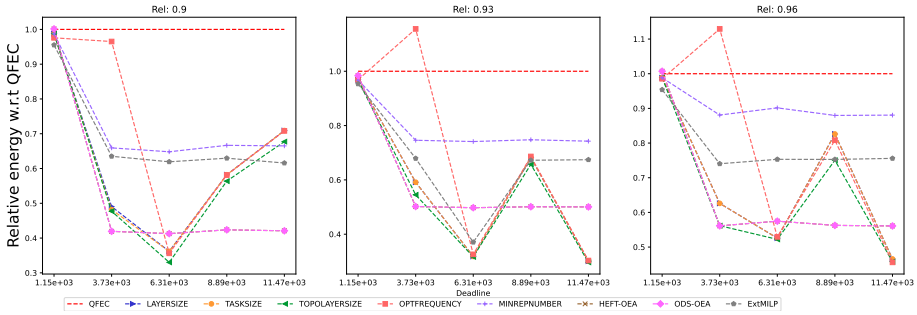

Figure 2151: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

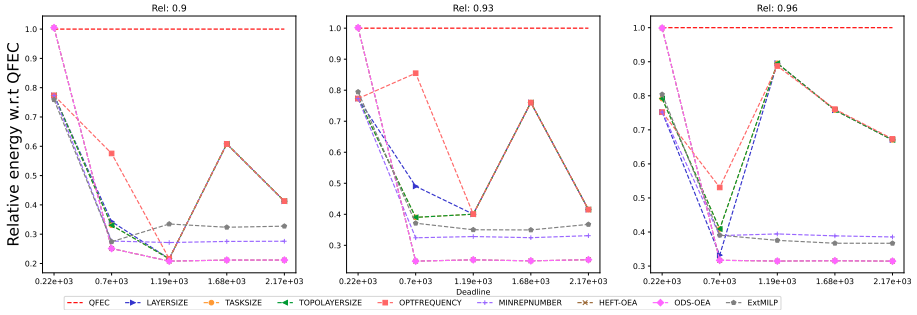

Figure 2152: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

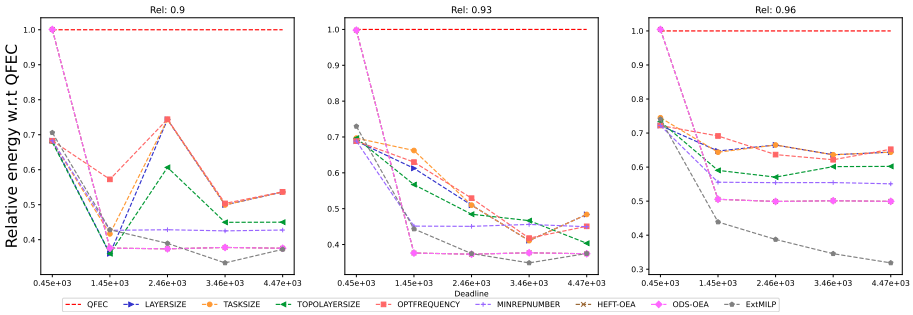

Figure 2153: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

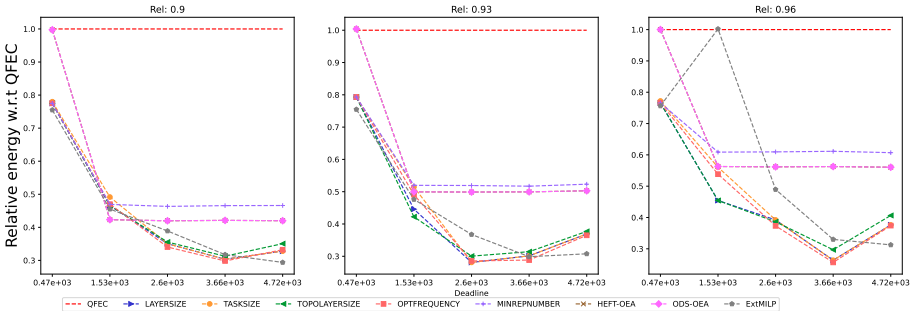

Figure 2154: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

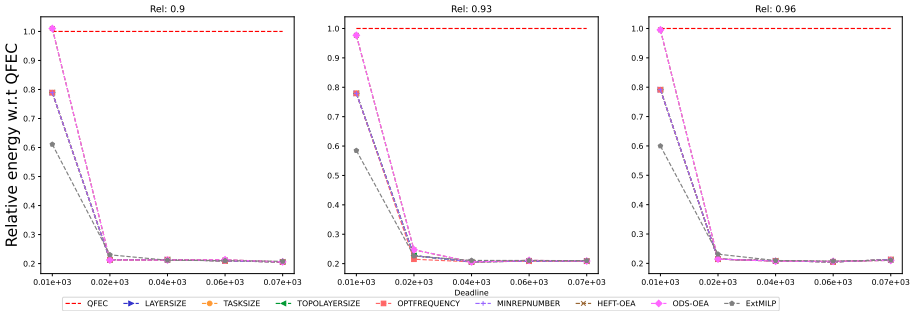

Figure 2155: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

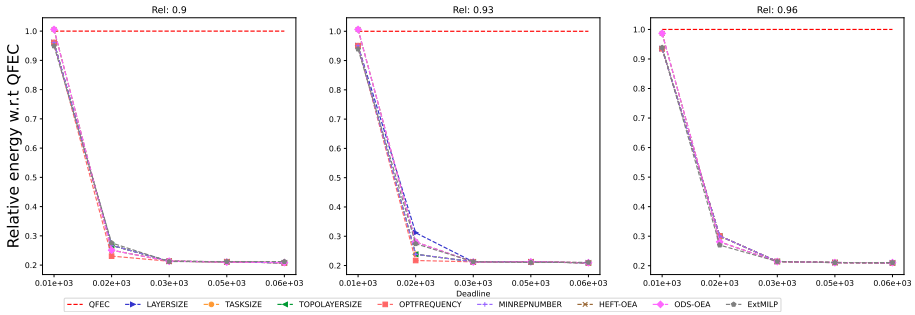

Figure 2156: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

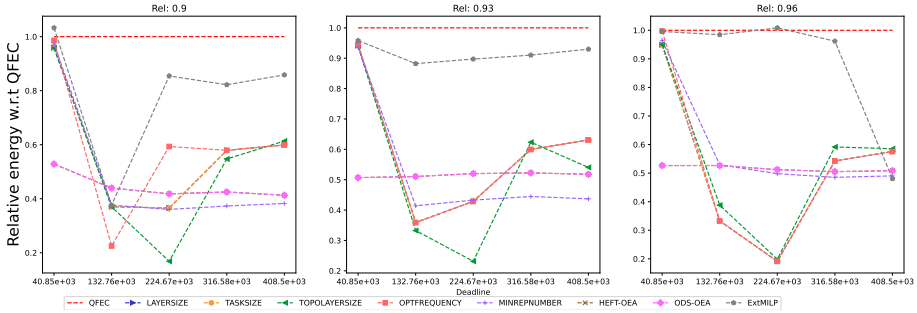

Figure 2157: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

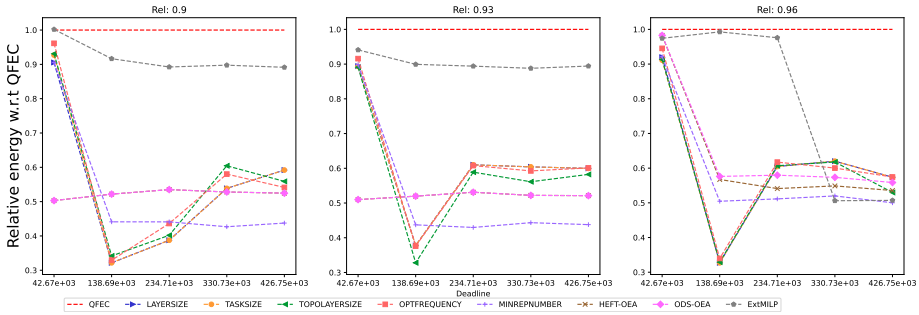

Figure 2158: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

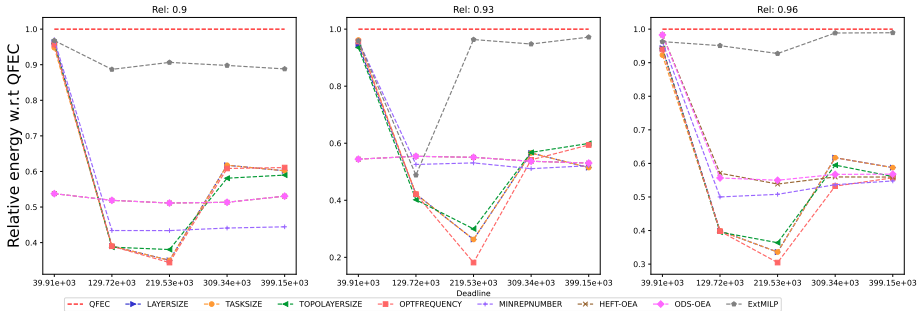

Figure 2159: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

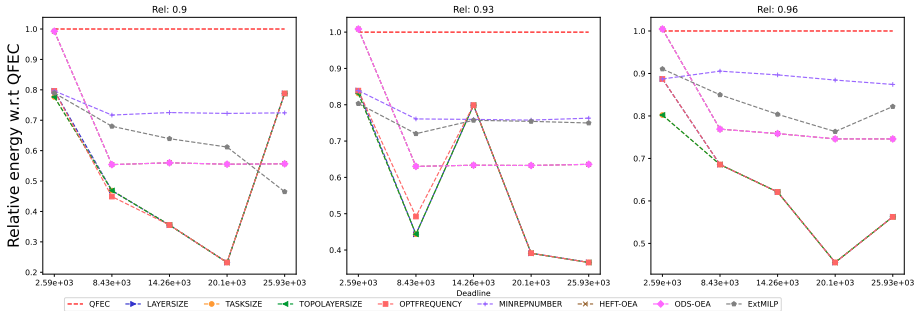

Figure 2160: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

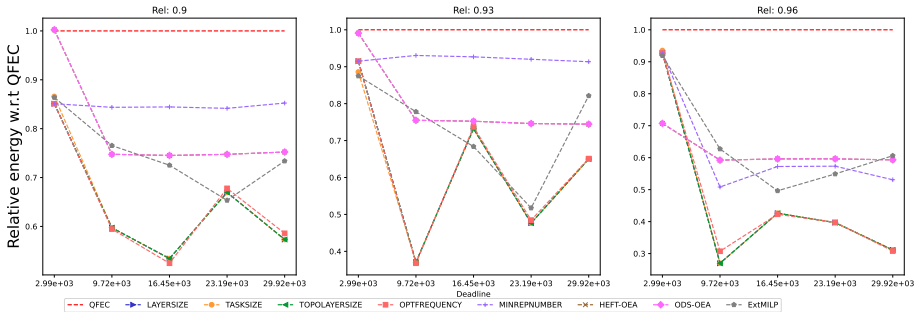

Figure 2161: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.9 $BC/WC = 0.9$

G.9.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

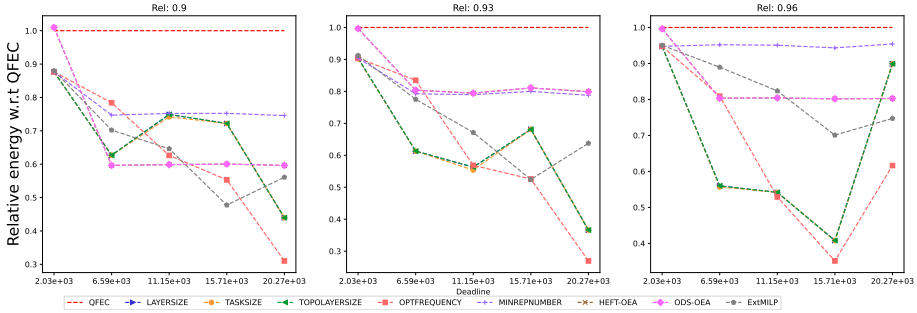

Figure 2162: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

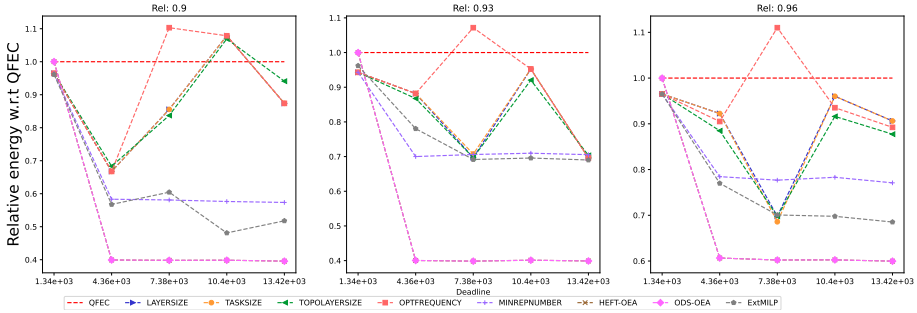

Figure 2163: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

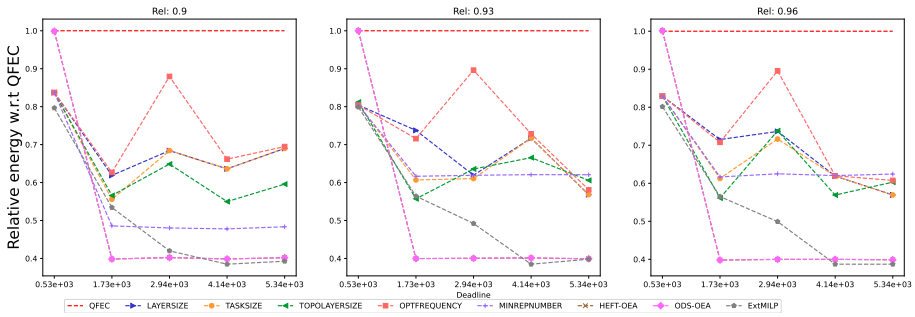

Figure 2164: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

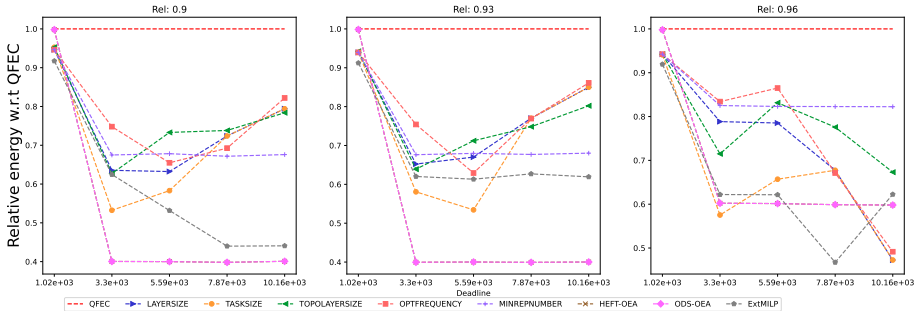

Figure 2165: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

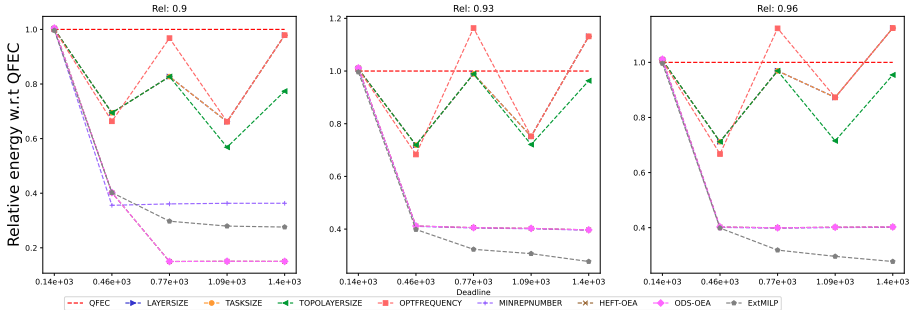

Figure 2166: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

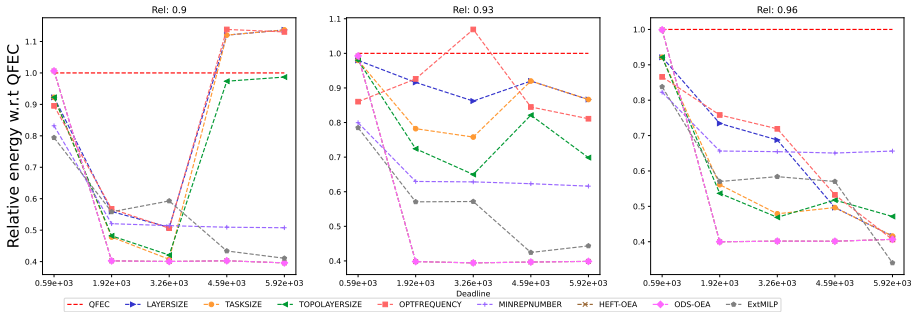

Figure 2167: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

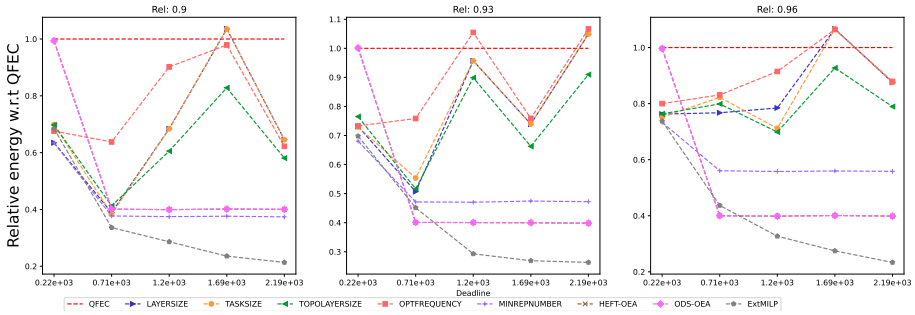

Figure 2168: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

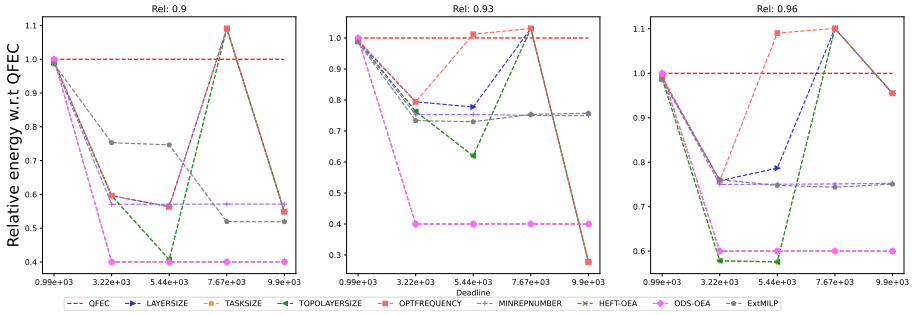

Figure 2169: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

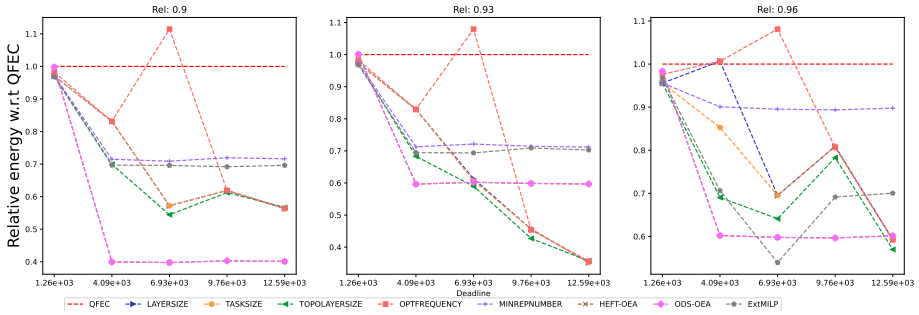

Figure 2170: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

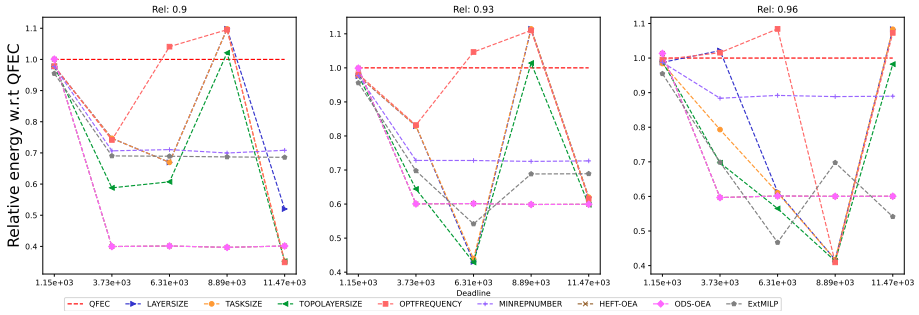

Figure 2171: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

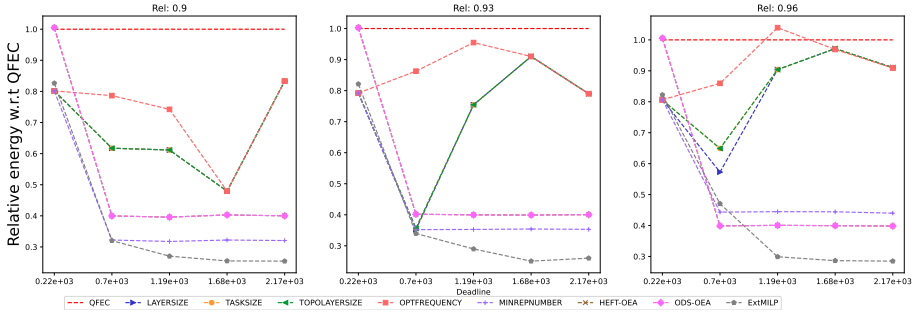

Figure 2172: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

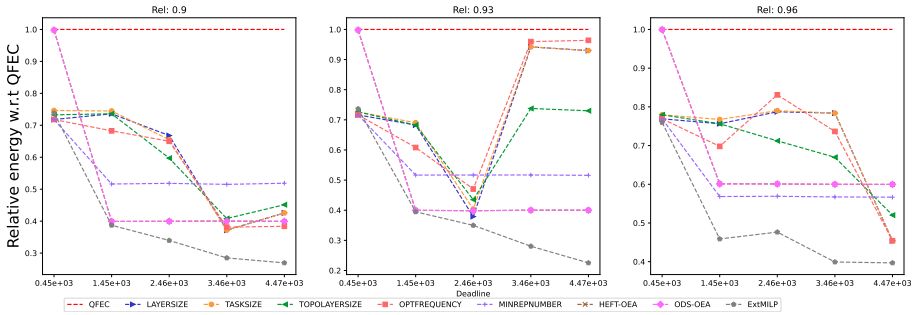

Figure 2173: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

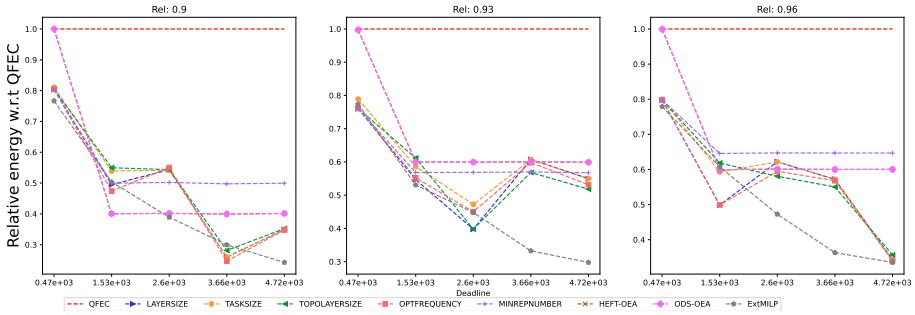

Figure 2174: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

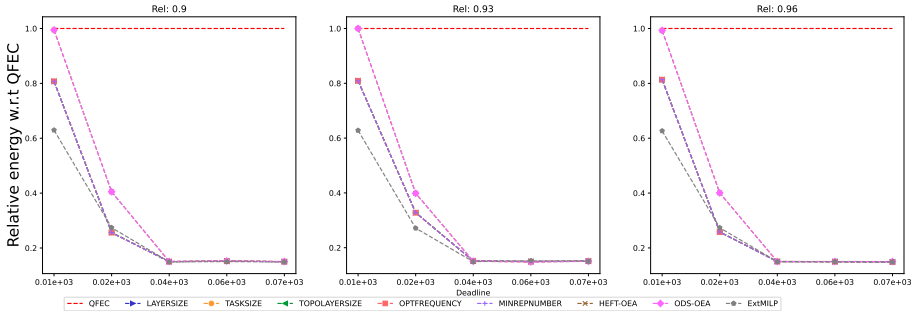

Figure 2175: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

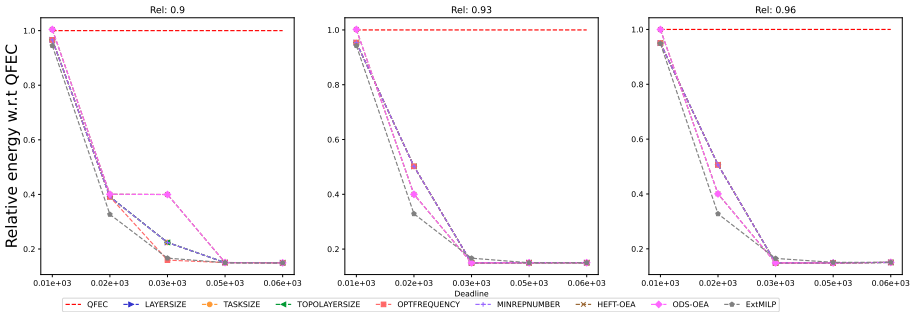

Figure 2176: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

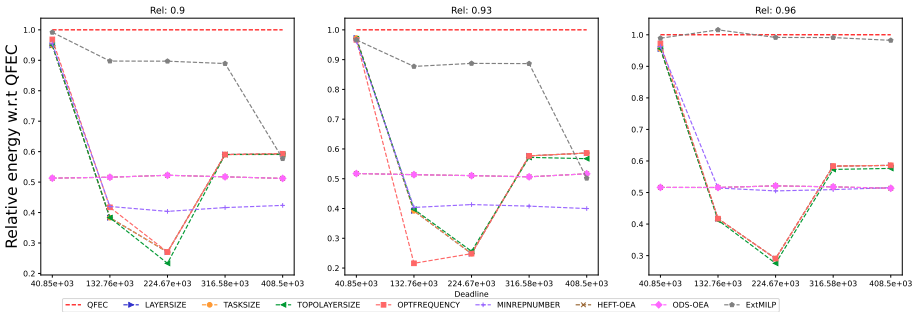

Figure 2177: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

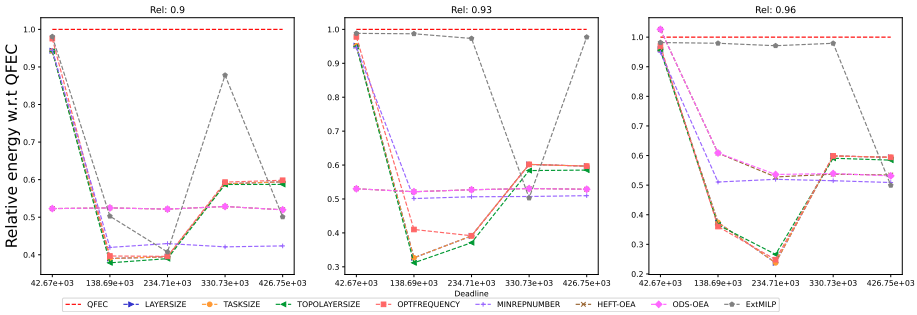

Figure 2178: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

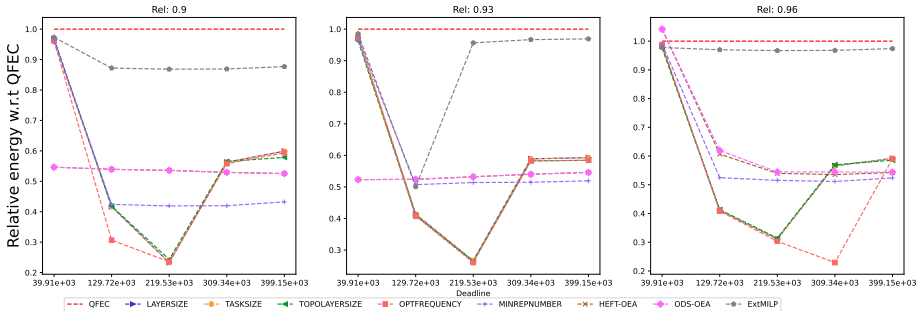

Figure 2179: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

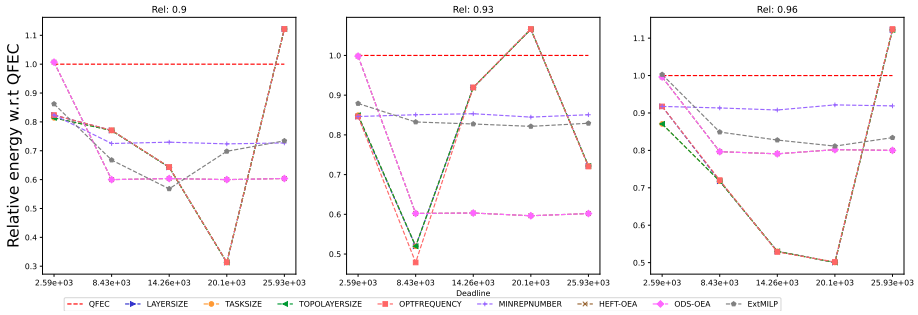

Figure 2180: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

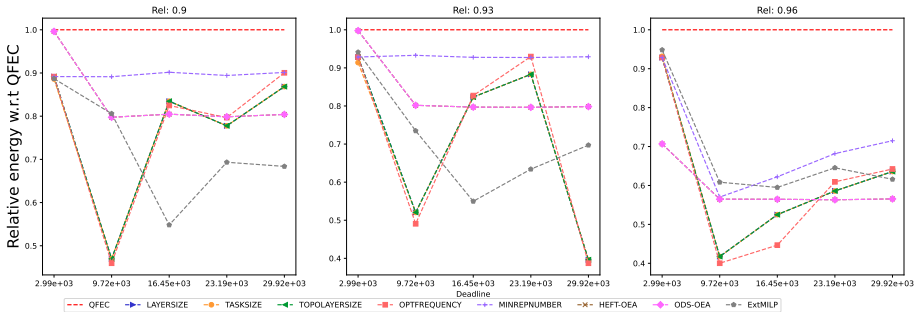

Figure 2181: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.9.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

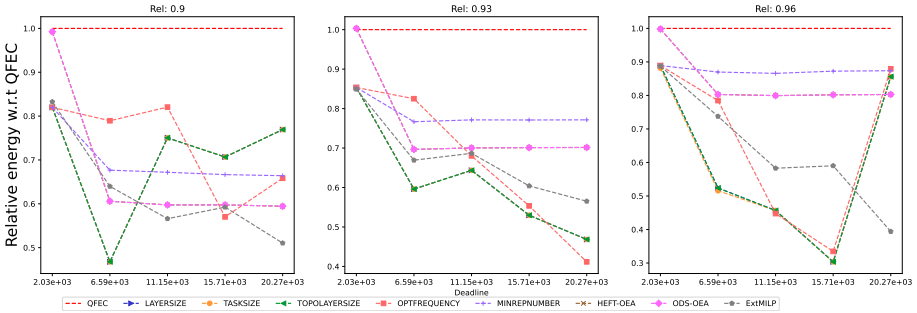

Figure 2182: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

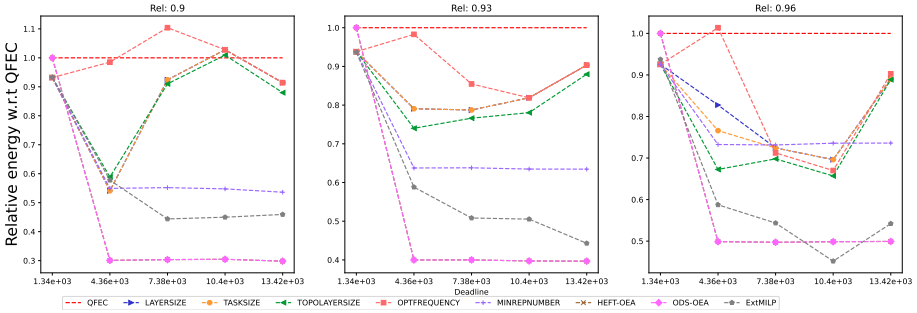

Figure 2183: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

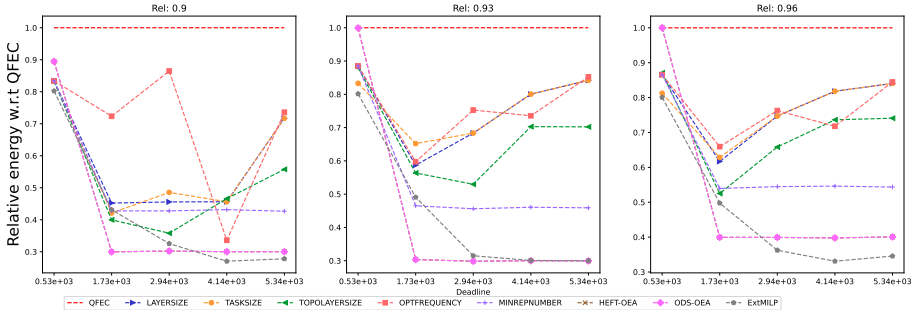

Figure 2184: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

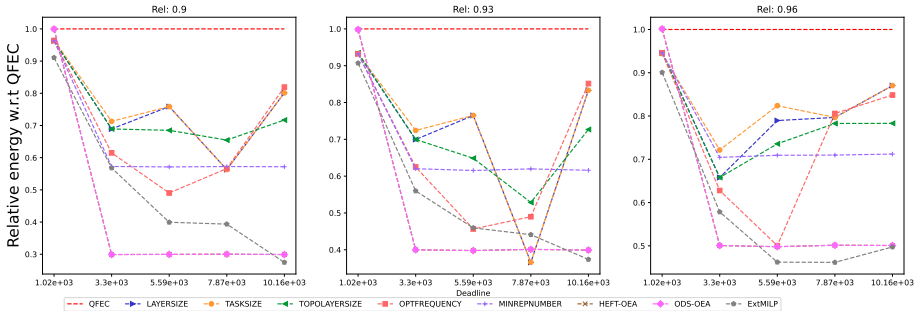

Figure 2185: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

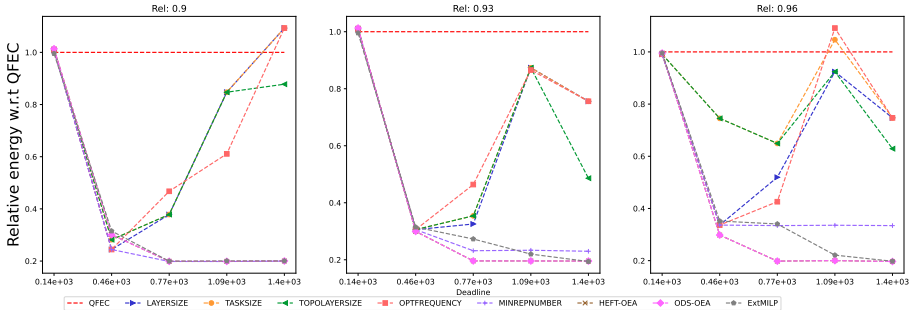

Figure 2186: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

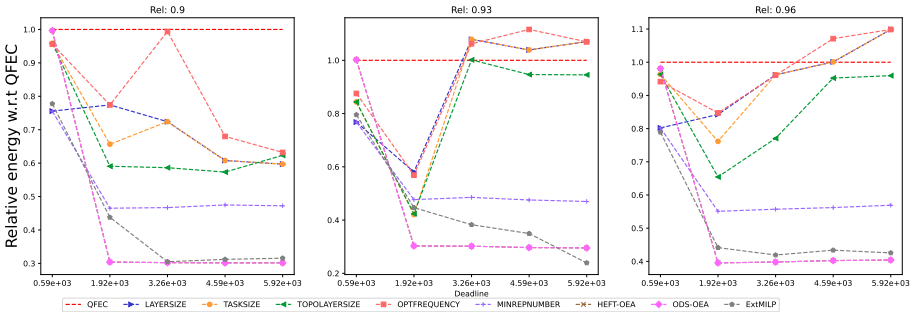

Figure 2187: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

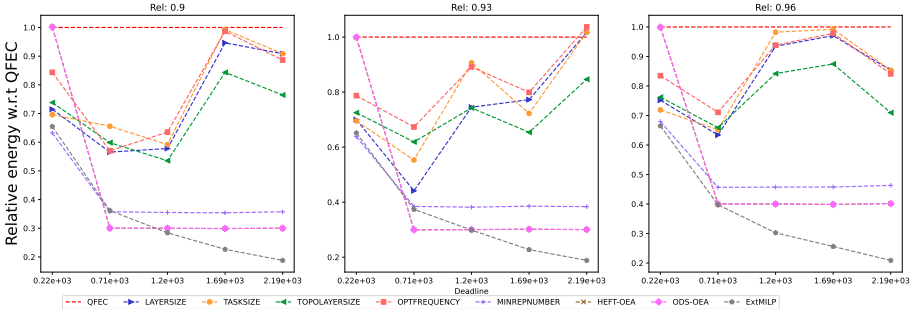

Figure 2188: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

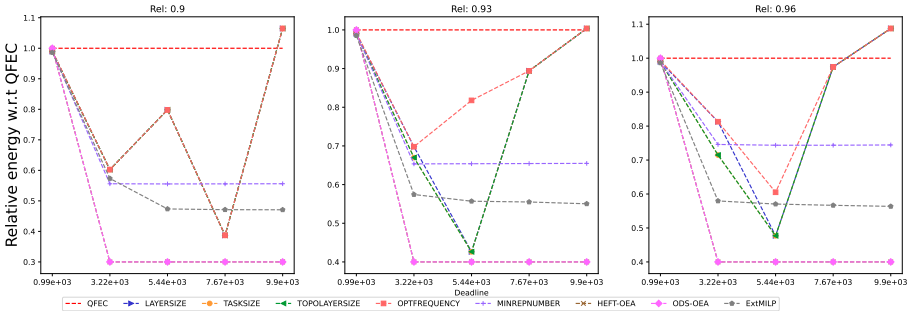

Figure 2189: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

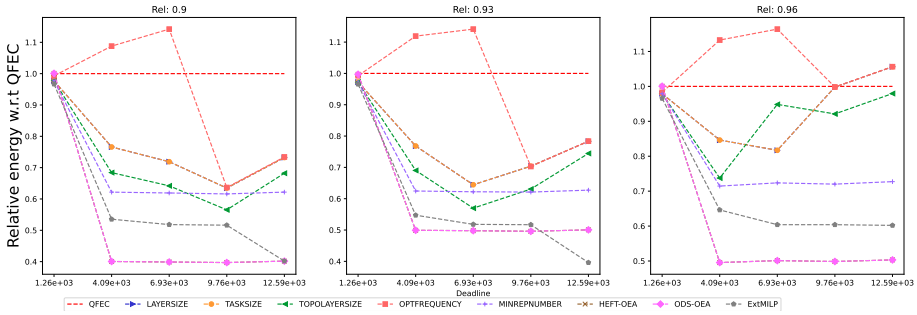

Figure 2190: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

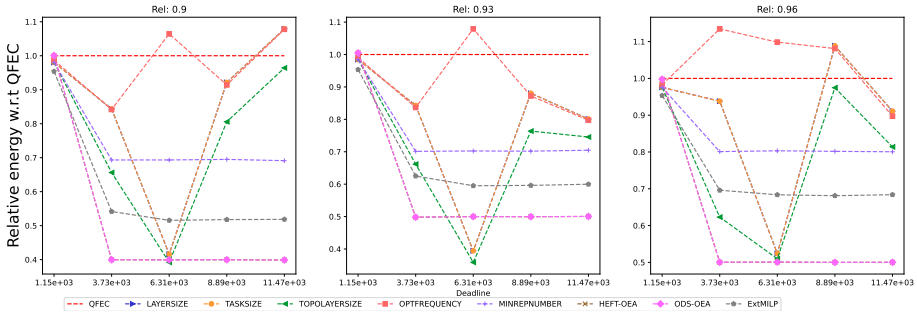

Figure 2191: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

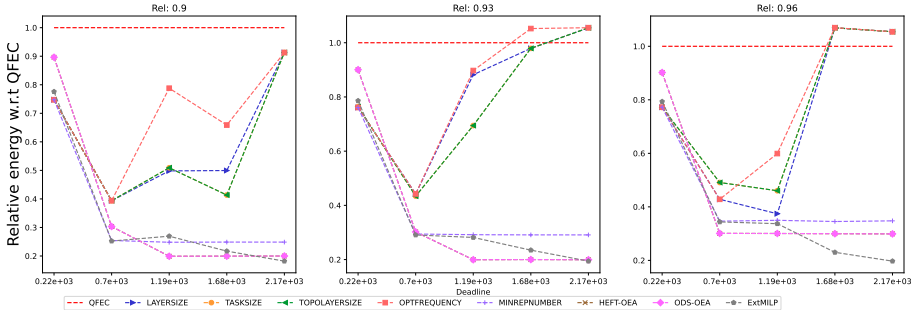

Figure 2192: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

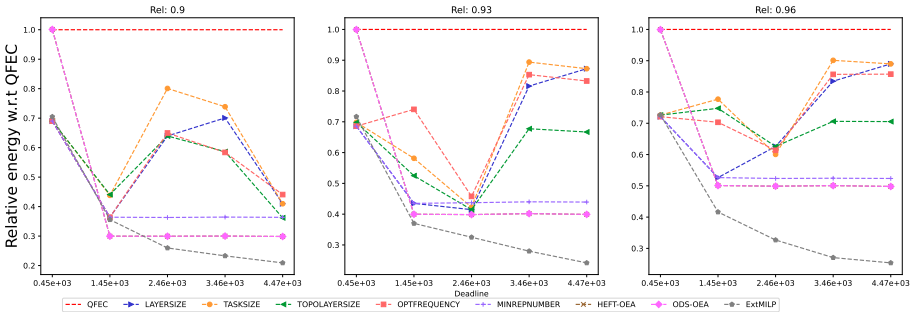

Figure 2193: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

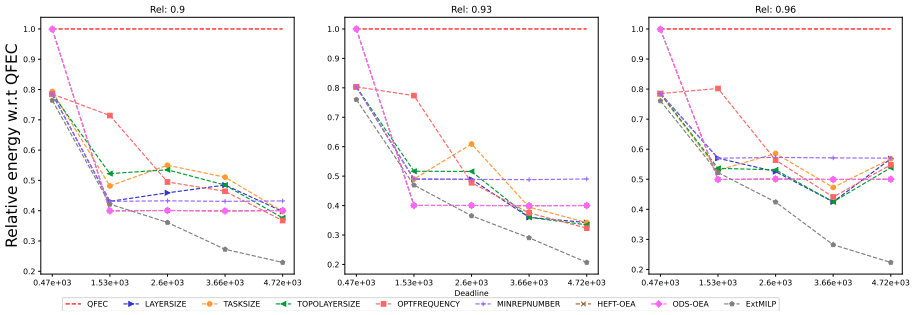

Figure 2194: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

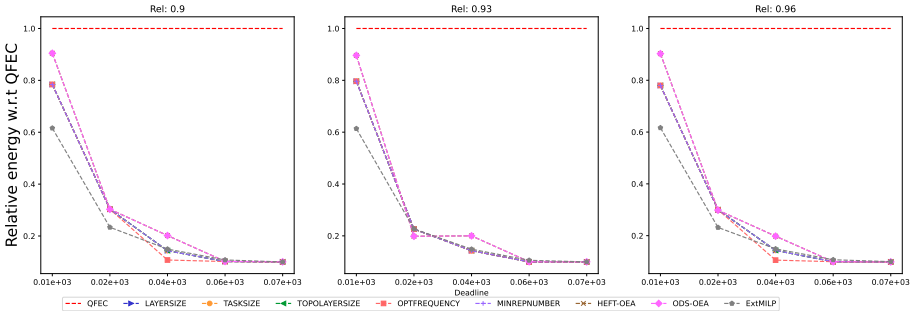

Figure 2195: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

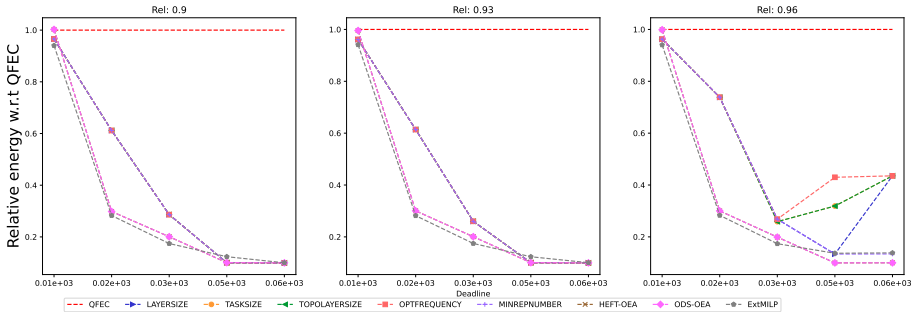

Figure 2196: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

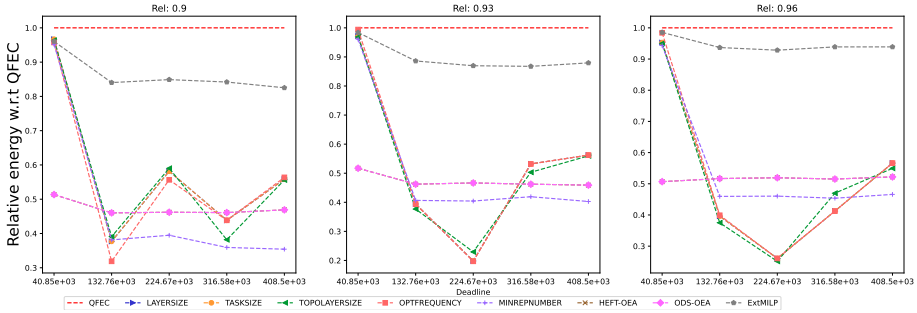

Figure 2197: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

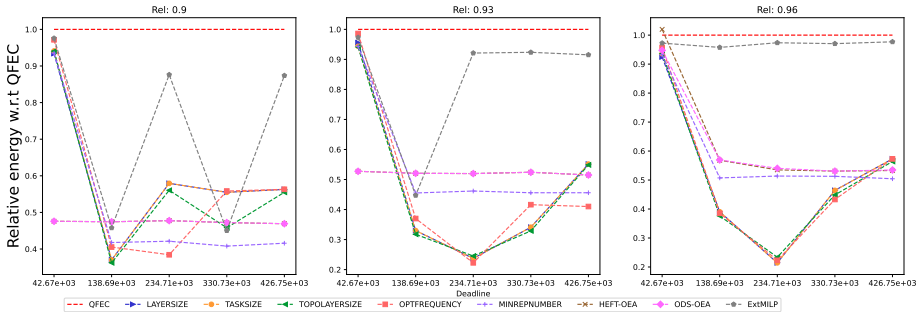

Figure 2198: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

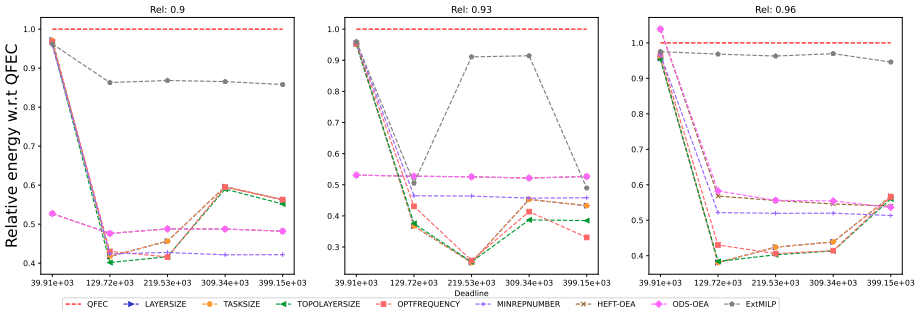

Figure 2199: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

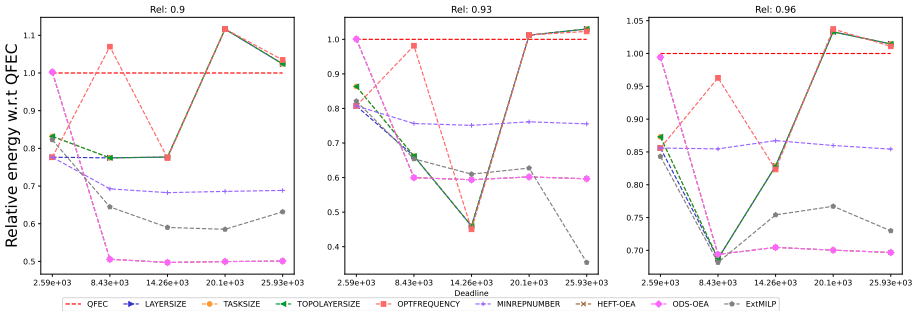

Figure 2200: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

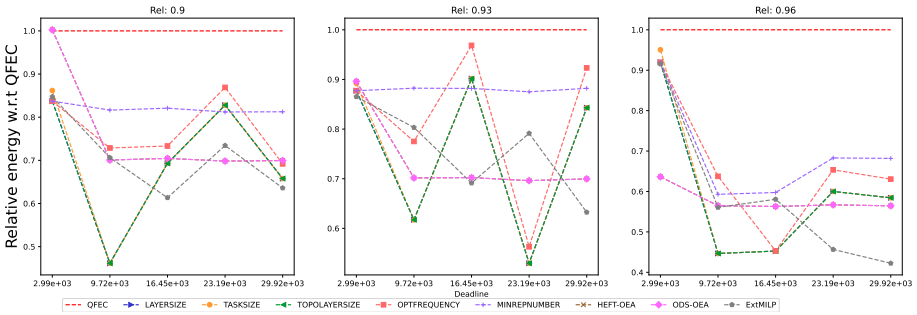

Figure 2201: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.9.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

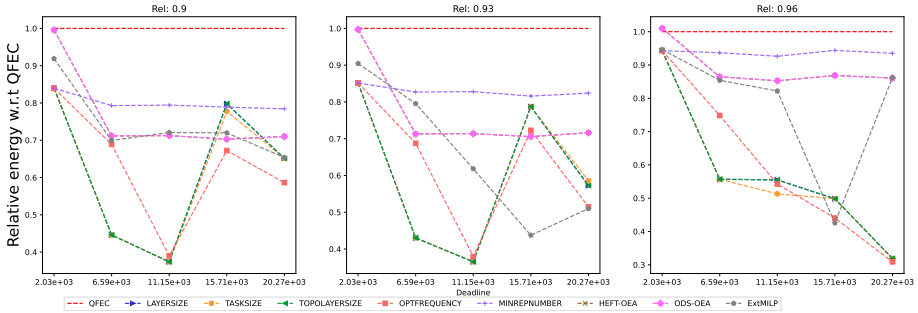

Figure 2202: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

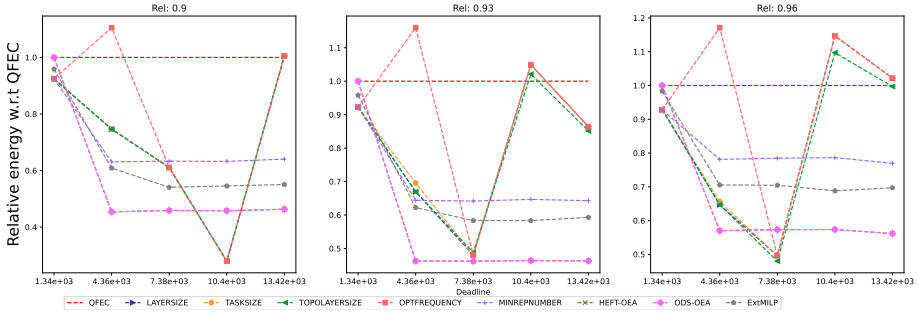

Figure 2203: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

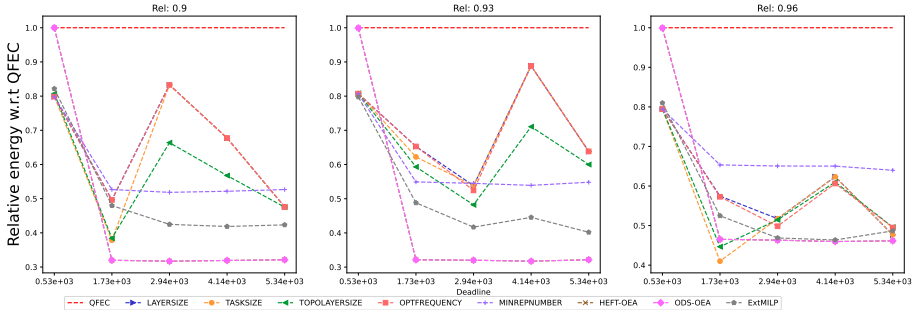

Figure 2204: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

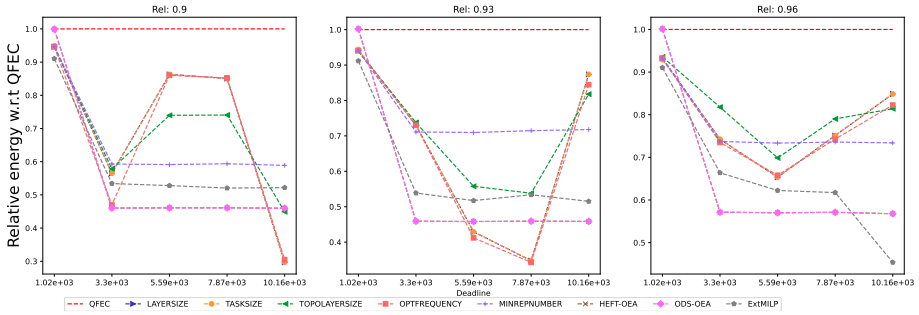

Figure 2205: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

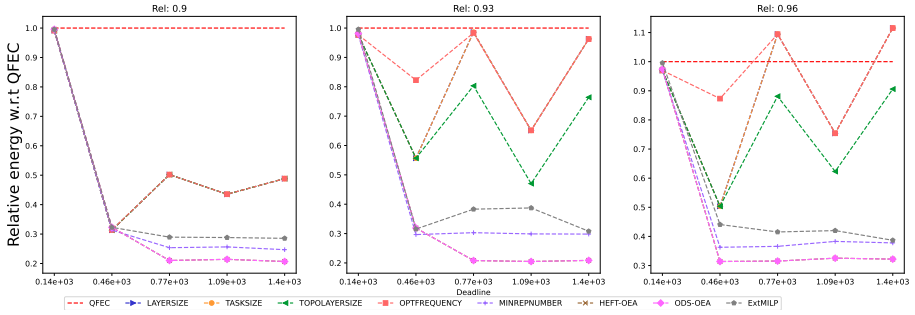

Figure 2206: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

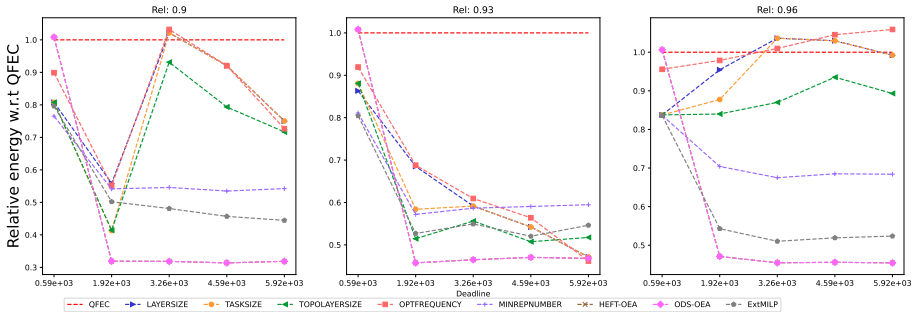

Figure 2207: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

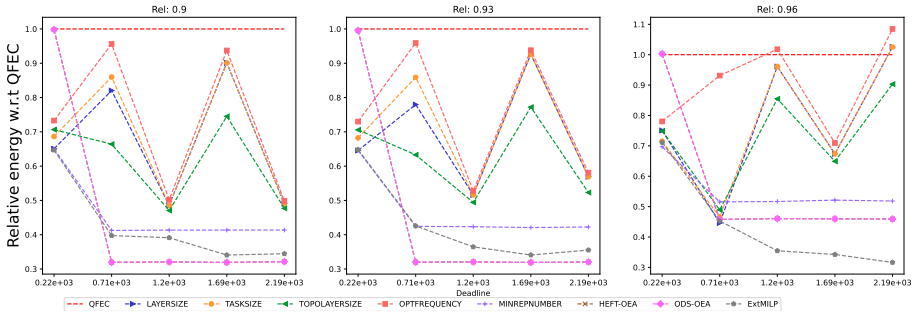

Figure 2208: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

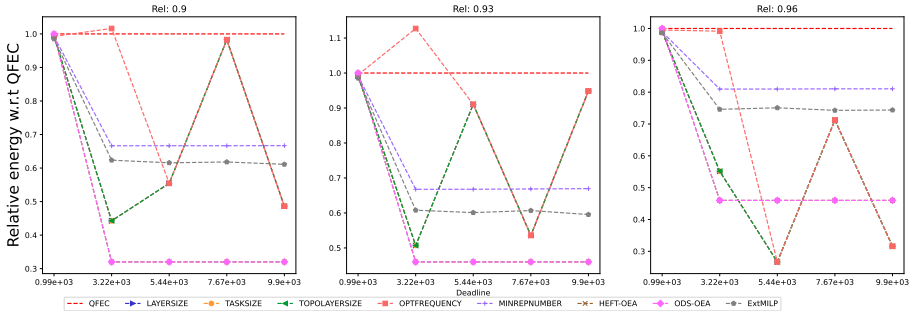

Figure 2209: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

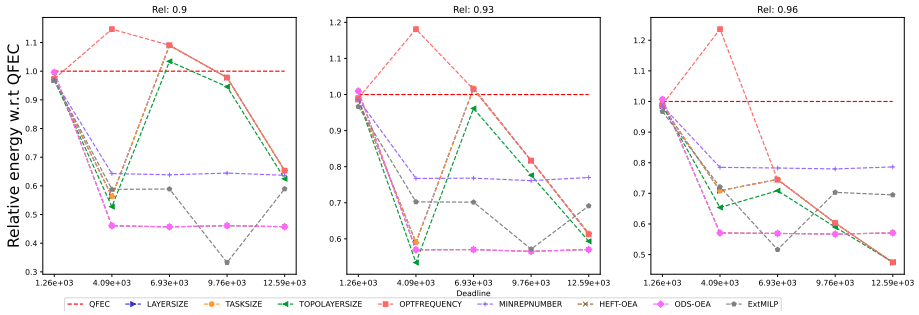

Figure 2210: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

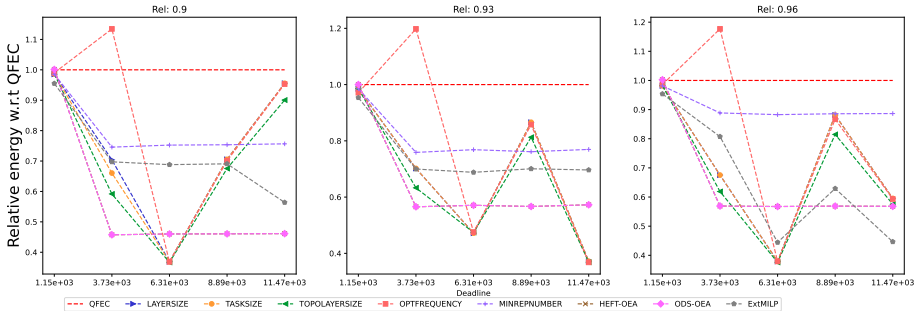

Figure 2211: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

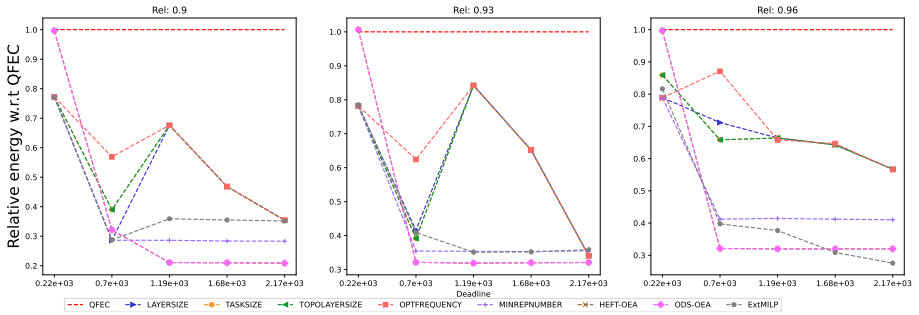

Figure 2212: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

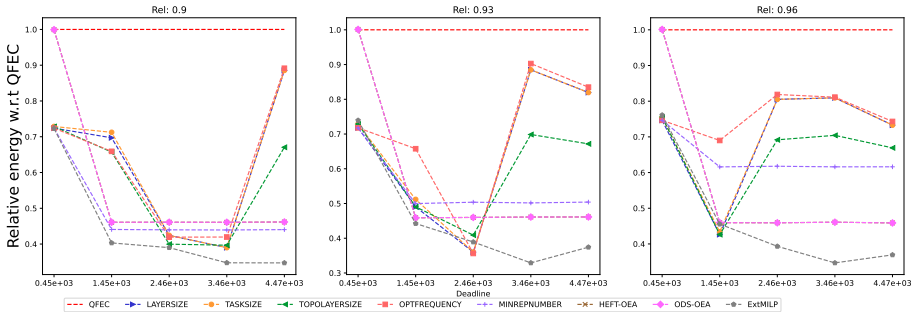

Figure 2213: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

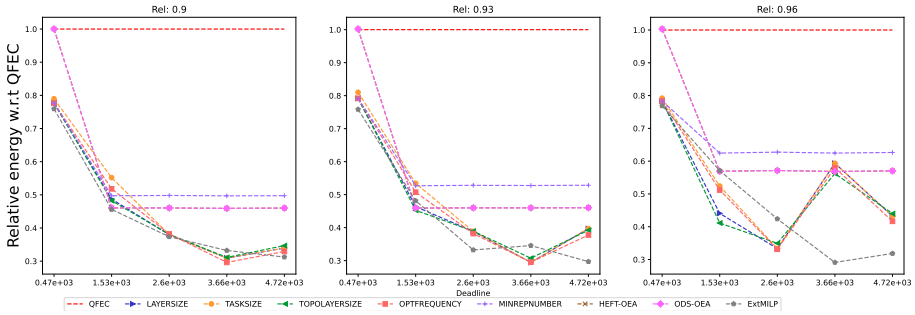

Figure 2214: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

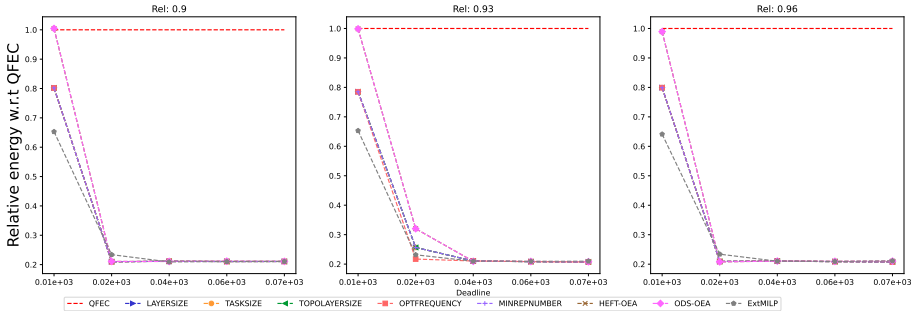

Figure 2215: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

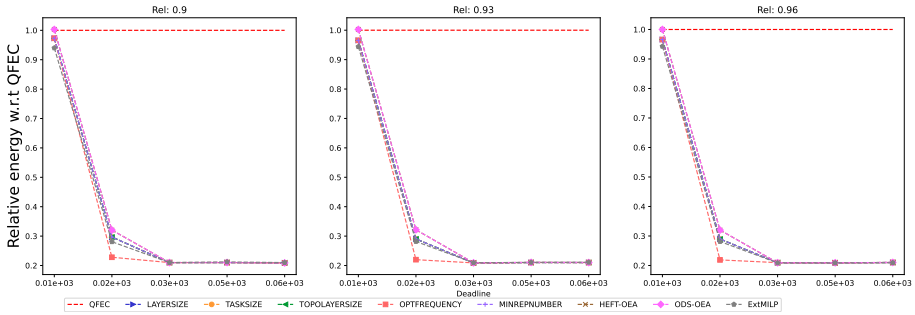

Figure 2216: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

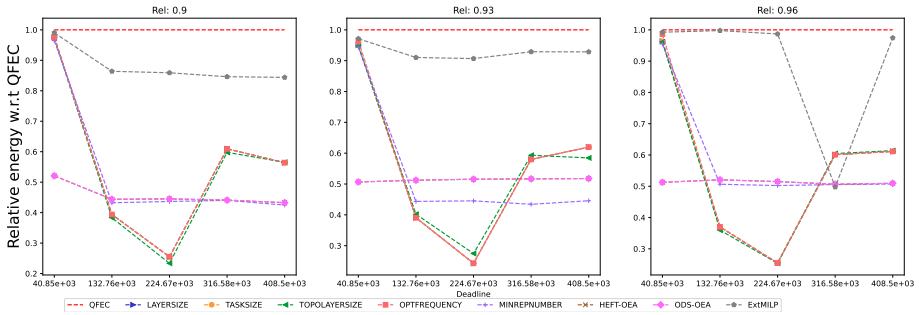

Figure 2217: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

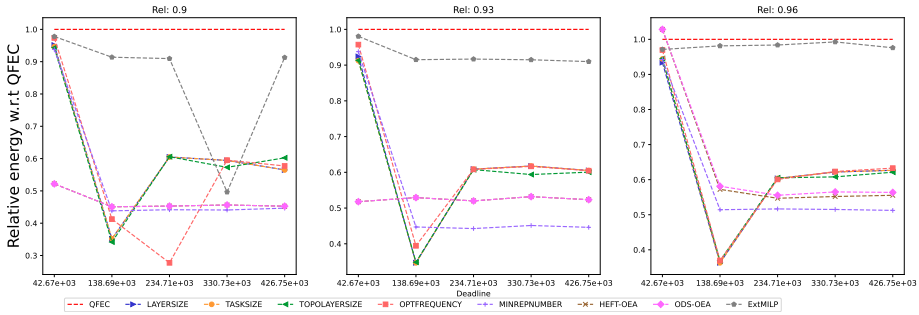

Figure 2218: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

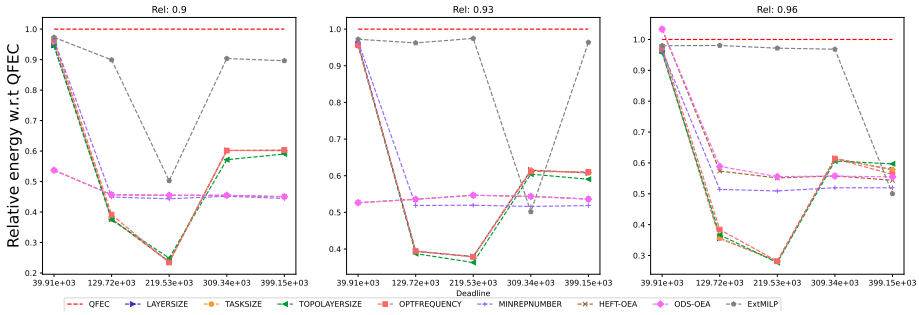

Figure 2219: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

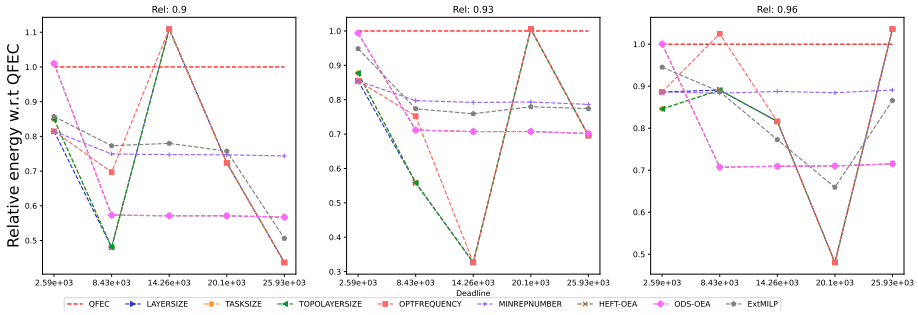

Figure 2220: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

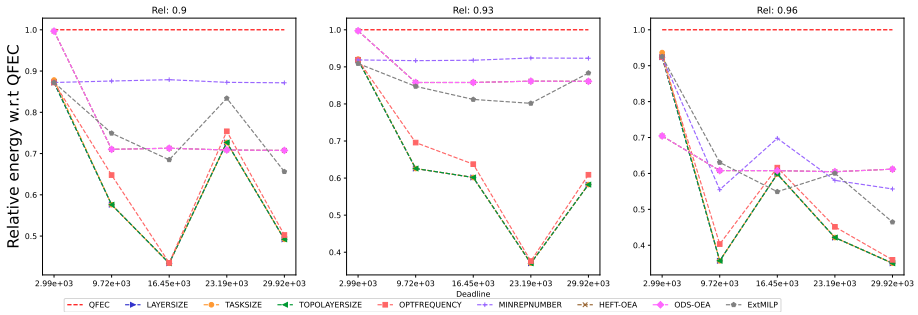

Figure 2221: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

G.9.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

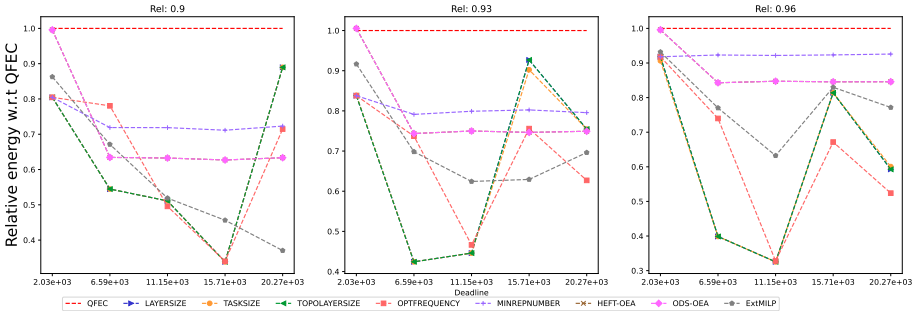

Figure 2222: Assessing the performance of ExtMILP on the BLAST workflow (with 10 tasks).

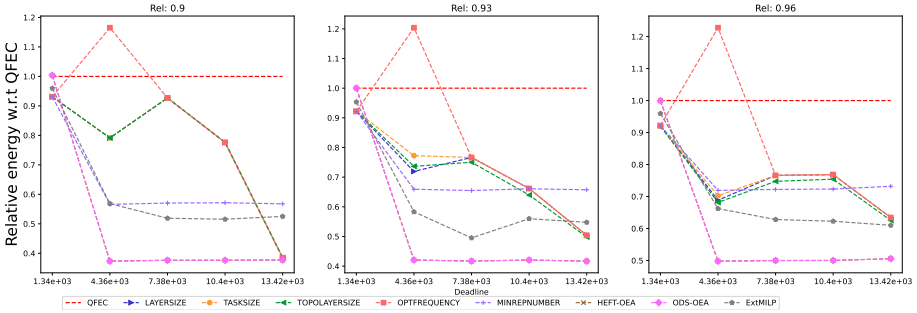

Figure 2223: Assessing the performance of ExtMILP on the BWA workflow (with 10 tasks).

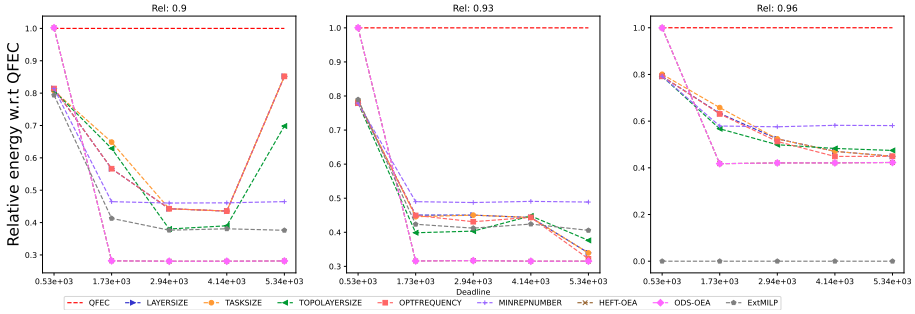

Figure 2224: Assessing the performance of ExtMILP on the BWA workflow (with 20 tasks).

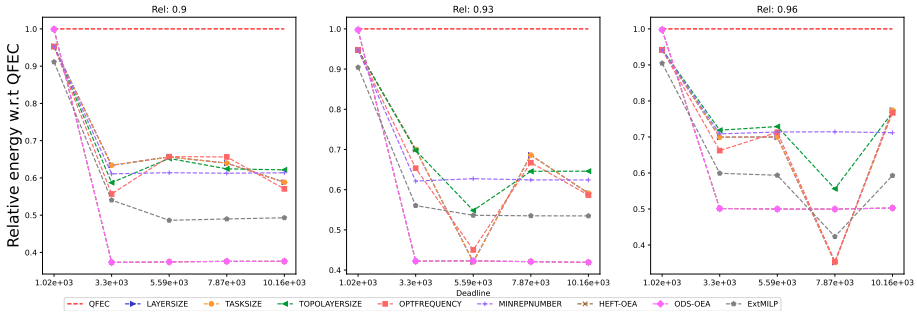

Figure 2225: Assessing the performance of ExtMILP on the BWA workflow (with 30 tasks).

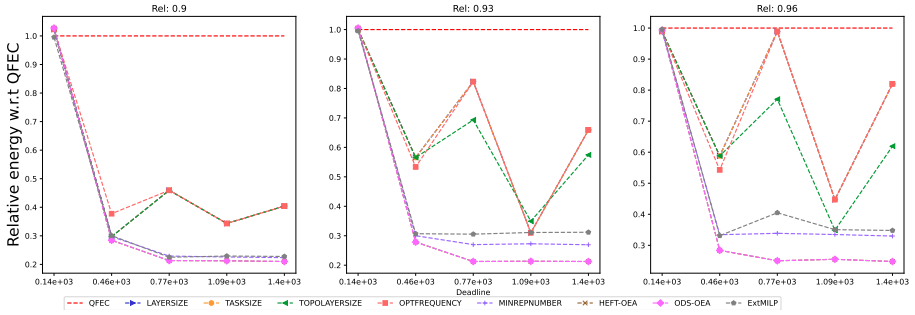

Figure 2226: Assessing the performance of ExtMILP on the Cycles workflow (with 10 tasks).

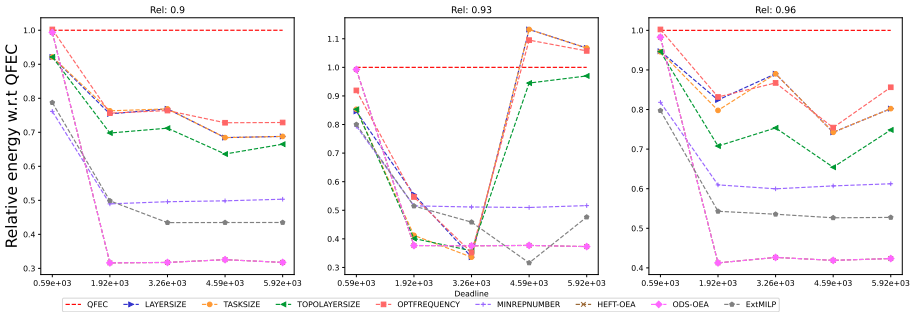

Figure 2227: Assessing the performance of ExtMILP on the Cycles workflow (with 20 tasks).

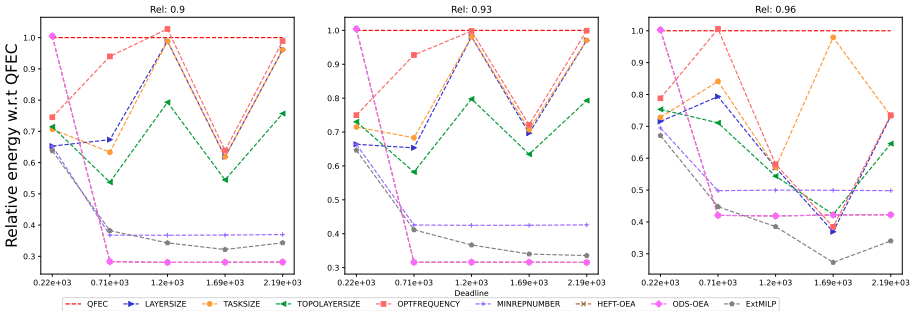

Figure 2228: Assessing the performance of ExtMILP on the Cycles workflow (with 30 tasks).

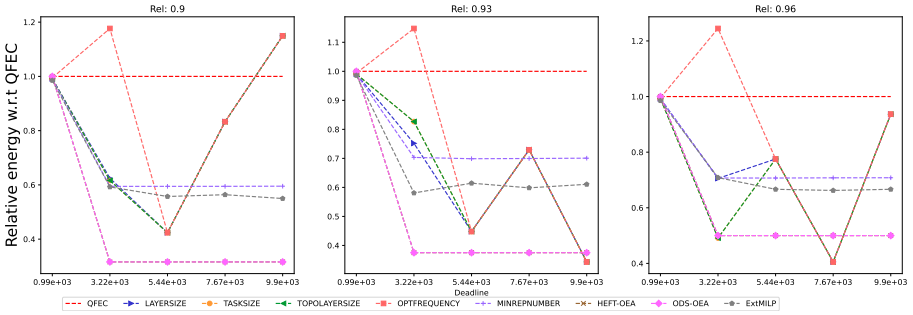

Figure 2229: Assessing the performance of ExtMILP on the Epigenomics workflow (with 10 tasks).

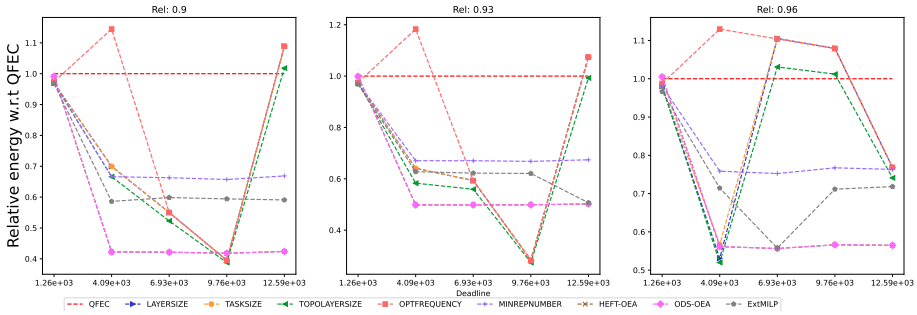

Figure 2230: Assessing the performance of ExtMILP on the Epigenomics workflow (with 20 tasks).

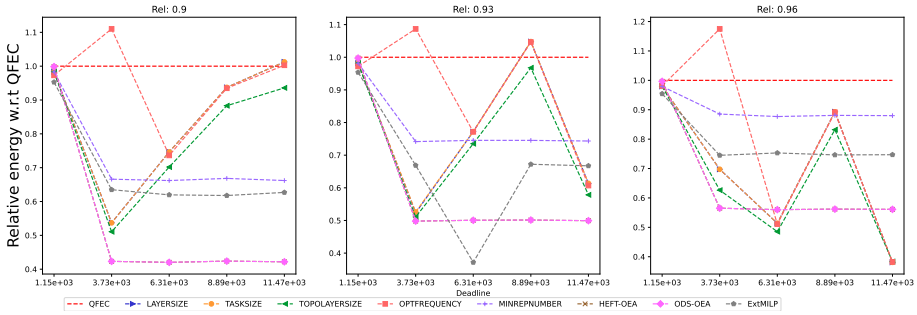

Figure 2231: Assessing the performance of ExtMILP on the Epigenomics workflow (with 30 tasks).

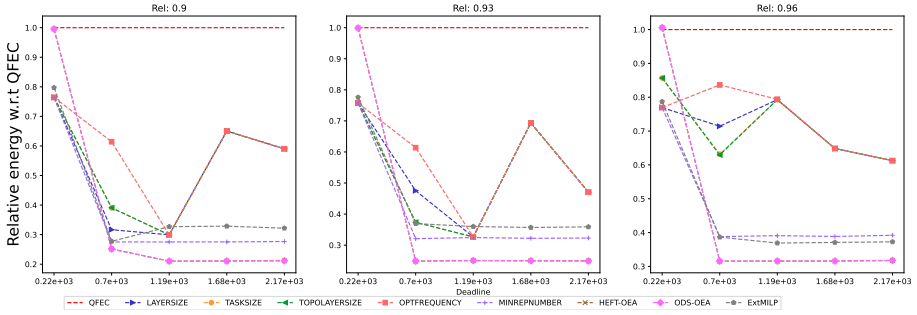

Figure 2232: Assessing the performance of ExtMILP on the Genome workflow (with 10 tasks).

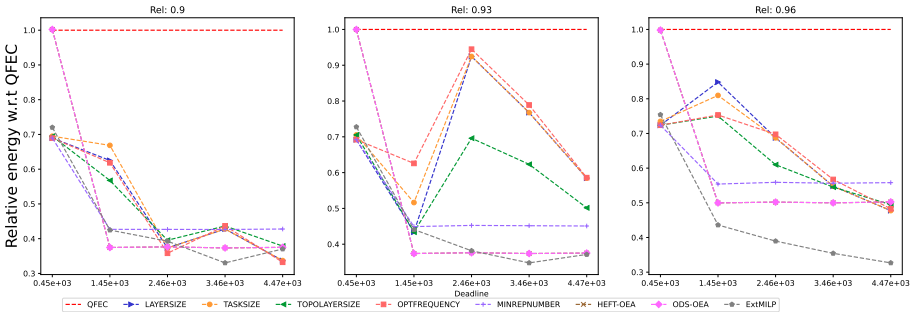

Figure 2233: Assessing the performance of ExtMILP on the Genome workflow (with 20 tasks).

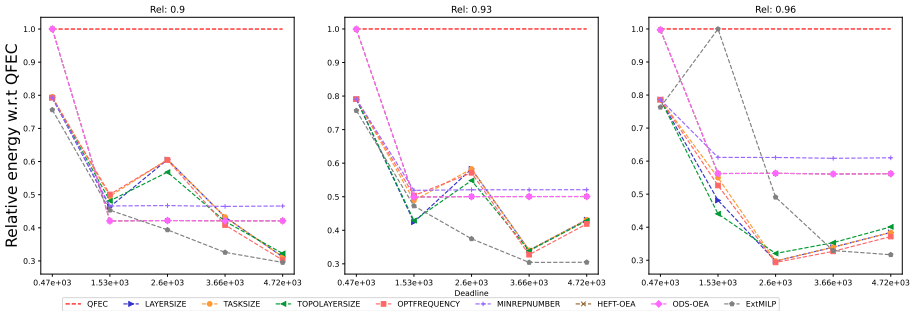

Figure 2234: Assessing the performance of ExtMILP on the Genome workflow (with 30 tasks).

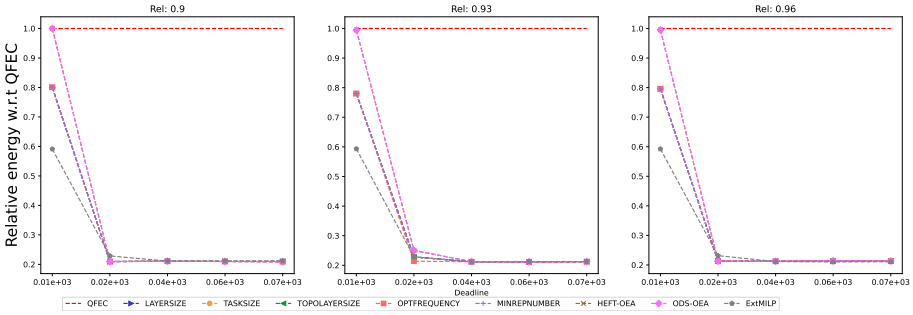

Figure 2235: Assessing the performance of ExtMILP on the Seismology workflow (with 10 tasks).

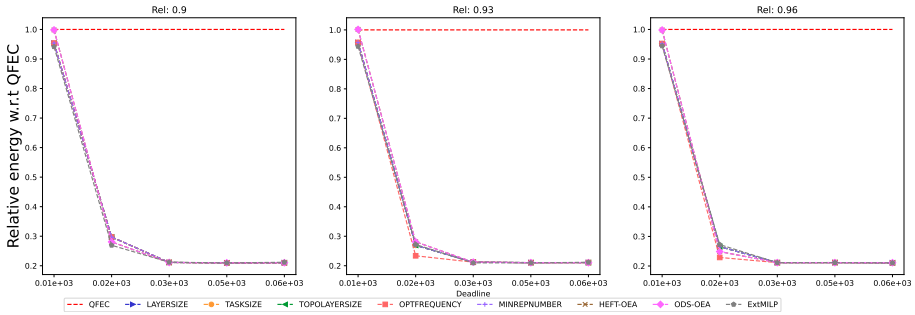

Figure 2236: Assessing the performance of ExtMILP on the Seismology workflow (with 20 tasks).

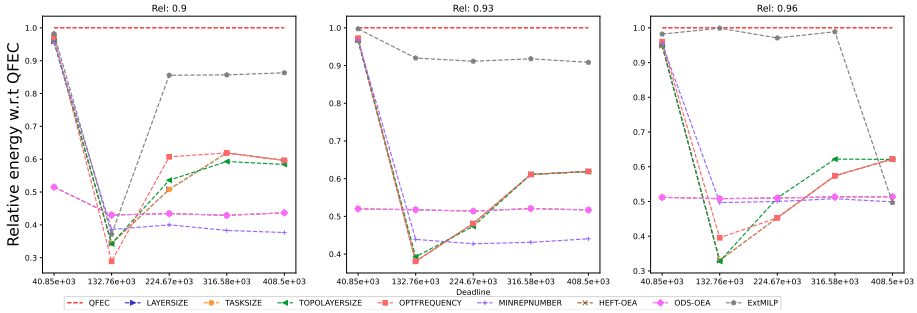

Figure 2237: Assessing the performance of ExtMILP on the SoyKB workflow (with 10 tasks).

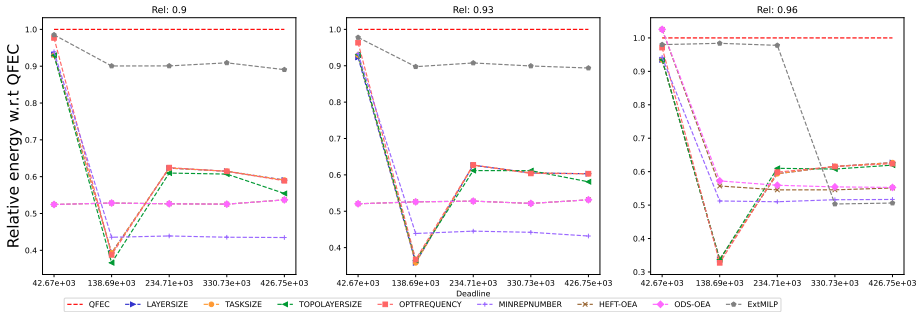

Figure 2238: Assessing the performance of ExtMILP on the SoyKB workflow (with 20 tasks).

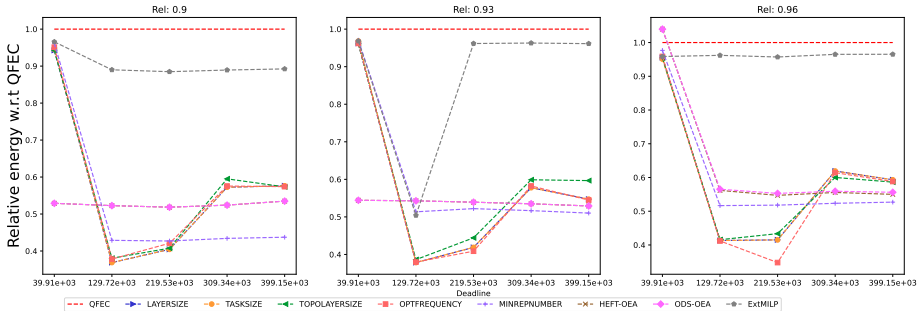

Figure 2239: Assessing the performance of ExtMILP on the SoyKB workflow (with 30 tasks).

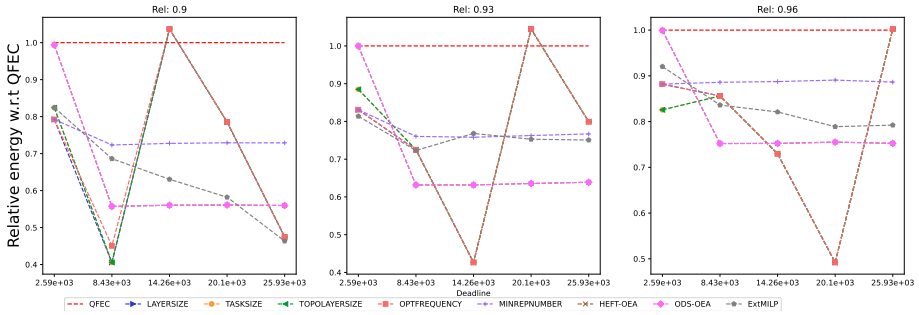

Figure 2240: Assessing the performance of ExtMILP on the SRASearch workflow (with 10 tasks).

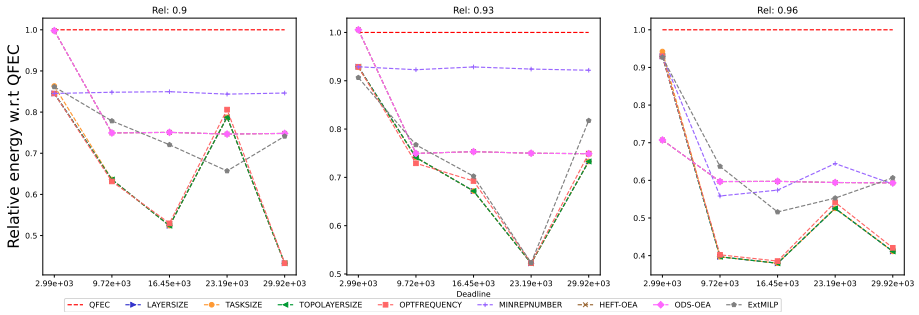

Figure 2241: Assessing the performance of ExtMILP on the SRASearch workflow (with 20 tasks).

H Study of different processor number influence when actual execution times are drawn from a truncated normal distribution

H.1 $BC/WC = 0.1$

H.1.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

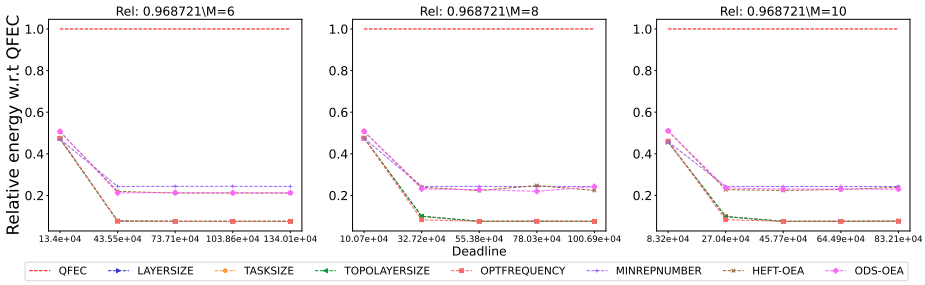

Figure 2242: Assessing the performance of different processor number on the BLAST workflow.

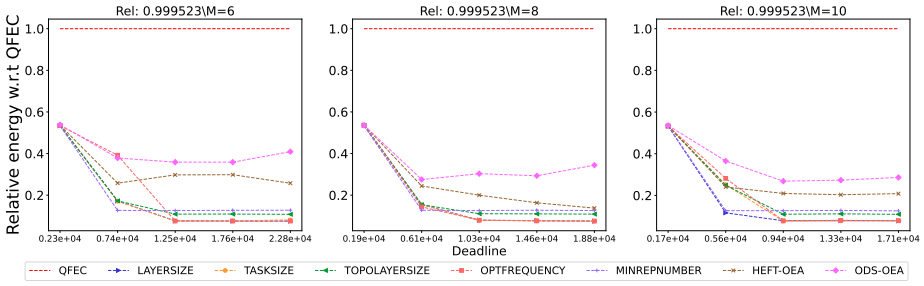

Figure 2243: Assessing the performance of different processor number on the BWA workflow.

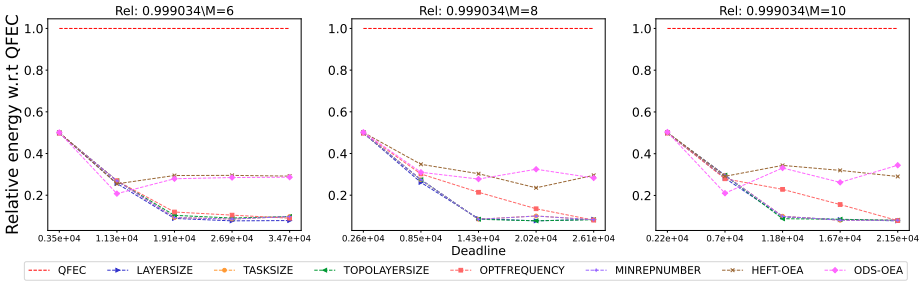

Figure 2244: Assessing the performance of different processor number on the Cholesky workflow.

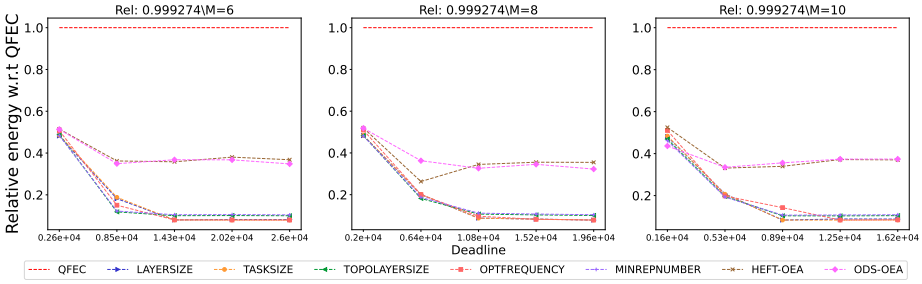

Figure 2245: Assessing the performance of different processor number on the Cycles workflow.

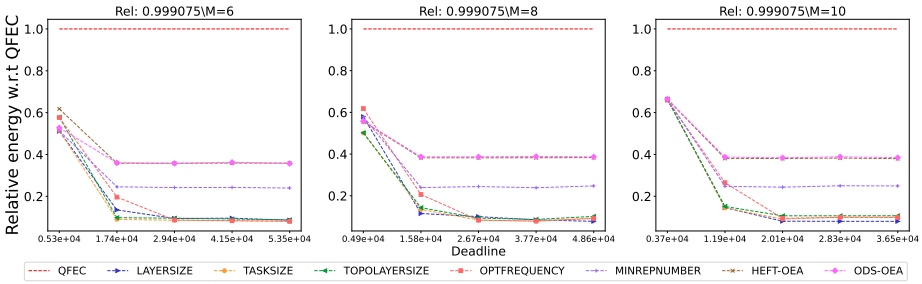

Figure 2246: Assessing the performance of different processor number on the Epigenomics workflow.

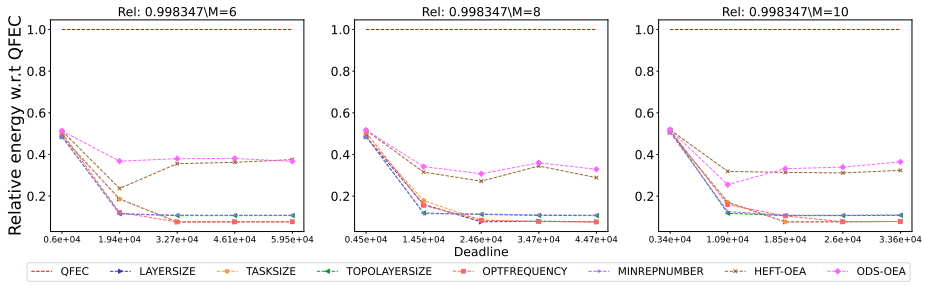

Figure 2247: Assessing the performance of different processor number on the Genome workflow.

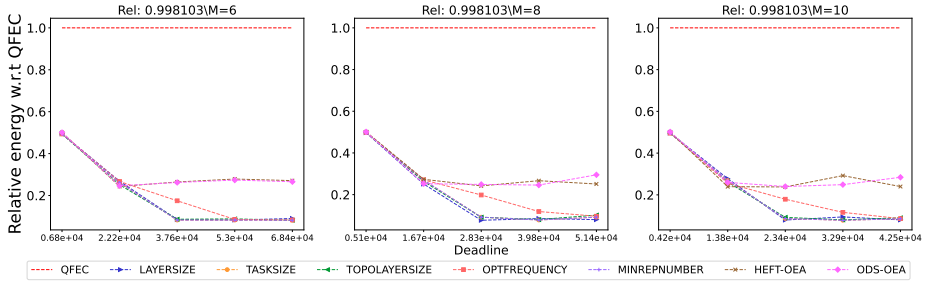

Figure 2248: Assessing the performance of different processor number on the LU.

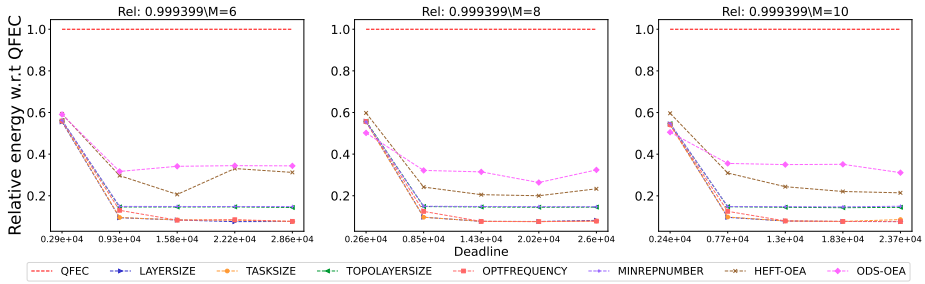

Figure 2249: Assessing the performance of different processor number on the Montage workflow.

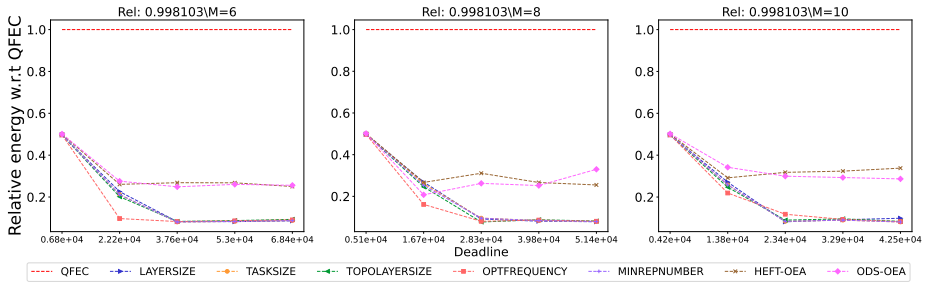

Figure 2250: Assessing the performance of different processor number on the QR.

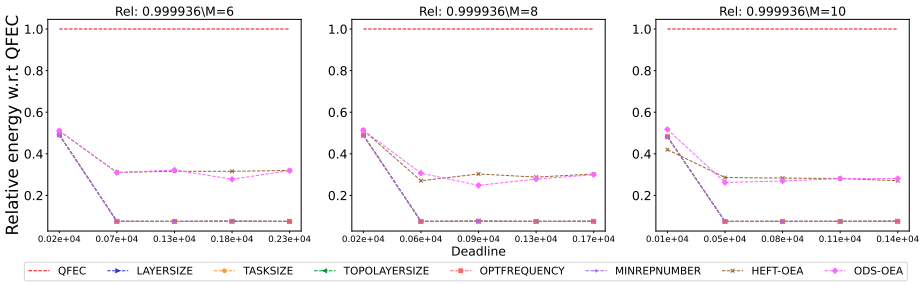

Figure 2251: Assessing the performance of different processor number on the Seismology workflow.

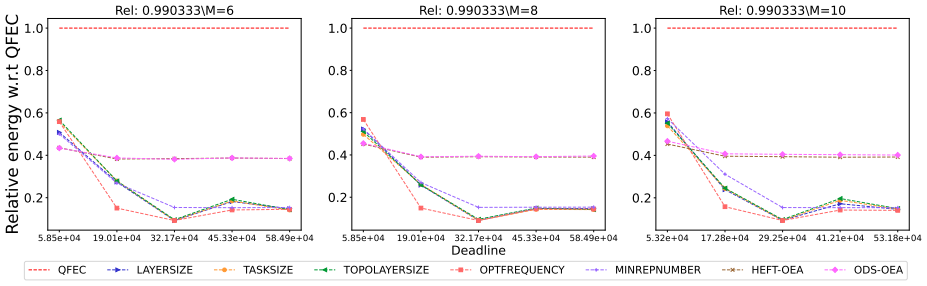

Figure 2252: Assessing the performance of different processor number on the SoyKB workflow.

H.1.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

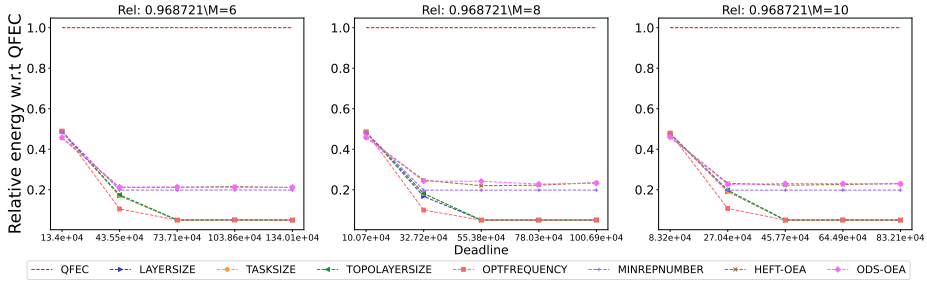

Figure 2253: Assessing the performance of different processor number on the BLAST workflow.

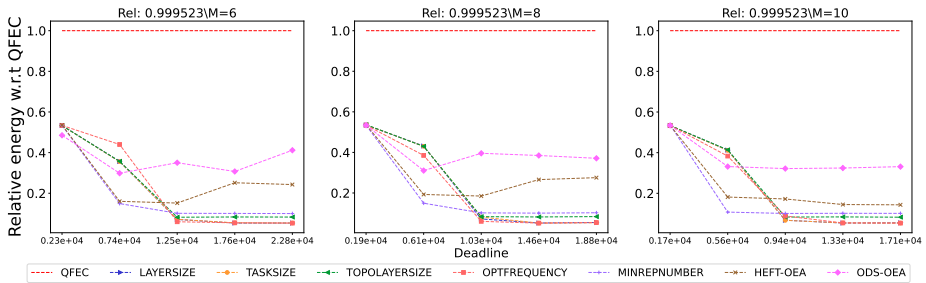

Figure 2254: Assessing the performance of different processor number on the BWA workflow.

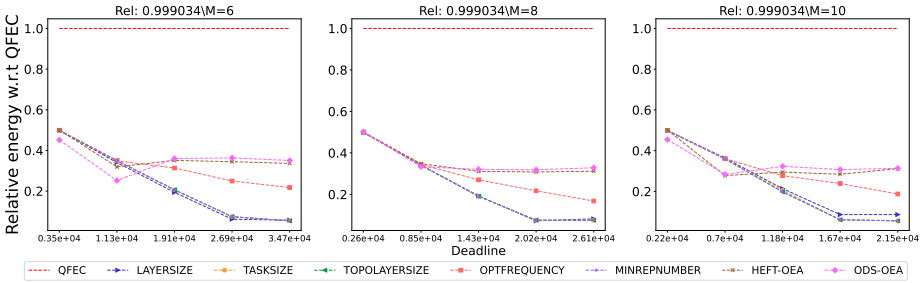

Figure 2255: Assessing the performance of different processor number on the Cholesky workflow.

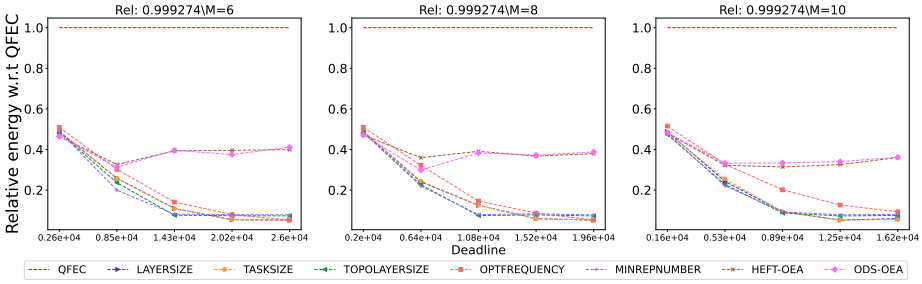

Figure 2256: Assessing the performance of different processor number on the Cycles workflow.

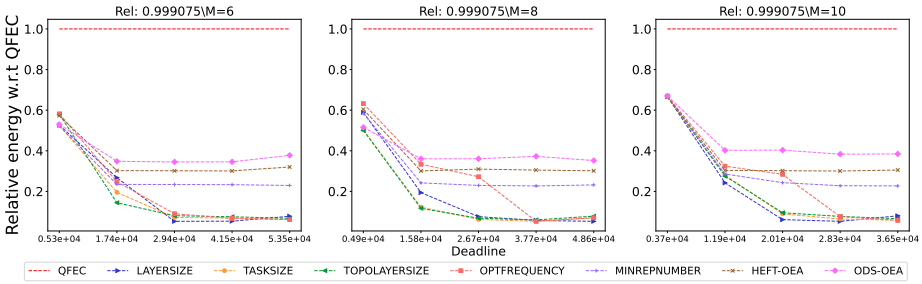

Figure 2257: Assessing the performance of different processor number on the Epigenomics workflow.

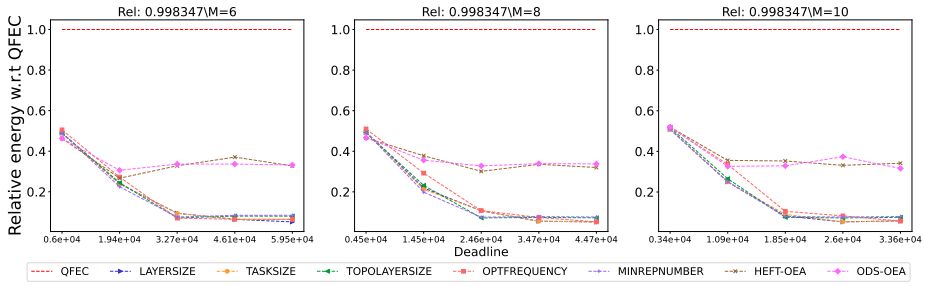

Figure 2258: Assessing the performance of different processor number on the Genome workflow.

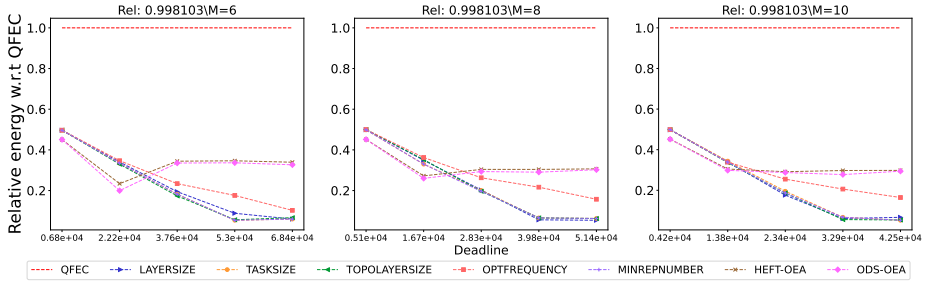

Figure 2259: Assessing the performance of different processor number on the LU.

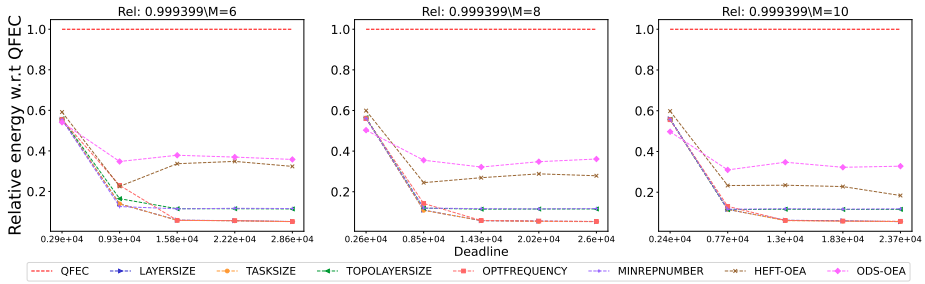

Figure 2260: Assessing the performance of different processor number on the Montage workflow.

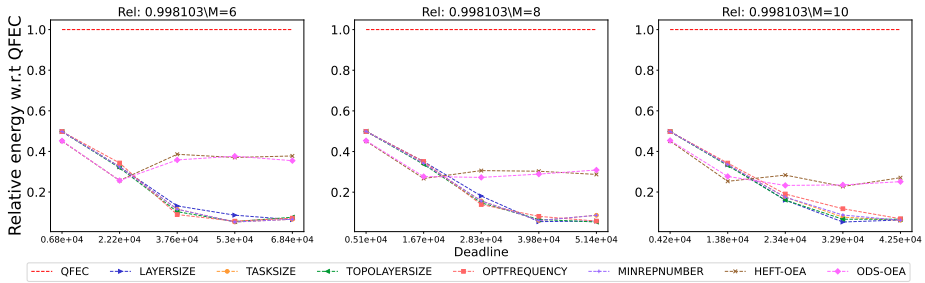

Figure 2261: Assessing the performance of different processor number on the QR.

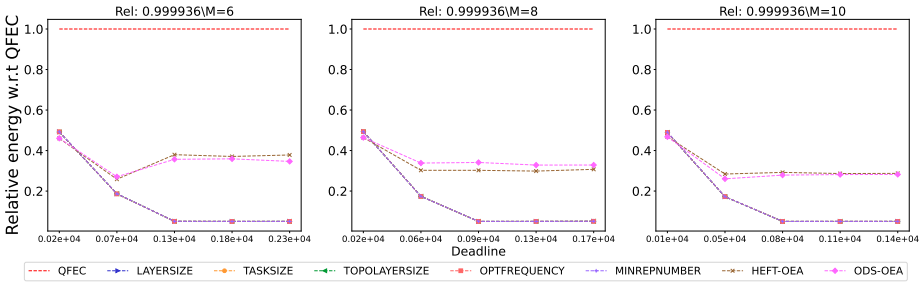

Figure 2262: Assessing the performance of different processor number on the Seismology workflow.

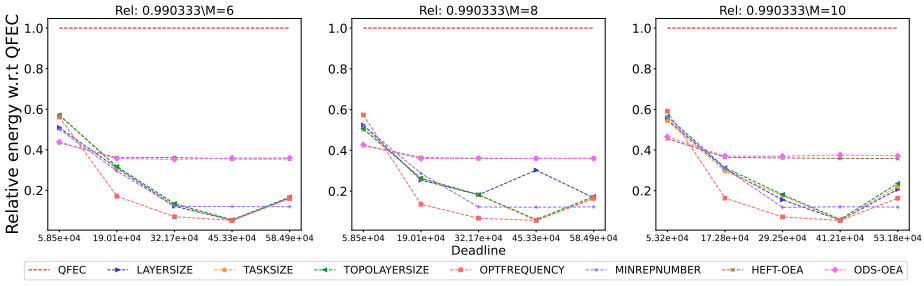

Figure 2263: Assessing the performance of different processor number on the SoyKB workflow.

H.1.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

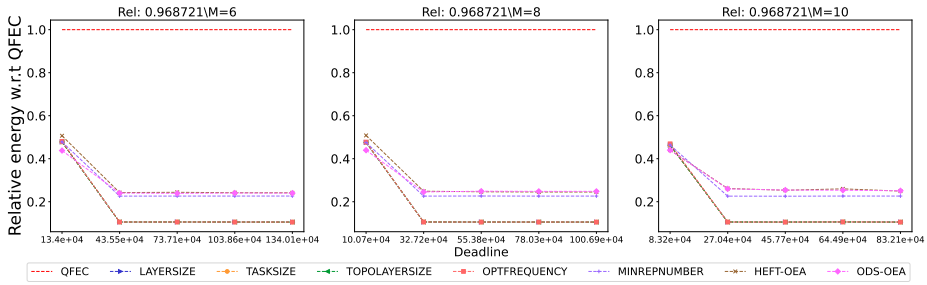

Figure 2264: Assessing the performance of different processor number on the BLAST workflow.

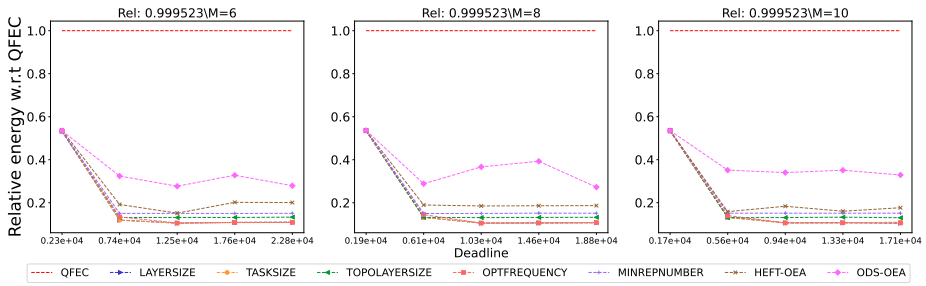

Figure 2265: Assessing the performance of different processor number on the BWA workflow.

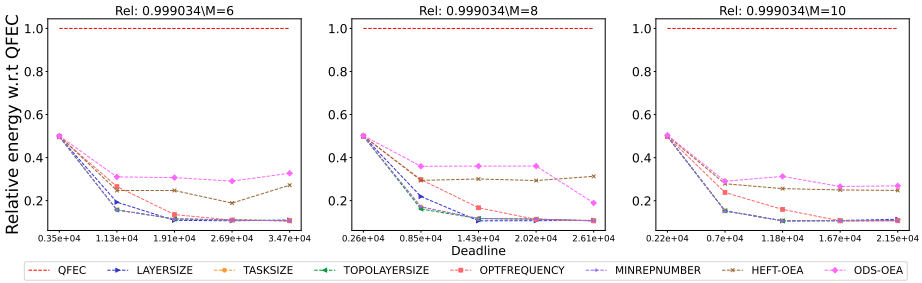

Figure 2266: Assessing the performance of different processor number on the Cholesky workflow.

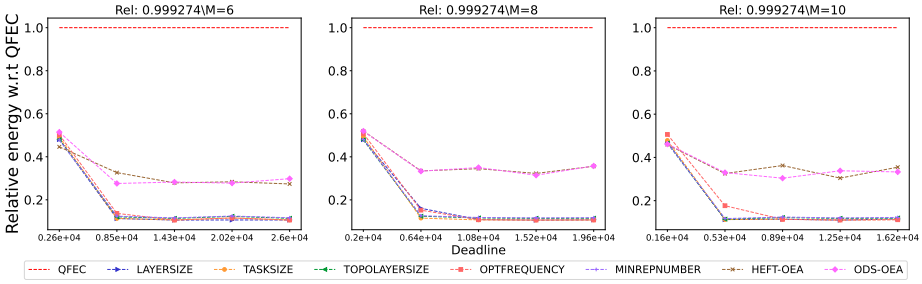

Figure 2267: Assessing the performance of different processor number on the Cycles workflow.

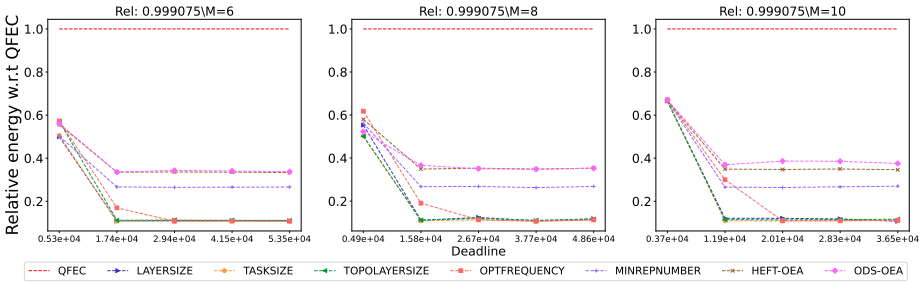

Figure 2268: Assessing the performance of different processor number on the Epigenomics workflow.

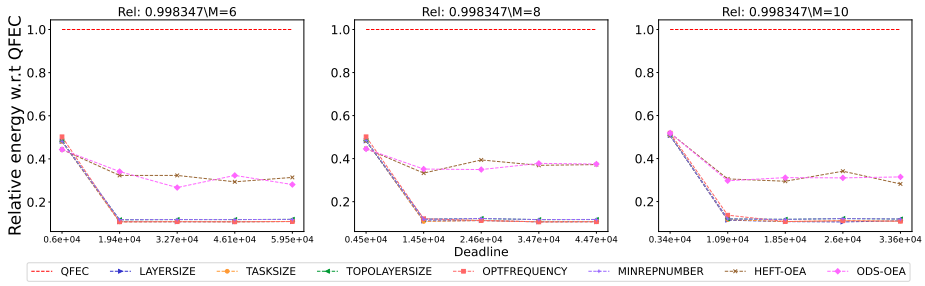

Figure 2269: Assessing the performance of different processor number on the Genome workflow.

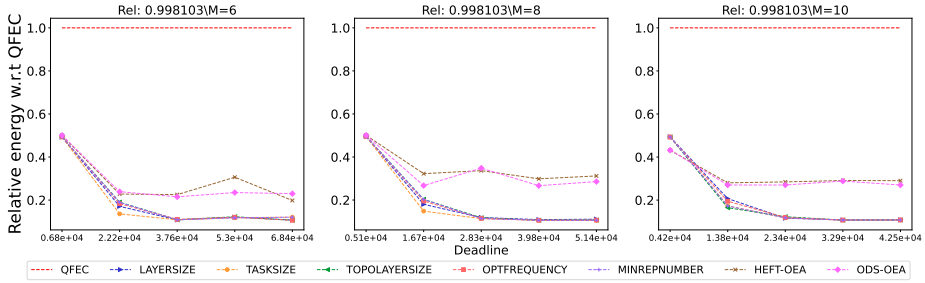

Figure 2270: Assessing the performance of different processor number on the LU.

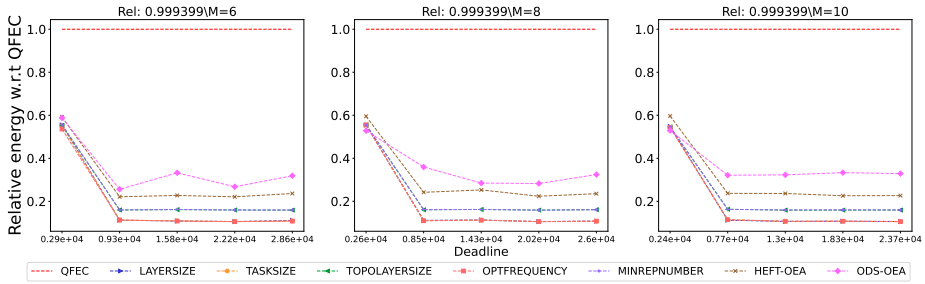

Figure 2271: Assessing the performance of different processor number on the Montage workflow.

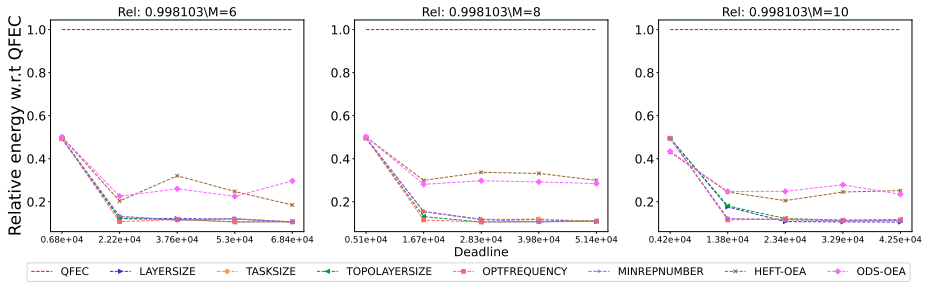

Figure 2272: Assessing the performance of different processor number on the QR.

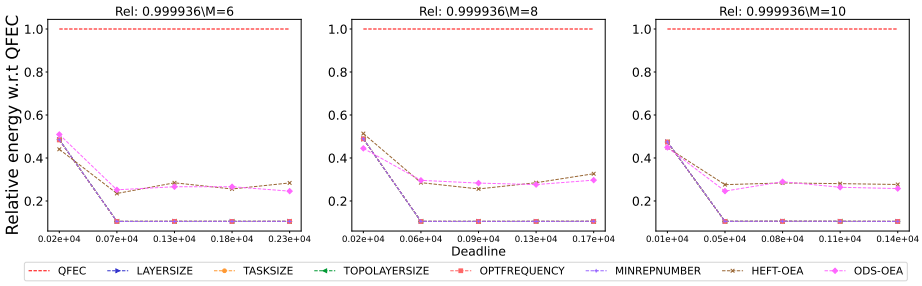

Figure 2273: Assessing the performance of different processor number on the Seismology workflow.

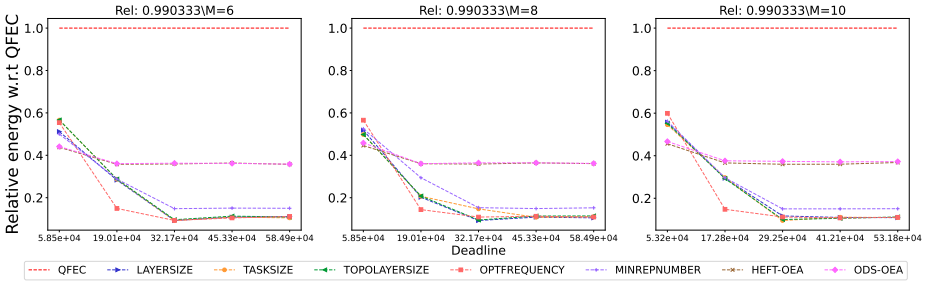

Figure 2274: Assessing the performance of different processor number on the SoyKB workflow.

H.1.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

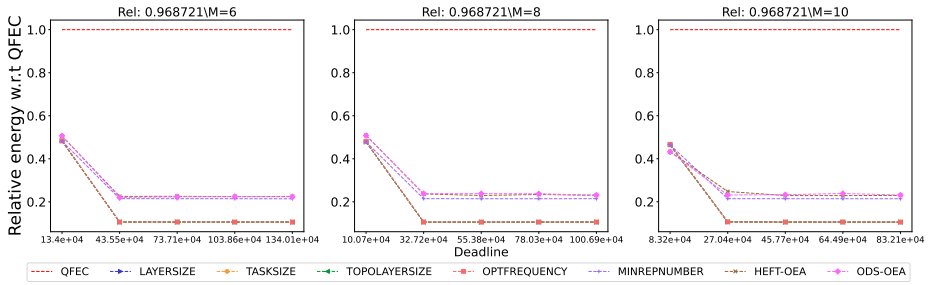

Figure 2275: Assessing the performance of different processor number on the BLAST workflow.

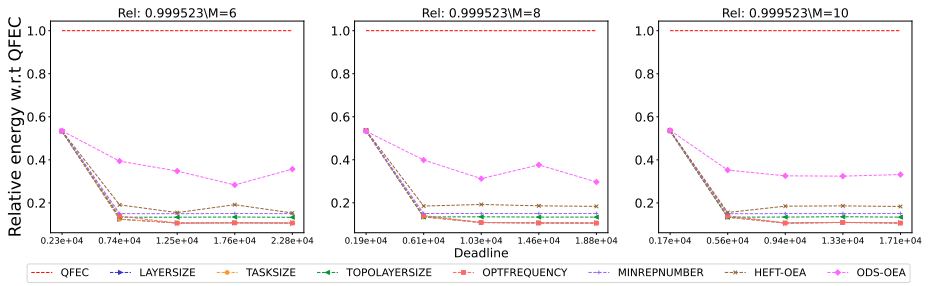

Figure 2276: Assessing the performance of different processor number on the BWA workflow.

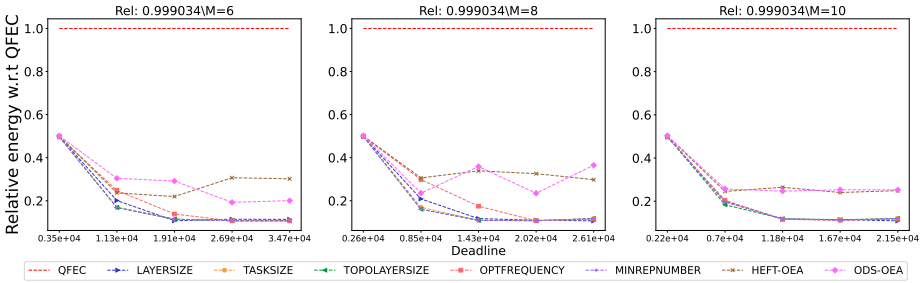

Figure 2277: Assessing the performance of different processor number on the Cholesky workflow.

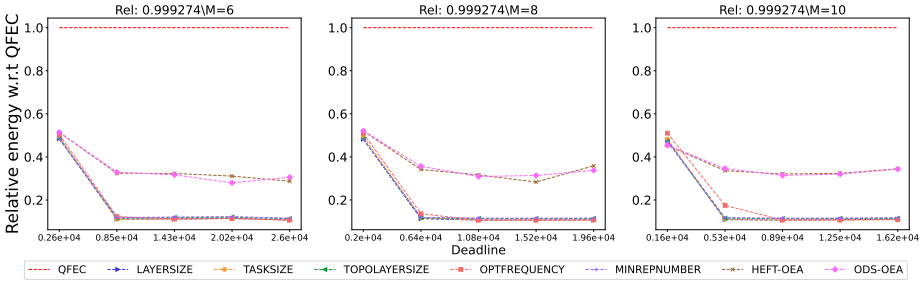

Figure 2278: Assessing the performance of different processor number on the Cycles workflow.

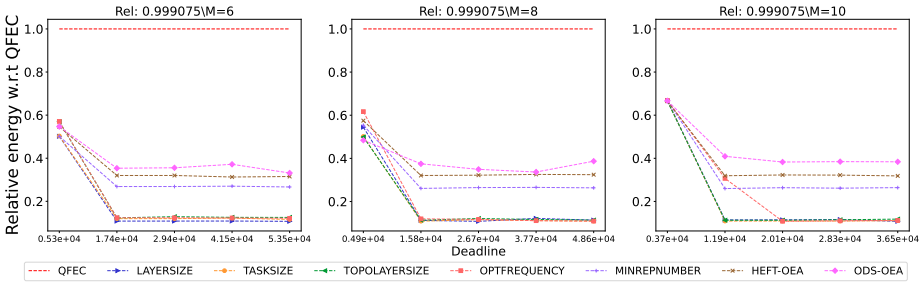

Figure 2279: Assessing the performance of different processor number on the Epigenomics workflow.

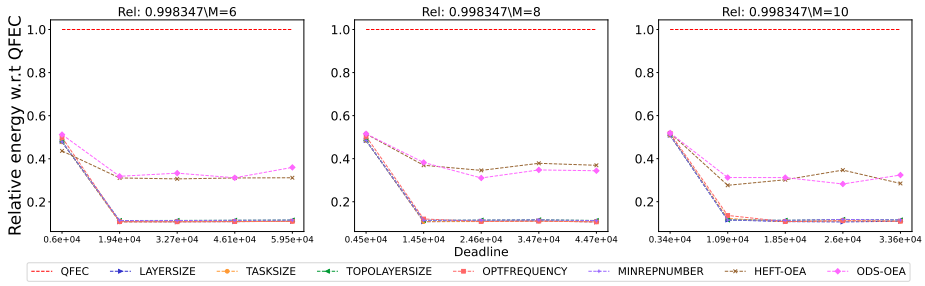

Figure 2280: Assessing the performance of different processor number on the Genome workflow.

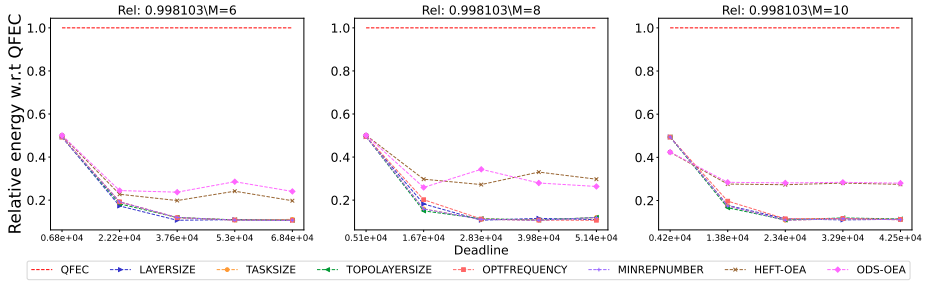

Figure 2281: Assessing the performance of different processor number on the LU.

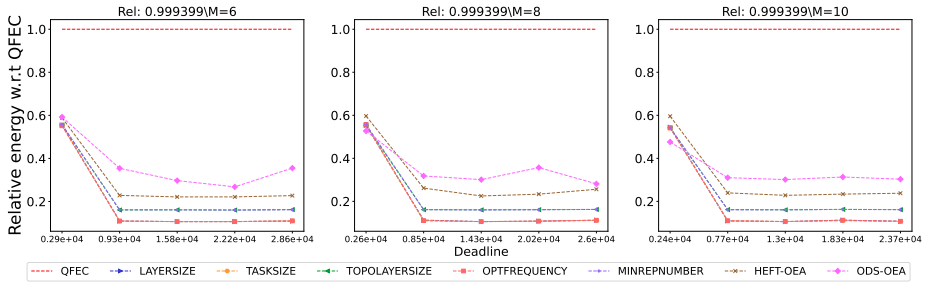

Figure 2282: Assessing the performance of different processor number on the Montage workflow.

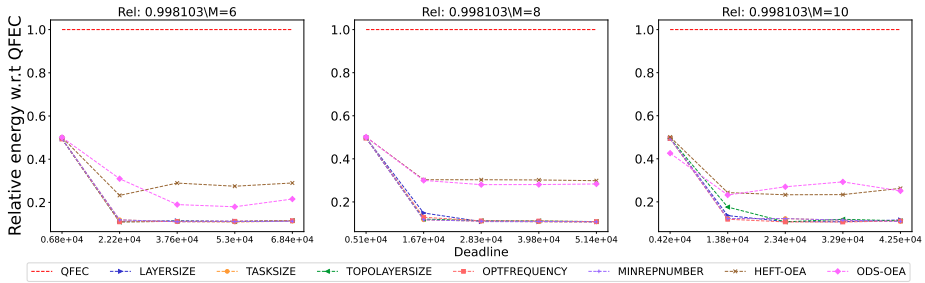

Figure 2283: Assessing the performance of different processor number on the QR.

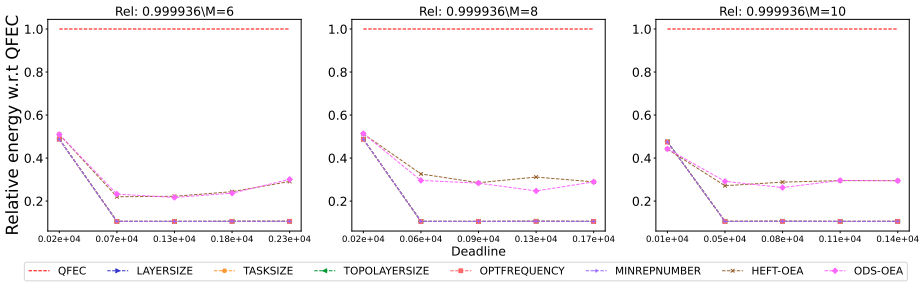

Figure 2284: Assessing the performance of different processor number on the Seismology workflow.

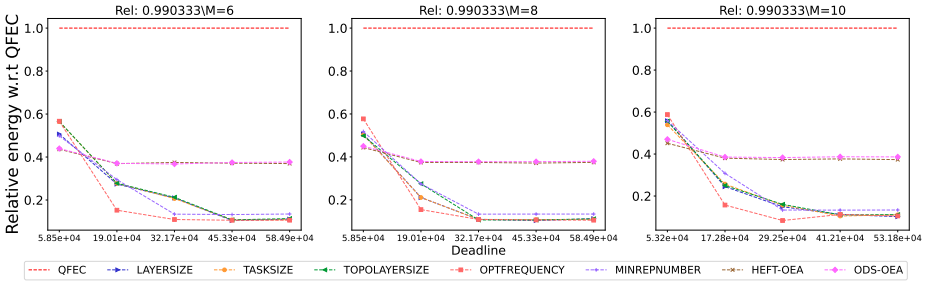

Figure 2285: Assessing the performance of different processor number on the SoyKB workflow.

H.2 $BC/WC = 0.2$

H.2.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

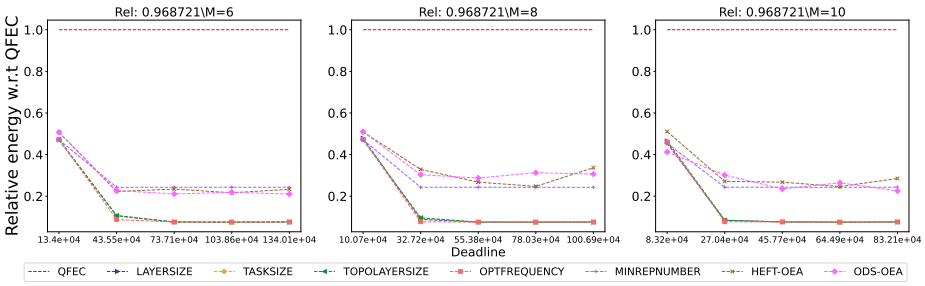

Figure 2286: Assessing the performance of different processor number on the BLAST workflow.

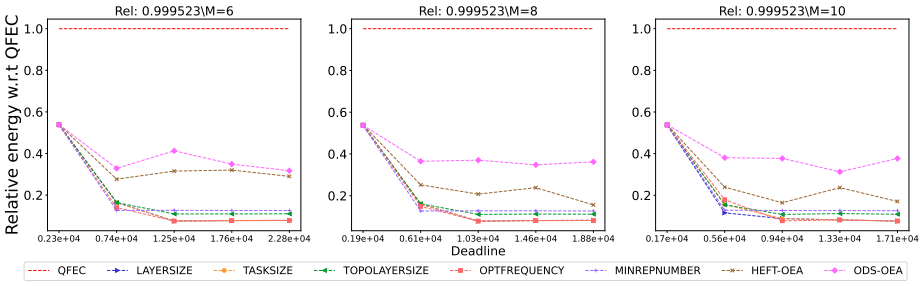

Figure 2287: Assessing the performance of different processor number on the BWA workflow.

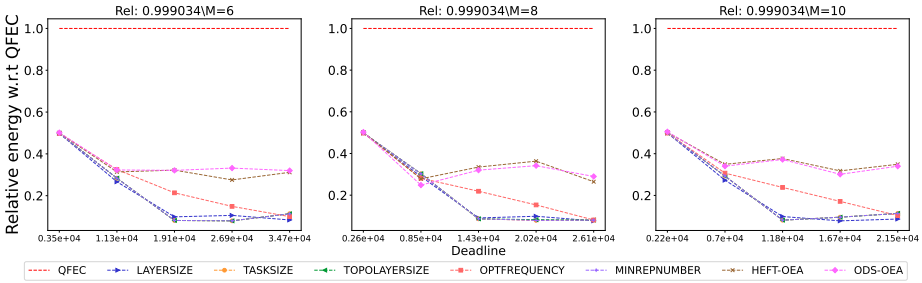

Figure 2288: Assessing the performance of different processor number on the Cholesky workflow.

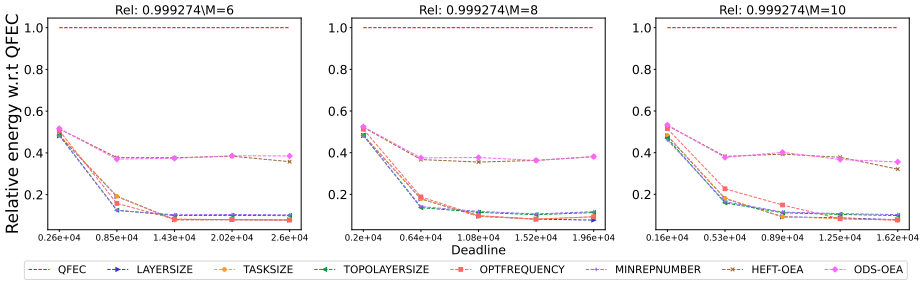

Figure 2289: Assessing the performance of different processor number on the Cycles workflow.

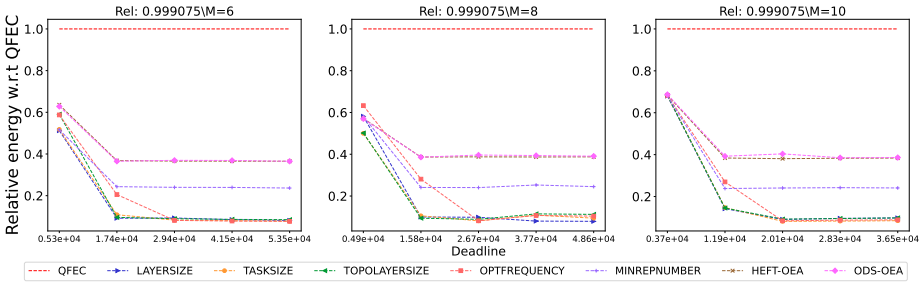

Figure 2290: Assessing the performance of different processor number on the Epigenomics workflow.

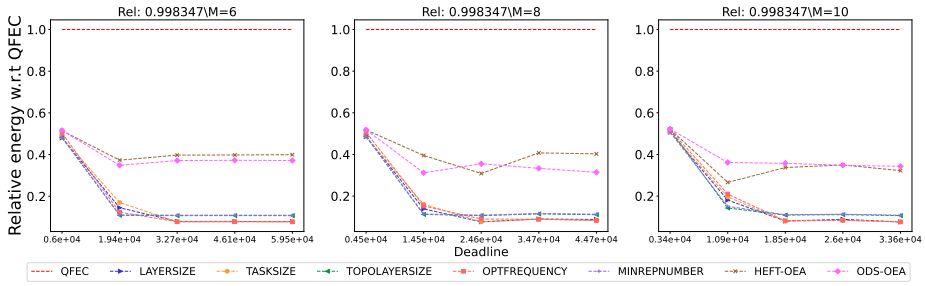

Figure 2291: Assessing the performance of different processor number on the Genome workflow.

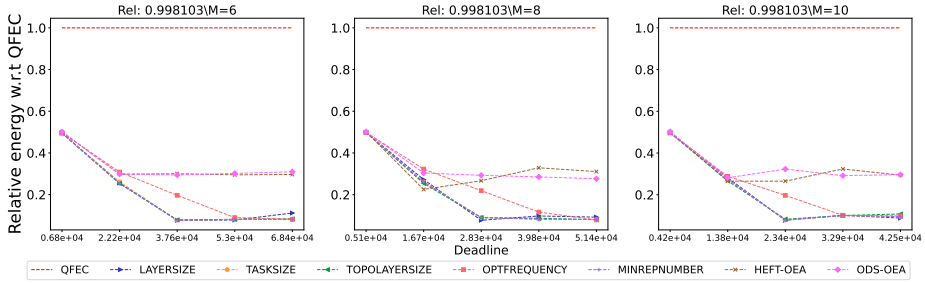

Figure 2292: Assessing the performance of different processor number on the LU.

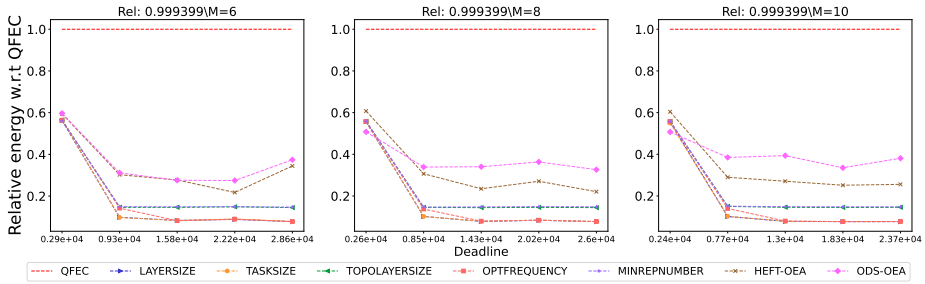

Figure 2293: Assessing the performance of different processor number on the Montage workflow.

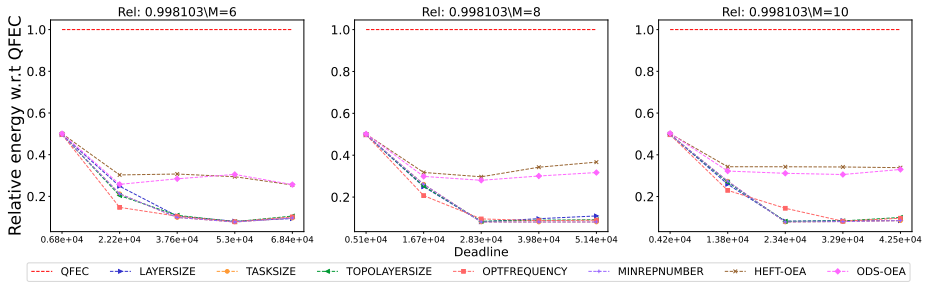

Figure 2294: Assessing the performance of different processor number on the QR.

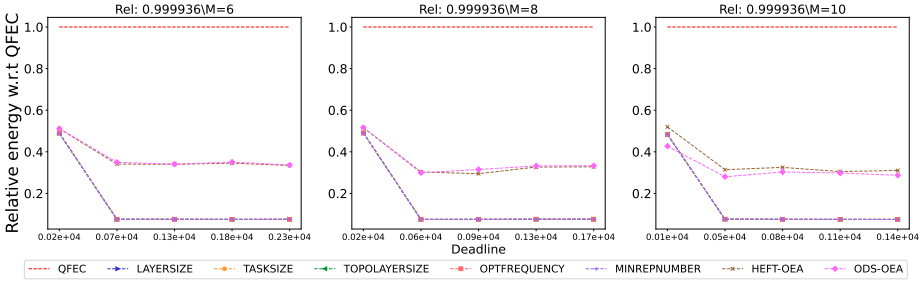

Figure 2295: Assessing the performance of different processor number on the Seismology workflow.

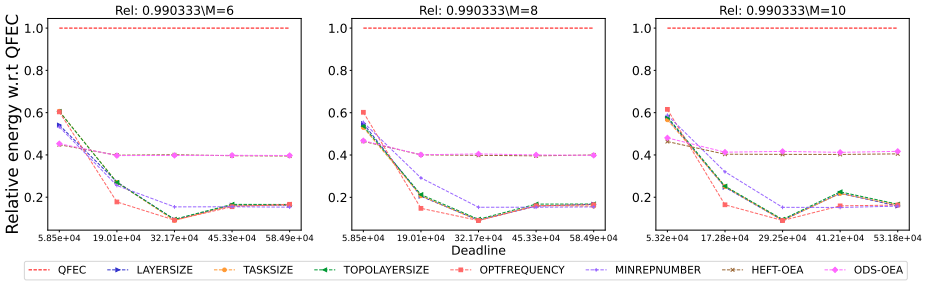

Figure 2296: Assessing the performance of different processor number on the SoyKB workflow.

H.2.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

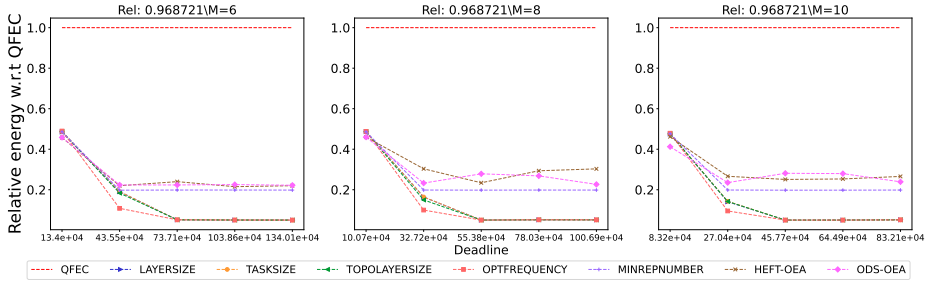

Figure 2297: Assessing the performance of different processor number on the BLAST workflow.

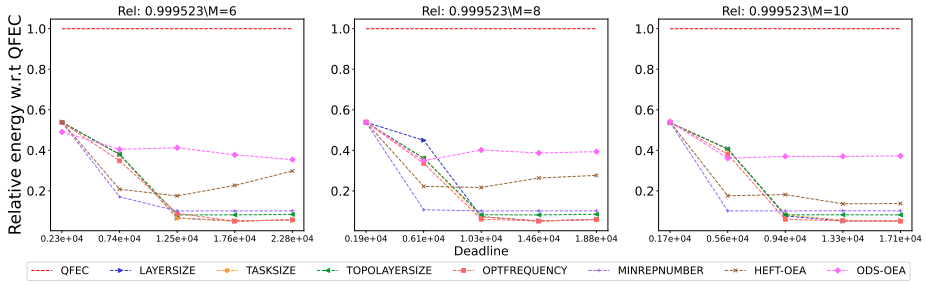

Figure 2298: Assessing the performance of different processor number on the BWA workflow.

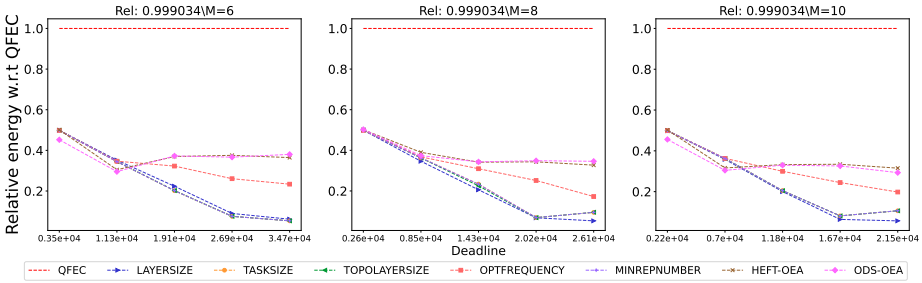

Figure 2299: Assessing the performance of different processor number on the Cholesky workflow.

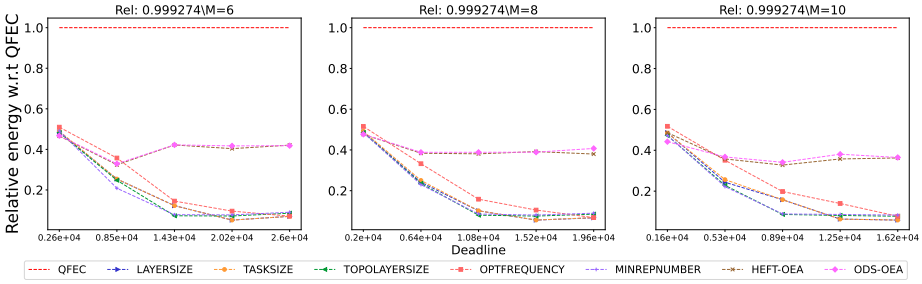

Figure 2300: Assessing the performance of different processor number on the Cycles workflow.

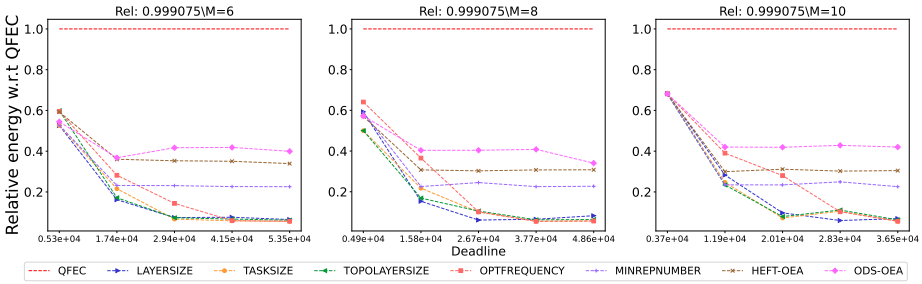

Figure 2301: Assessing the performance of different processor number on the Epigenomics workflow.

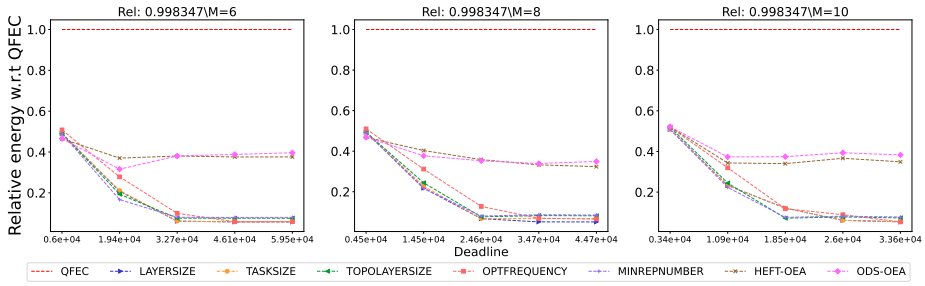

Figure 2302: Assessing the performance of different processor number on the Genome workflow.

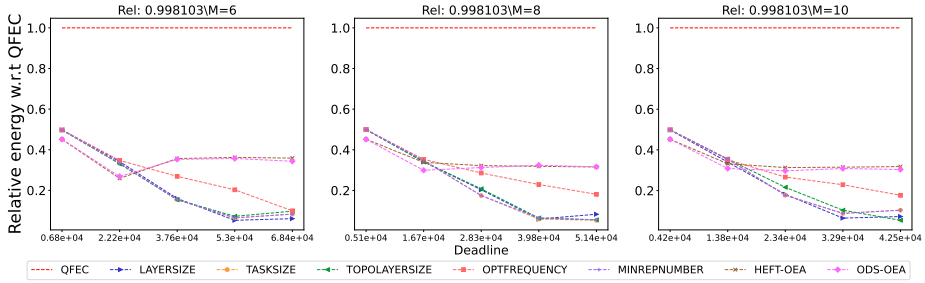

Figure 2303: Assessing the performance of different processor number on the LU.

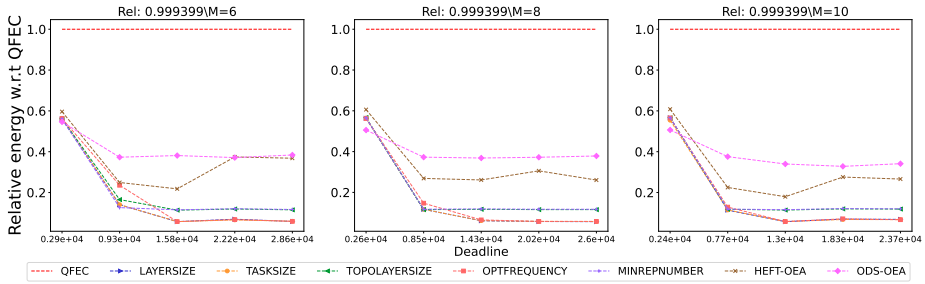

Figure 2304: Assessing the performance of different processor number on the Montage workflow.

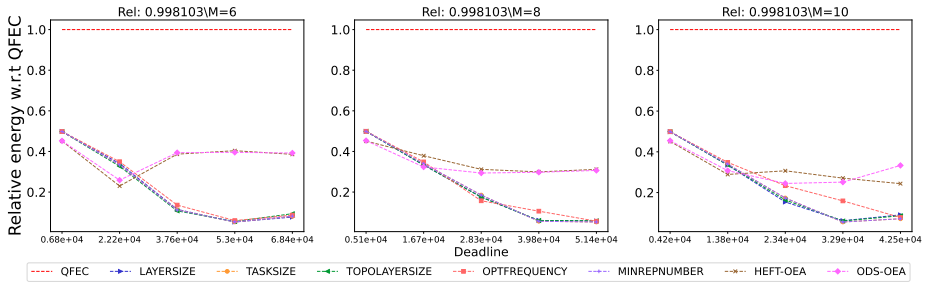

Figure 2305: Assessing the performance of different processor number on the QR.

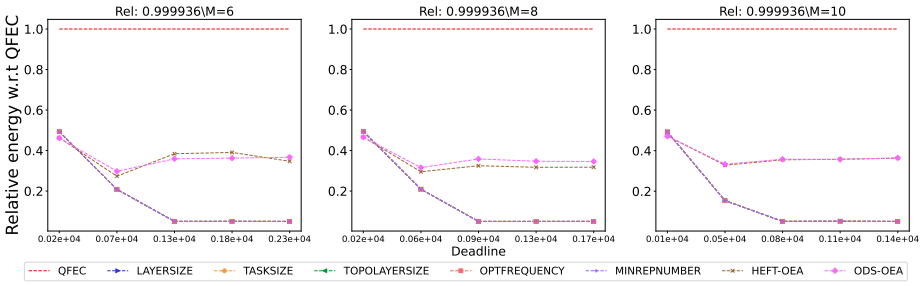

Figure 2306: Assessing the performance of different processor number on the Seismology workflow.

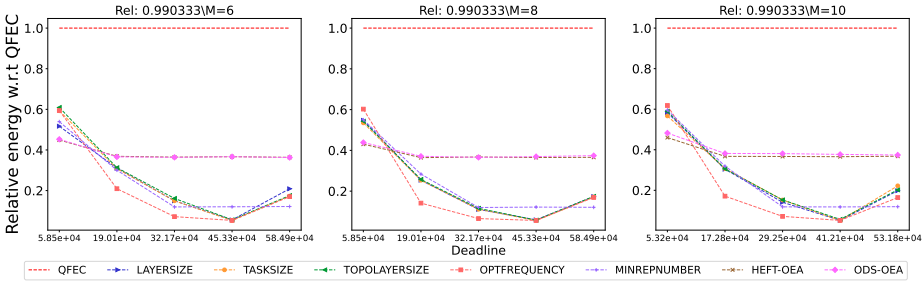

Figure 2307: Assessing the performance of different processor number on the SoyKB workflow.

H.2.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

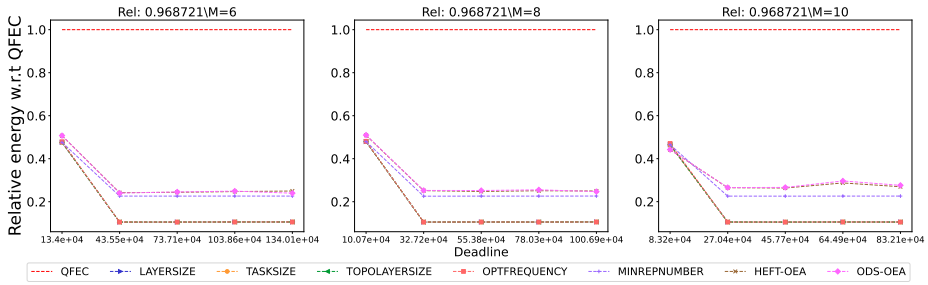

Figure 2308: Assessing the performance of different processor number on the BLAST workflow.

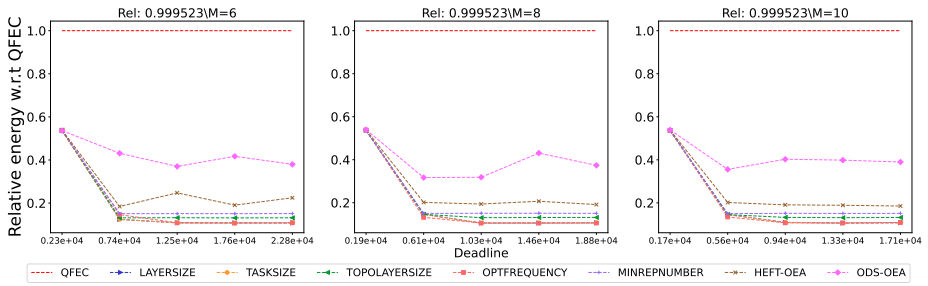

Figure 2309: Assessing the performance of different processor number on the BWA workflow.

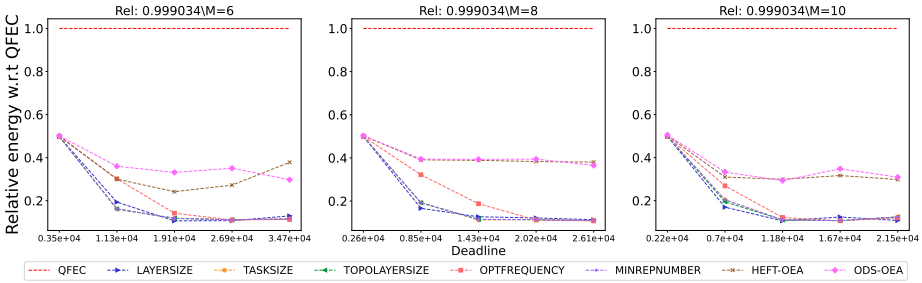

Figure 2310: Assessing the performance of different processor number on the Cholesky workflow.

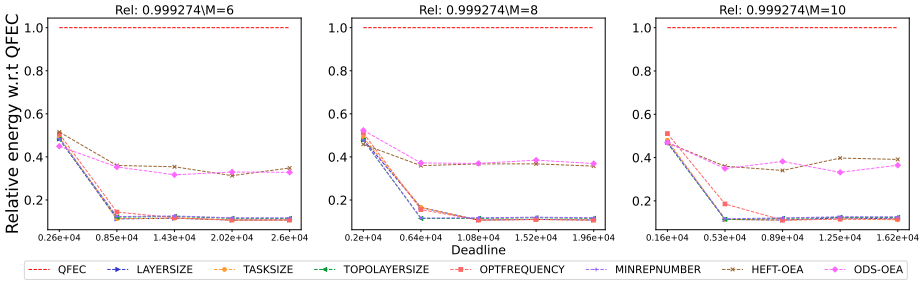

Figure 2311: Assessing the performance of different processor number on the Cycles workflow.

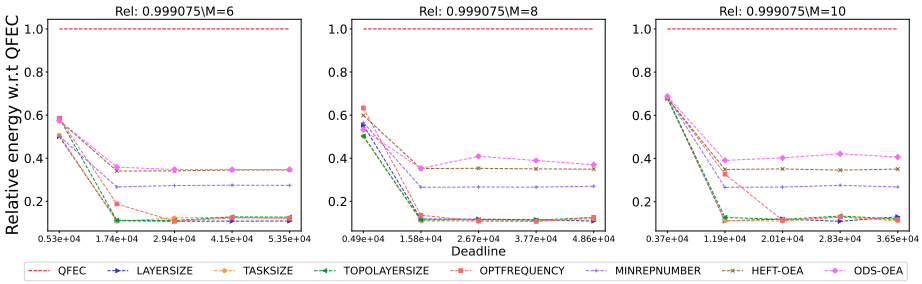

Figure 2312: Assessing the performance of different processor number on the Epigenomics workflow.

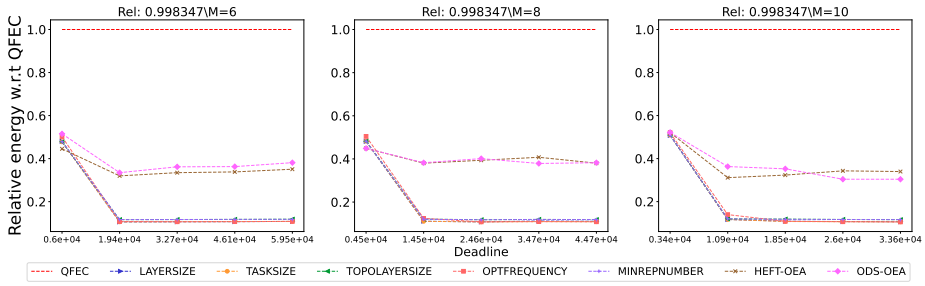

Figure 2313: Assessing the performance of different processor number on the Genome workflow.

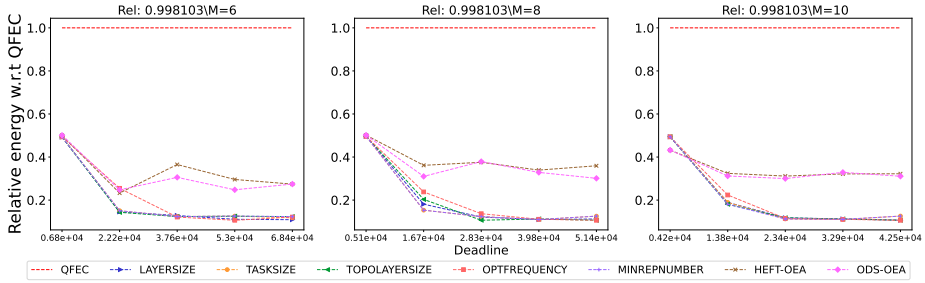

Figure 2314: Assessing the performance of different processor number on the LU.

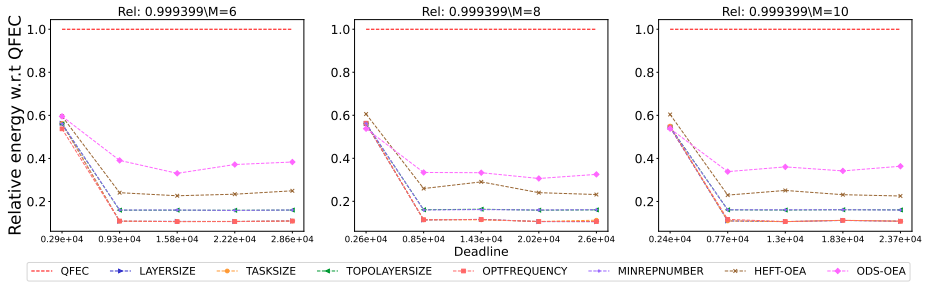

Figure 2315: Assessing the performance of different processor number on the Montage workflow.

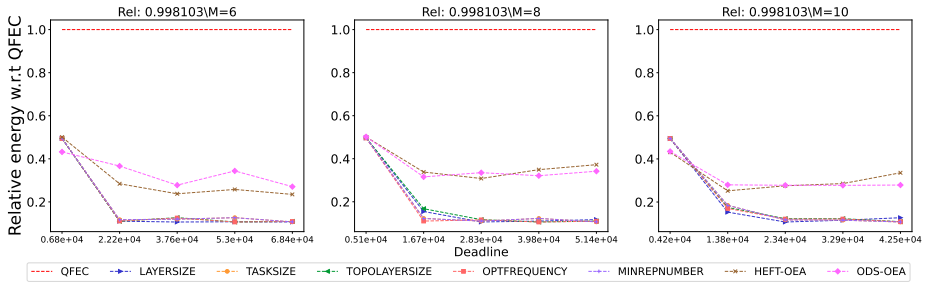

Figure 2316: Assessing the performance of different processor number on the QR.

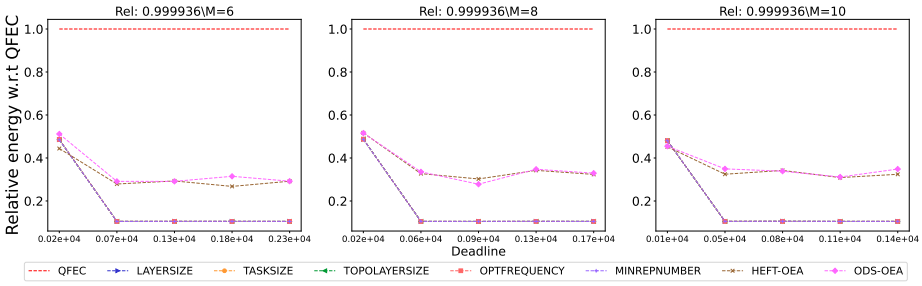

Figure 2317: Assessing the performance of different processor number on the Seismology workflow.

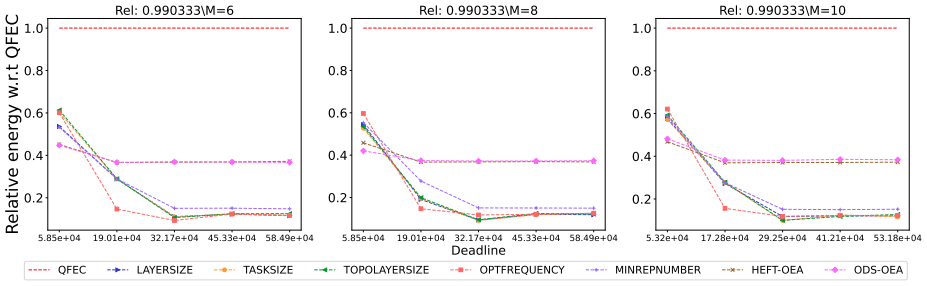

Figure 2318: Assessing the performance of different processor number on the SoyKB workflow.

H.2.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

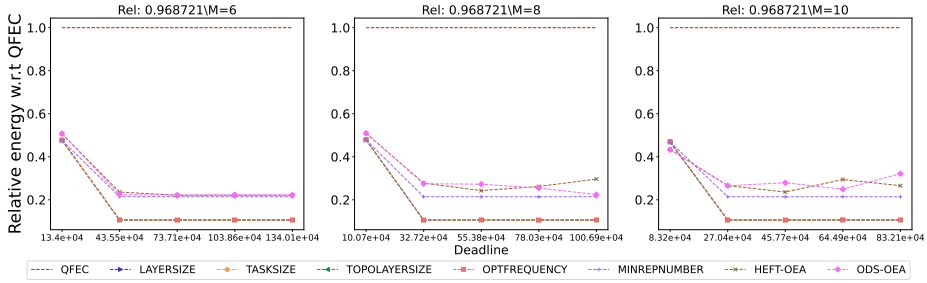

Figure 2319: Assessing the performance of different processor number on the BLAST workflow.

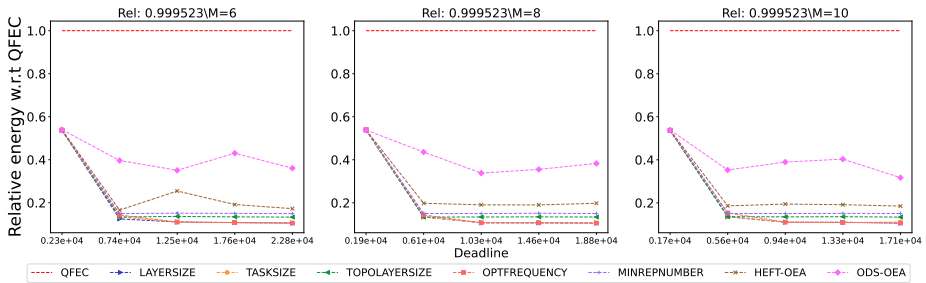

Figure 2320: Assessing the performance of different processor number on the BWA workflow.

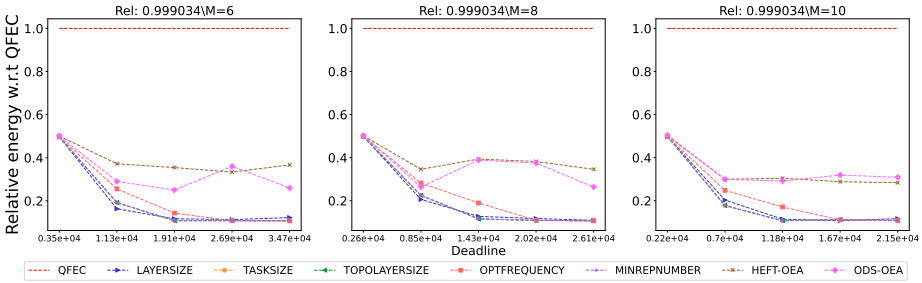

Figure 2321: Assessing the performance of different processor number on the Cholesky workflow.

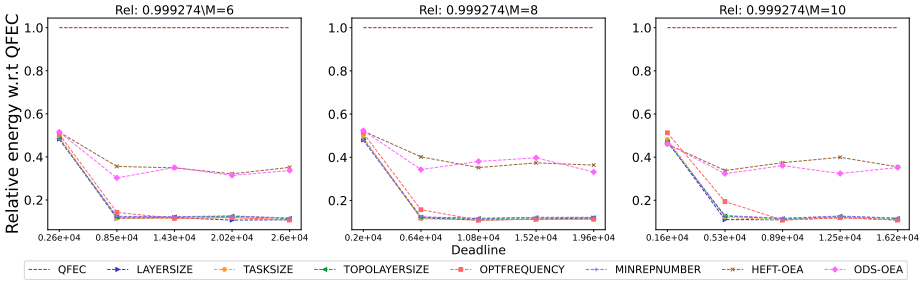

Figure 2322: Assessing the performance of different processor number on the Cycles workflow.

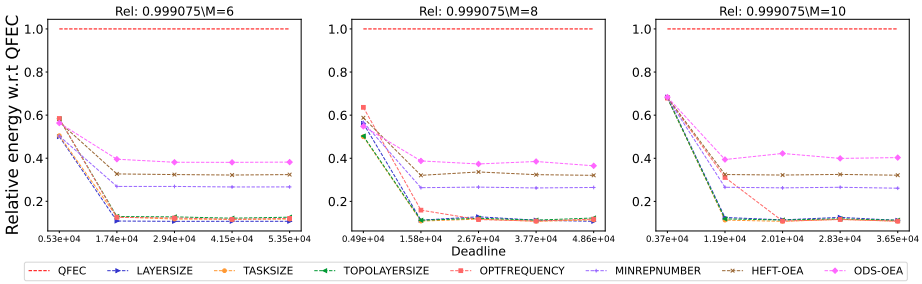

Figure 2323: Assessing the performance of different processor number on the Epigenomics workflow.

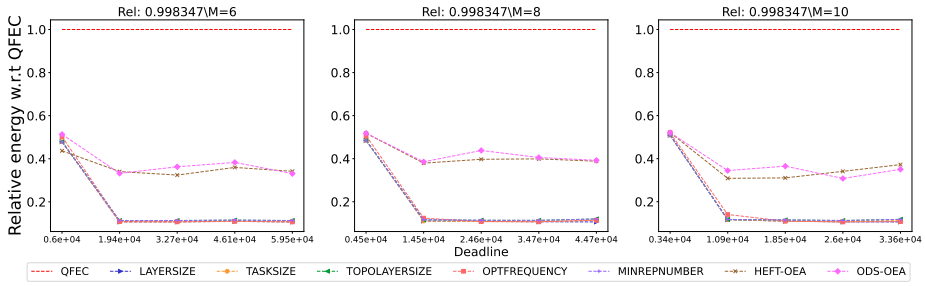

Figure 2324: Assessing the performance of different processor number on the Genome workflow.

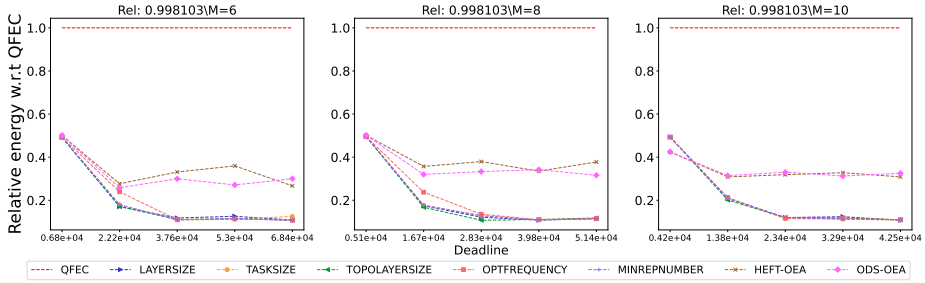

Figure 2325: Assessing the performance of different processor number on the LU.

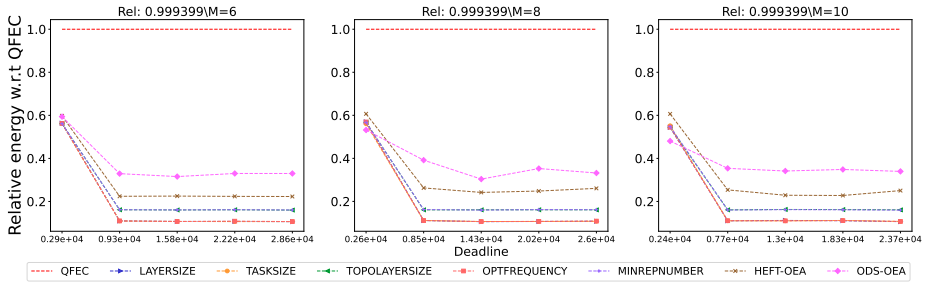

Figure 2326: Assessing the performance of different processor number on the Montage workflow.

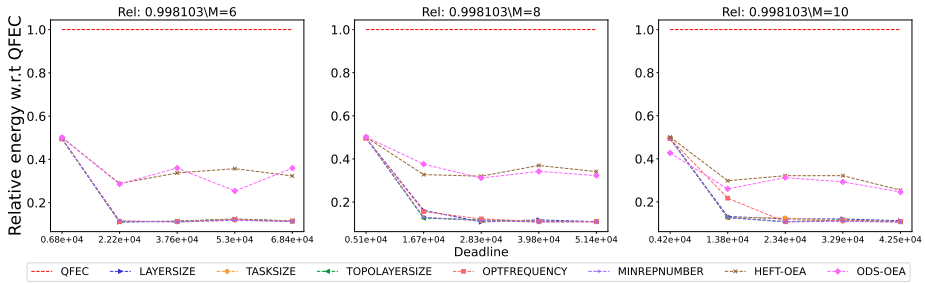

Figure 2327: Assessing the performance of different processor number on the QR.

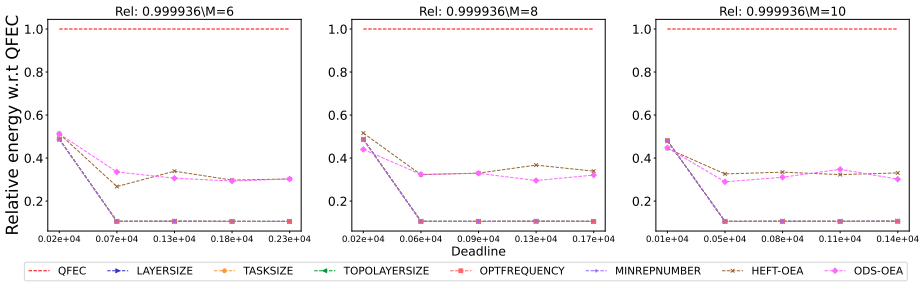

Figure 2328: Assessing the performance of different processor number on the Seismology workflow.

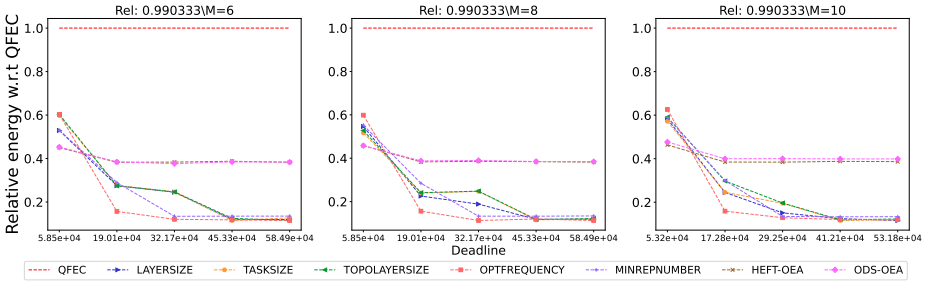

Figure 2329: Assessing the performance of different processor number on the SoyKB workflow.

H.3 $BC/WC = 0.3$

H.3.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

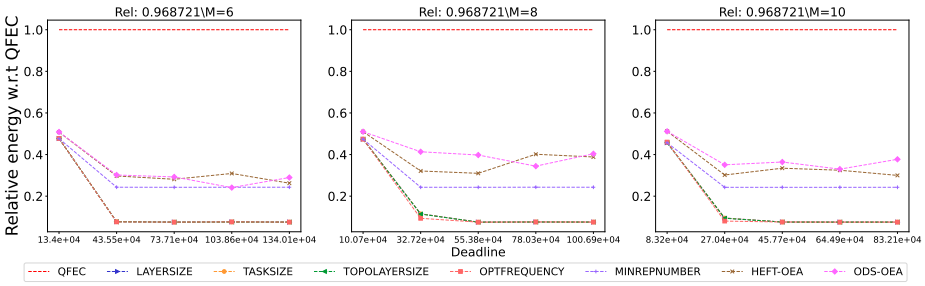

Figure 2330: Assessing the performance of different processor number on the BLAST workflow.

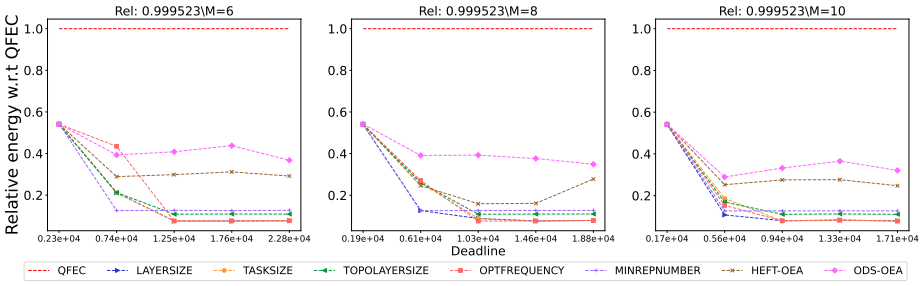

Figure 2331: Assessing the performance of different processor number on the BWA workflow.

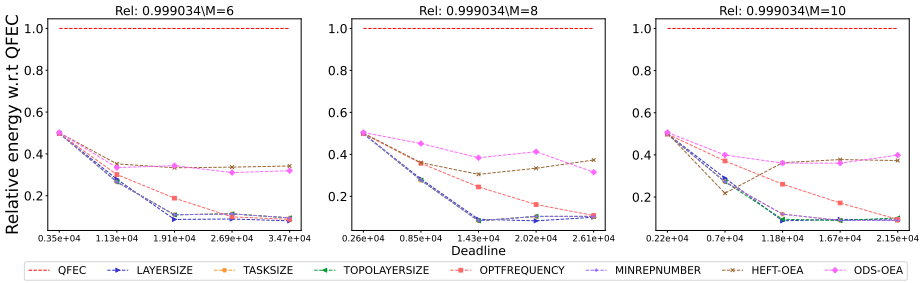

Figure 2332: Assessing the performance of different processor number on the Cholesky workflow.

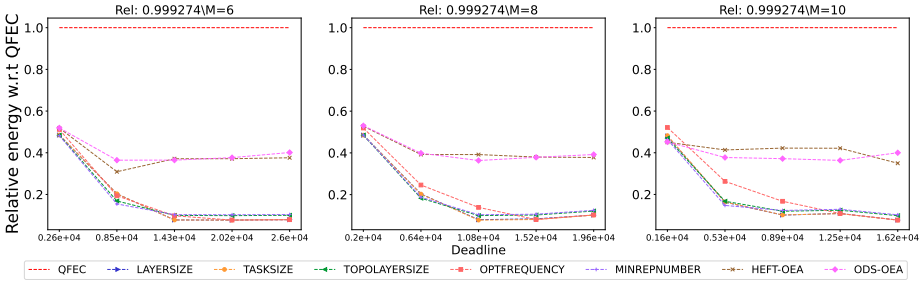

Figure 2333: Assessing the performance of different processor number on the Cycles workflow.

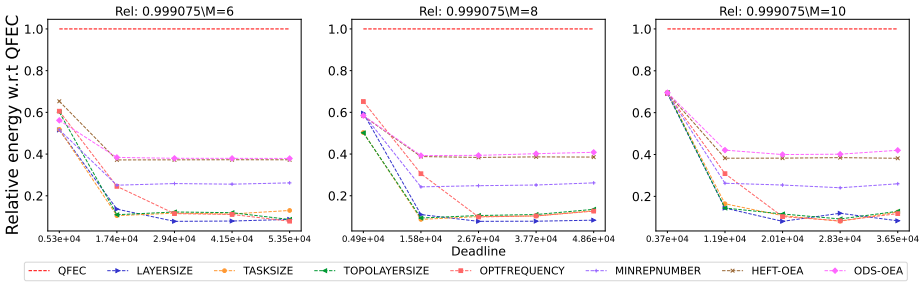

Figure 2334: Assessing the performance of different processor number on the Epigenomics workflow.

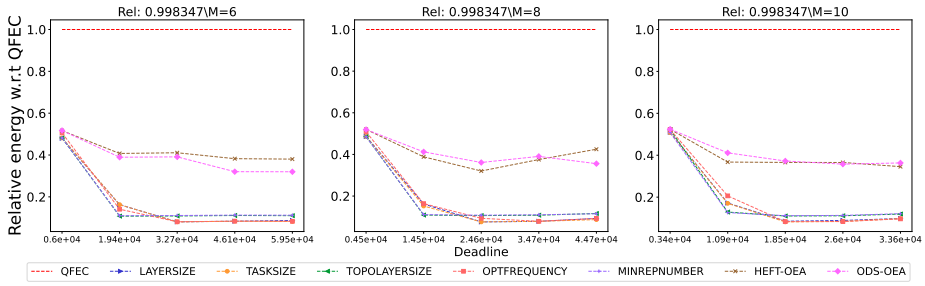

Figure 2335: Assessing the performance of different processor number on the Genome workflow.

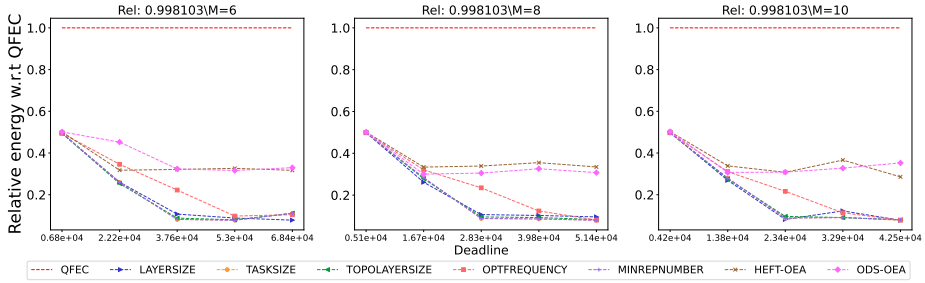

Figure 2336: Assessing the performance of different processor number on the LU.

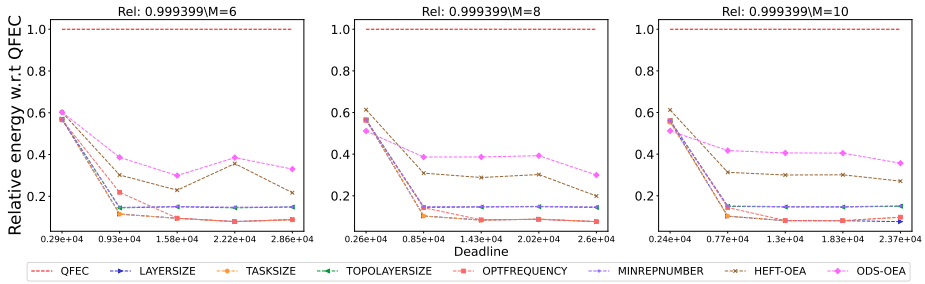

Figure 2337: Assessing the performance of different processor number on the Montage workflow.

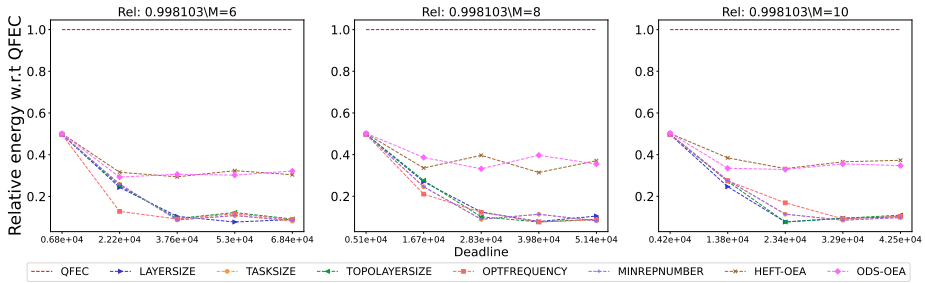

Figure 2338: Assessing the performance of different processor number on the QR.

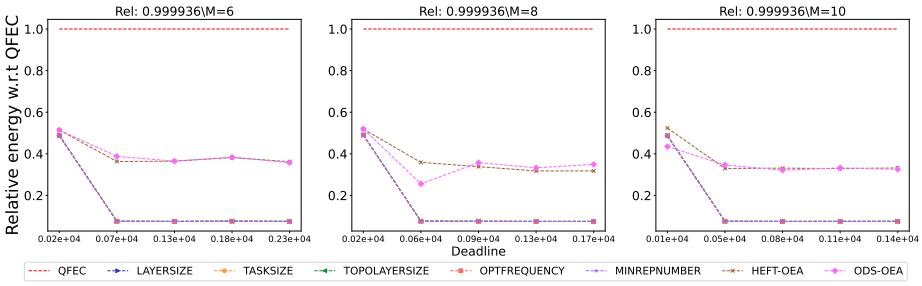

Figure 2339: Assessing the performance of different processor number on the Seismology workflow.

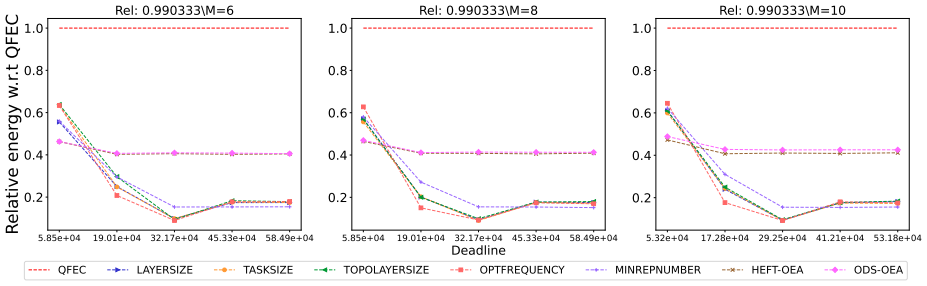

Figure 2340: Assessing the performance of different processor number on the SoyKB workflow.

H.3.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

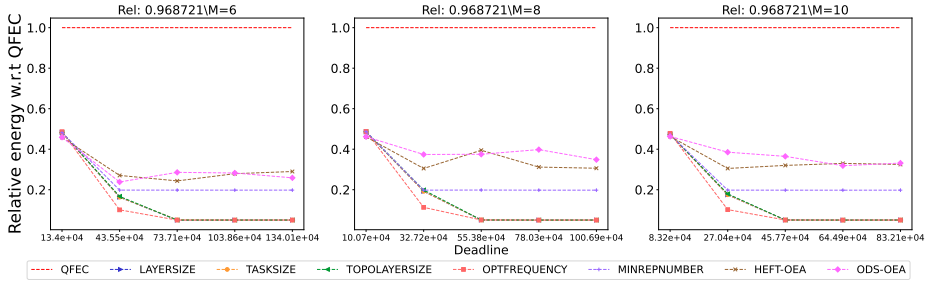

Figure 2341: Assessing the performance of different processor number on the BLAST workflow.

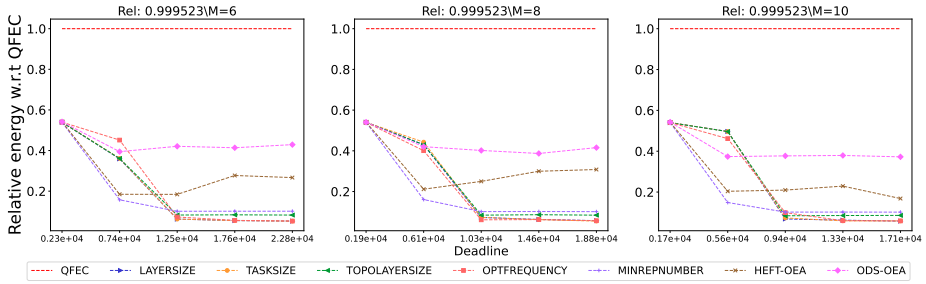

Figure 2342: Assessing the performance of different processor number on the BWA workflow.

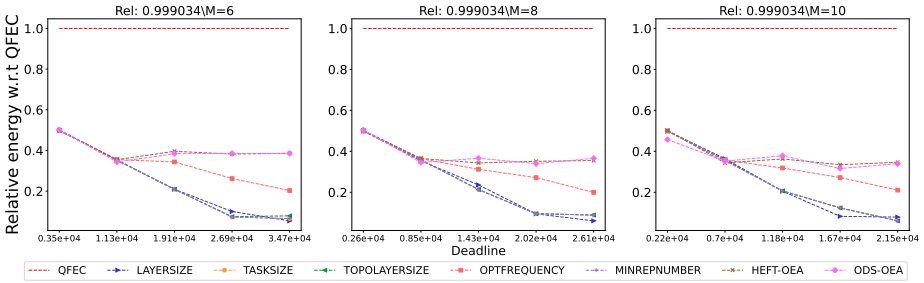

Figure 2343: Assessing the performance of different processor number on the Cholesky workflow.

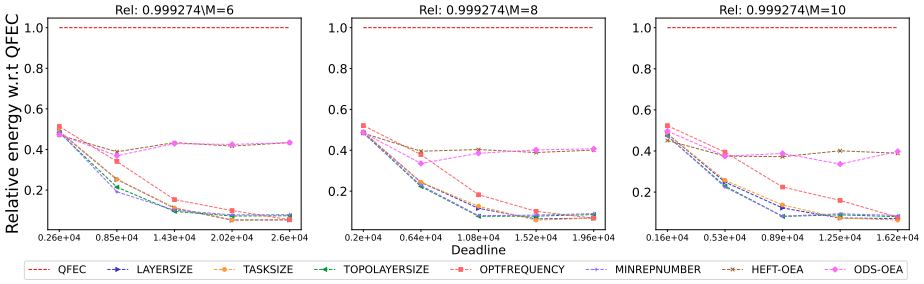

Figure 2344: Assessing the performance of different processor number on the Cycles workflow.

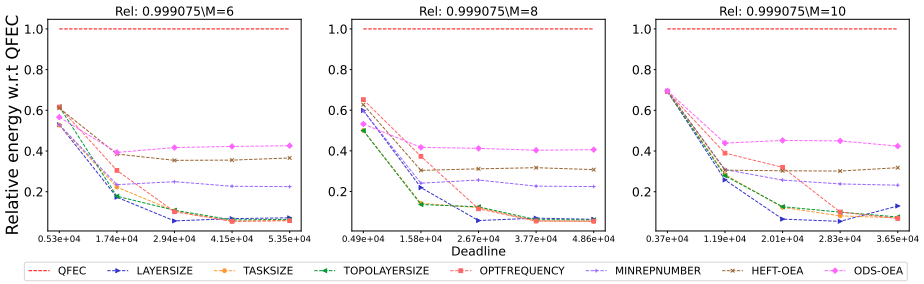

Figure 2345: Assessing the performance of different processor number on the Epigenomics workflow.

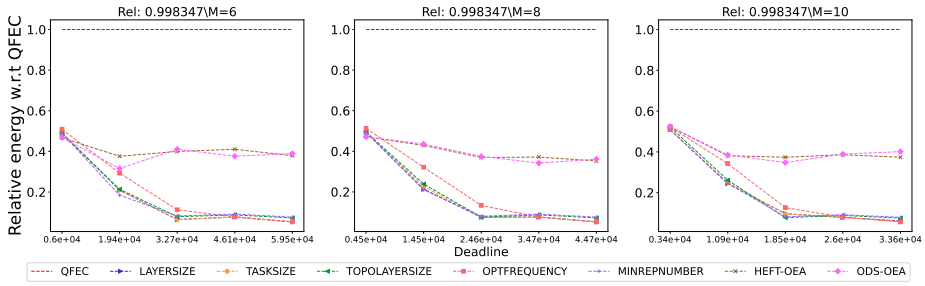

Figure 2346: Assessing the performance of different processor number on the Genome workflow.

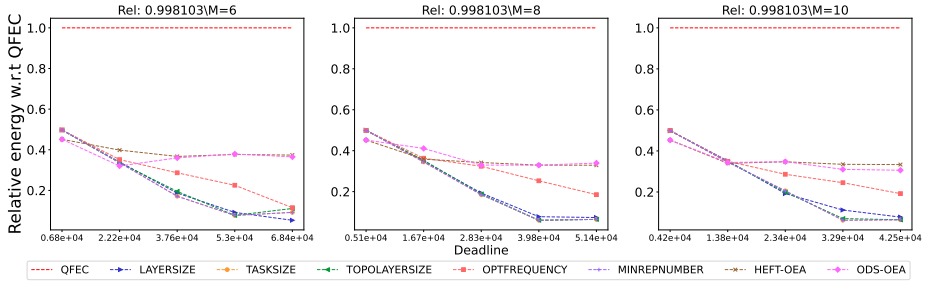

Figure 2347: Assessing the performance of different processor number on the LU.

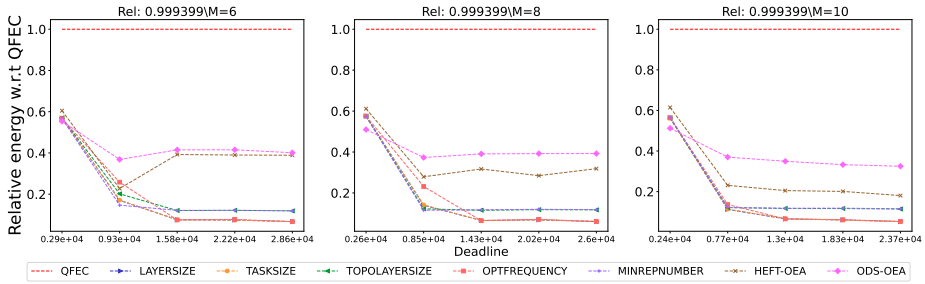

Figure 2348: Assessing the performance of different processor number on the Montage workflow.

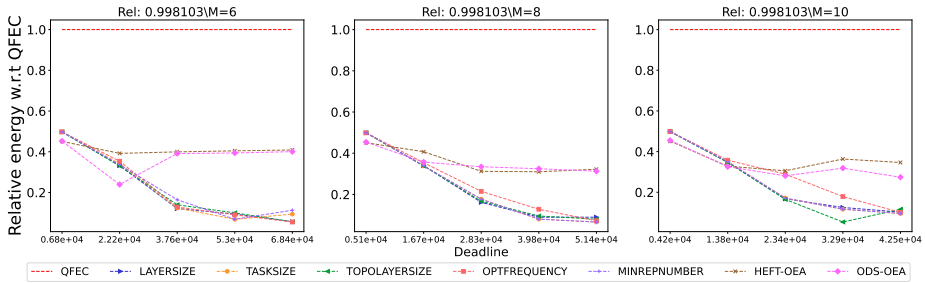

Figure 2349: Assessing the performance of different processor number on the QR.

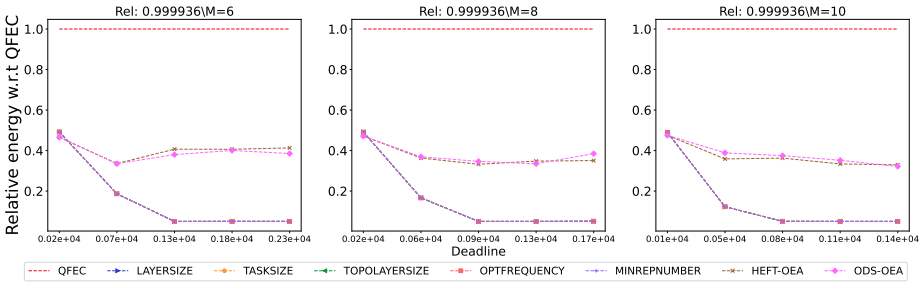

Figure 2350: Assessing the performance of different processor number on the Seismology workflow.

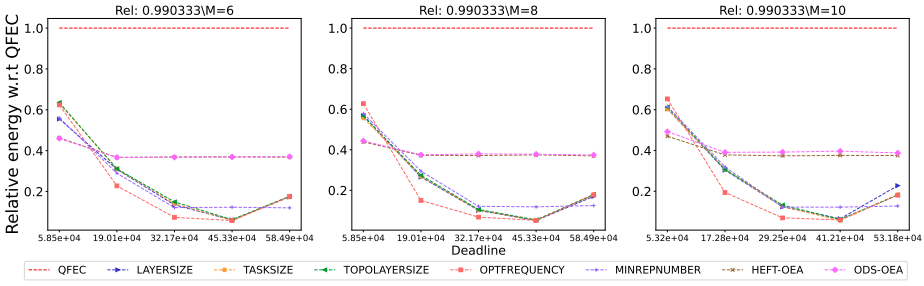

Figure 2351: Assessing the performance of different processor number on the SoyKB workflow.

H.3.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

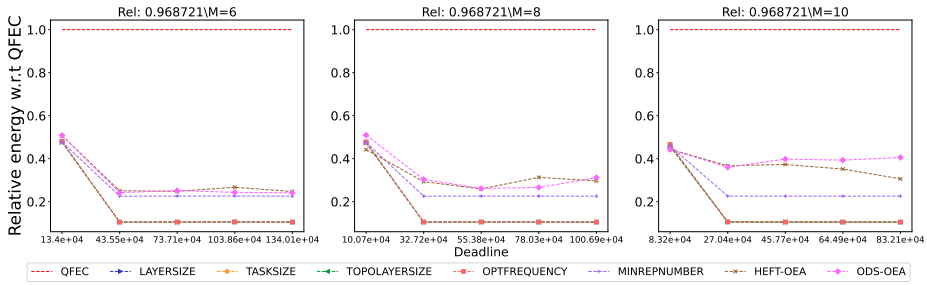

Figure 2352: Assessing the performance of different processor number on the BLAST workflow.

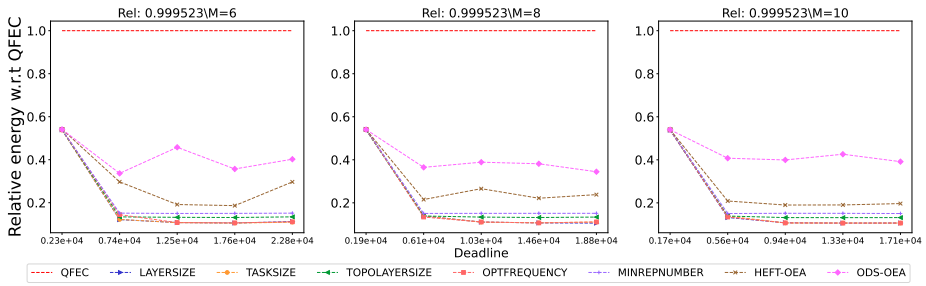

Figure 2353: Assessing the performance of different processor number on the BWA workflow.

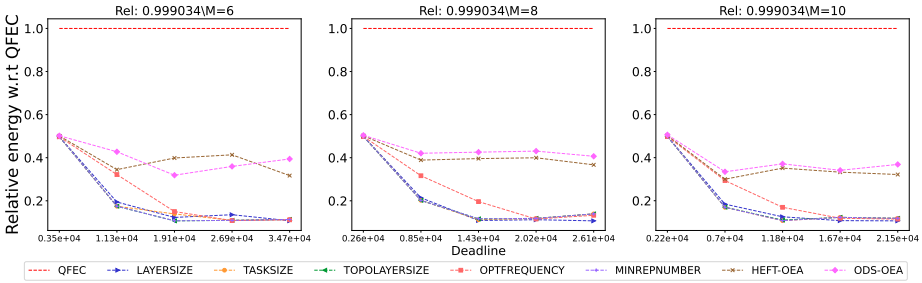

Figure 2354: Assessing the performance of different processor number on the Cholesky workflow.

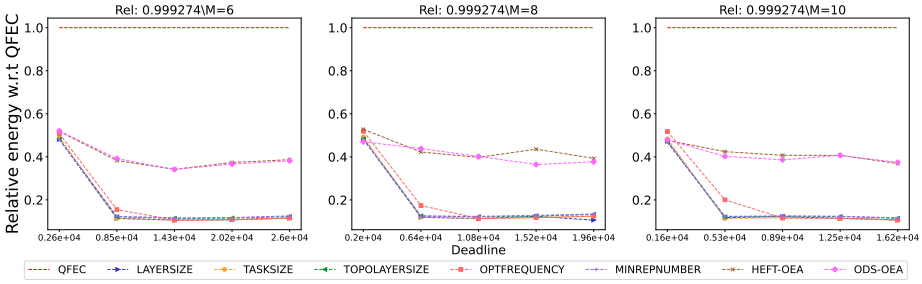

Figure 2355: Assessing the performance of different processor number on the Cycles workflow.

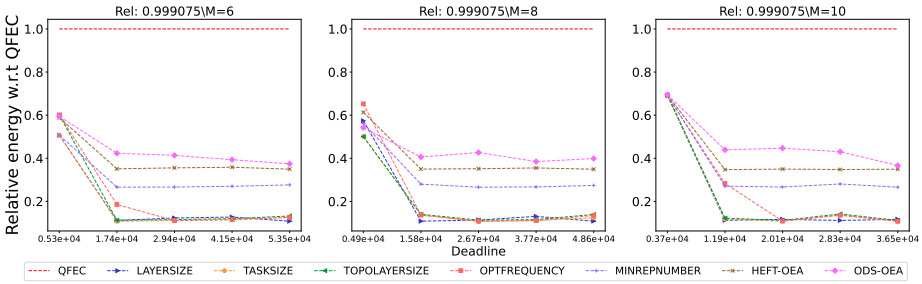

Figure 2356: Assessing the performance of different processor number on the Epigenomics workflow.

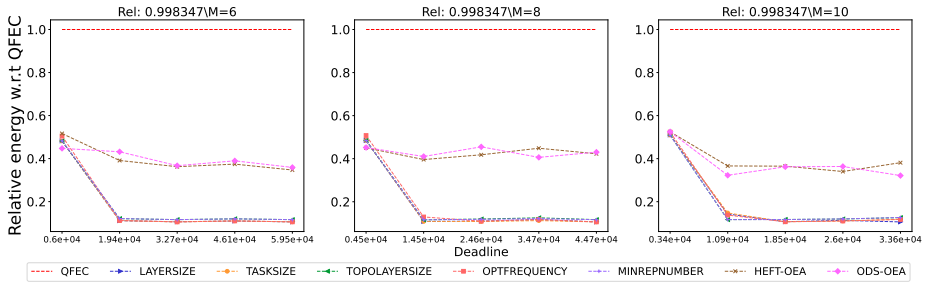

Figure 2357: Assessing the performance of different processor number on the Genome workflow.

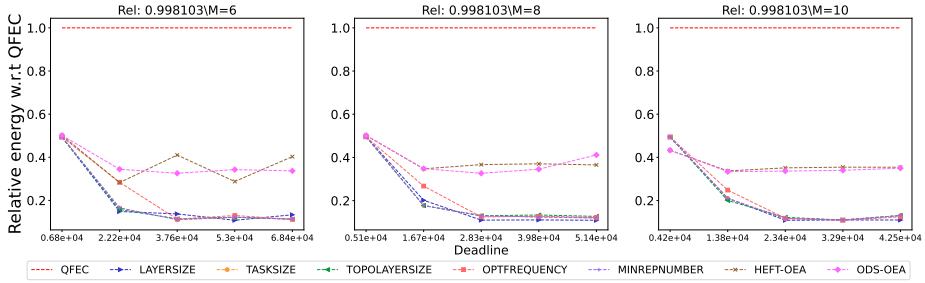

Figure 2358: Assessing the performance of different processor number on the LU.

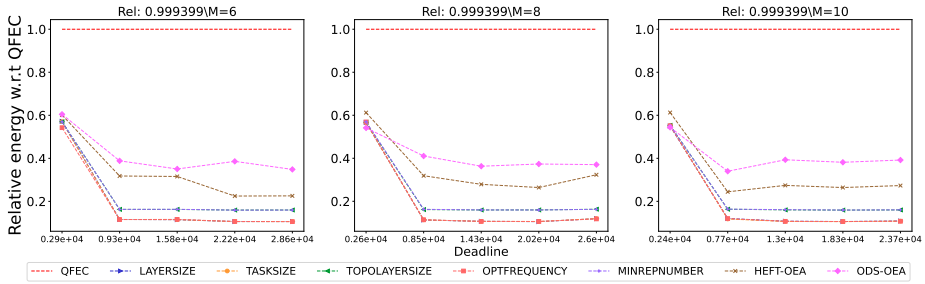

Figure 2359: Assessing the performance of different processor number on the Montage workflow.

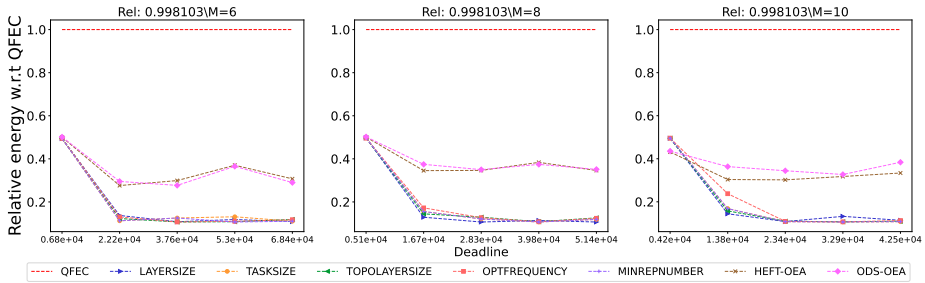

Figure 2360: Assessing the performance of different processor number on the QR.

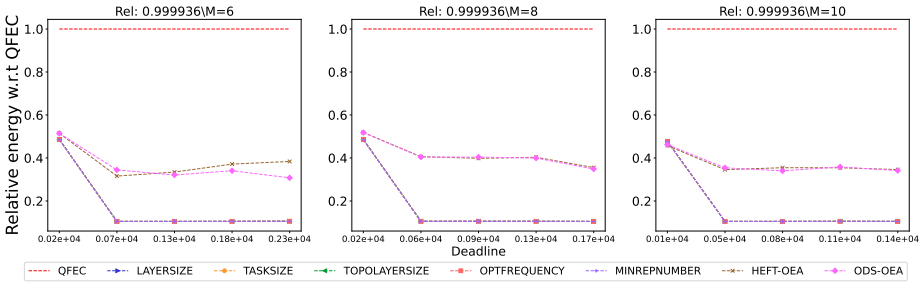

Figure 2361: Assessing the performance of different processor number on the Seismology workflow.

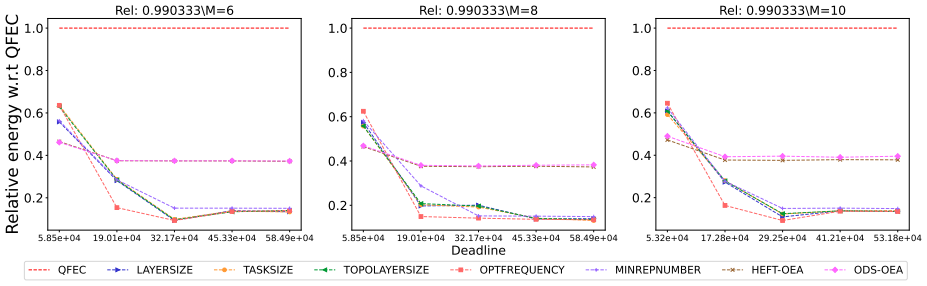

Figure 2362: Assessing the performance of different processor number on the SoyKB workflow.

H.3.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

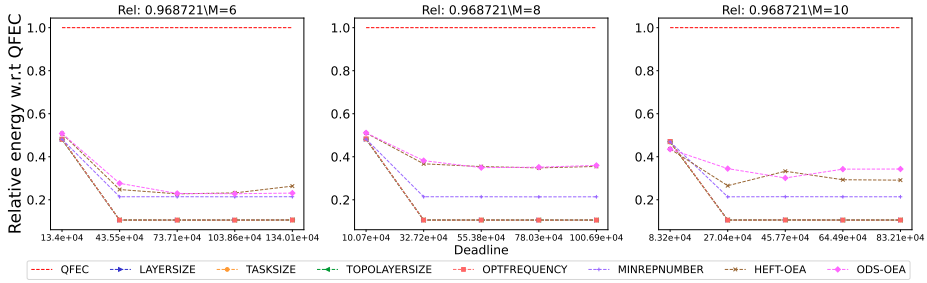

Figure 2363: Assessing the performance of different processor number on the BLAST workflow.

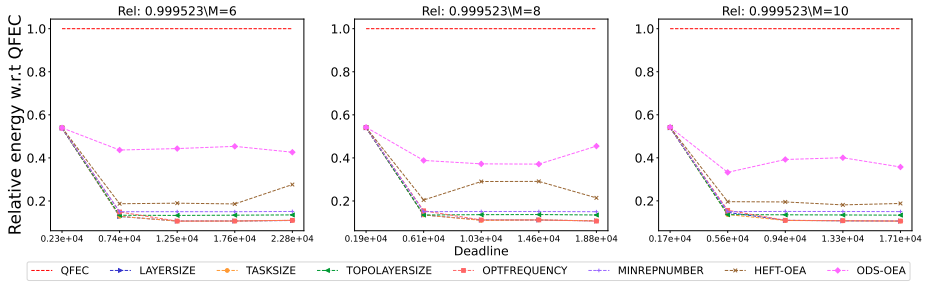

Figure 2364: Assessing the performance of different processor number on the BWA workflow.

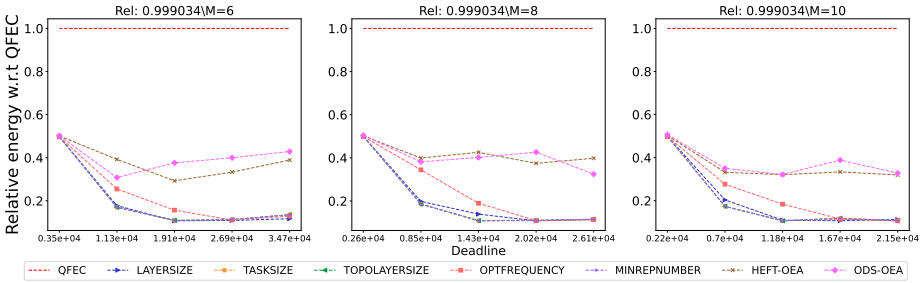

Figure 2365: Assessing the performance of different processor number on the Cholesky workflow.

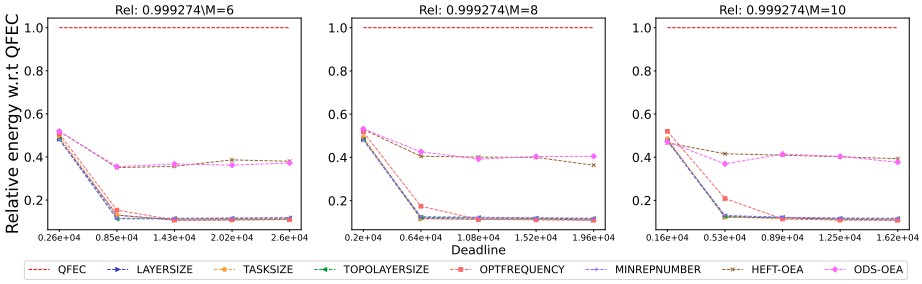

Figure 2366: Assessing the performance of different processor number on the Cycles workflow.

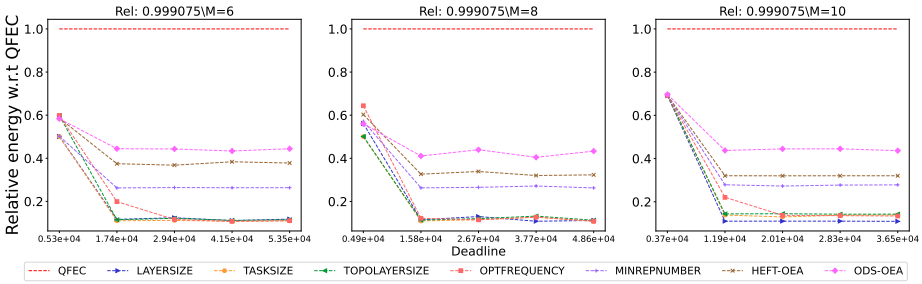

Figure 2367: Assessing the performance of different processor number on the Epigenomics workflow.

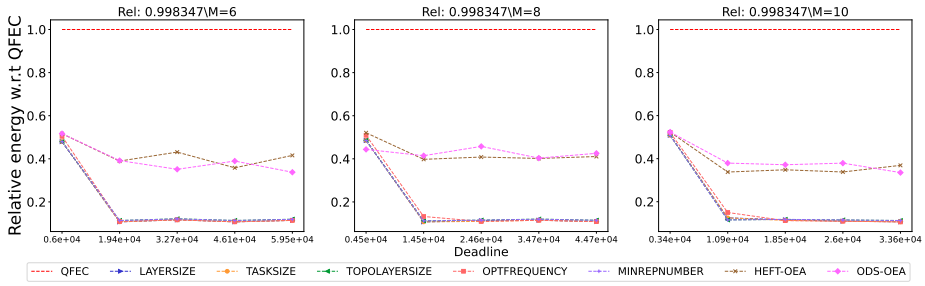

Figure 2368: Assessing the performance of different processor number on the Genome workflow.

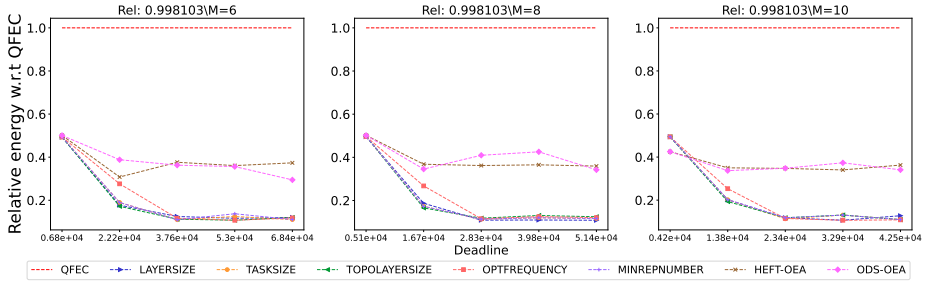

Figure 2369: Assessing the performance of different processor number on the LU.

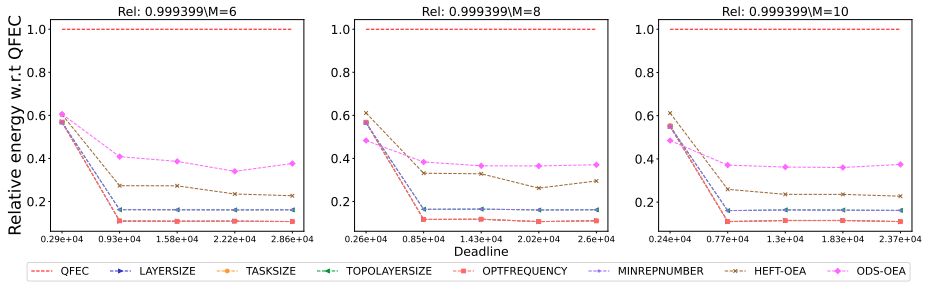

Figure 2370: Assessing the performance of different processor number on the Montage workflow.

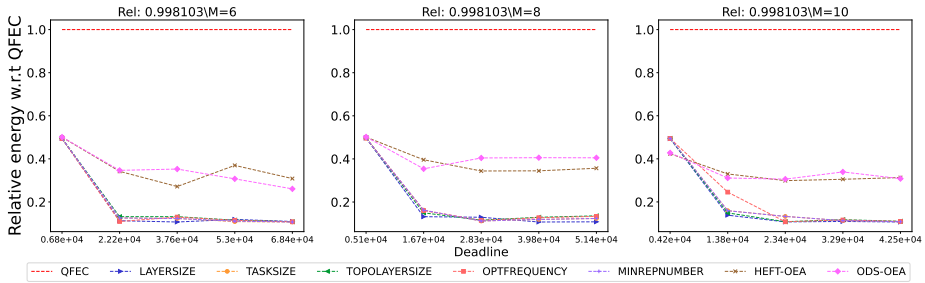

Figure 2371: Assessing the performance of different processor number on the QR.

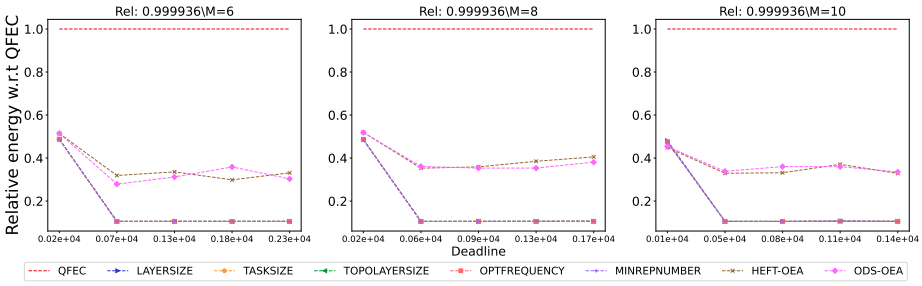

Figure 2372: Assessing the performance of different processor number on the Seismology workflow.

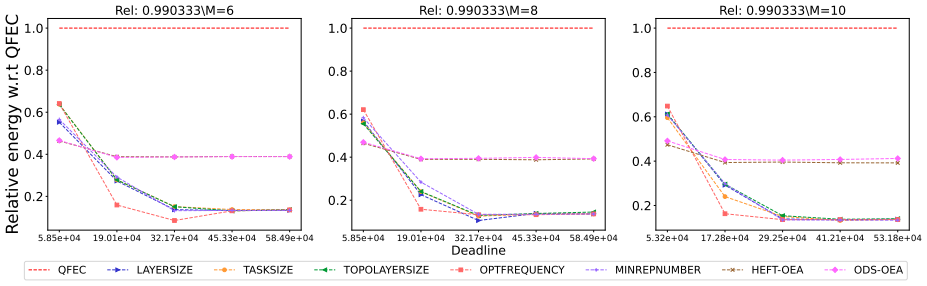

Figure 2373: Assessing the performance of different processor number on the SoyKB workflow.

H.4 $BC/WC = 0.4$

H.4.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

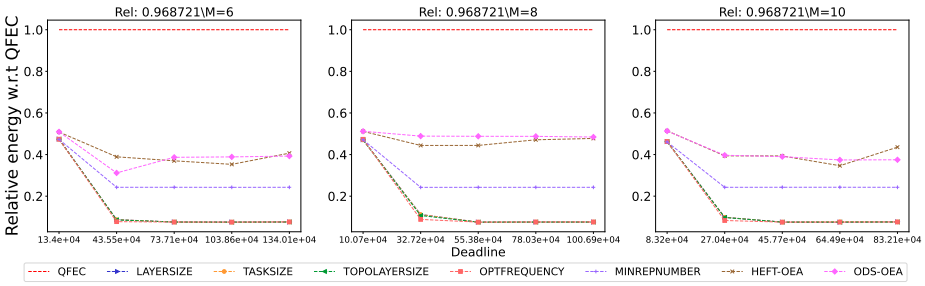

Figure 2374: Assessing the performance of different processor number on the BLAST workflow.

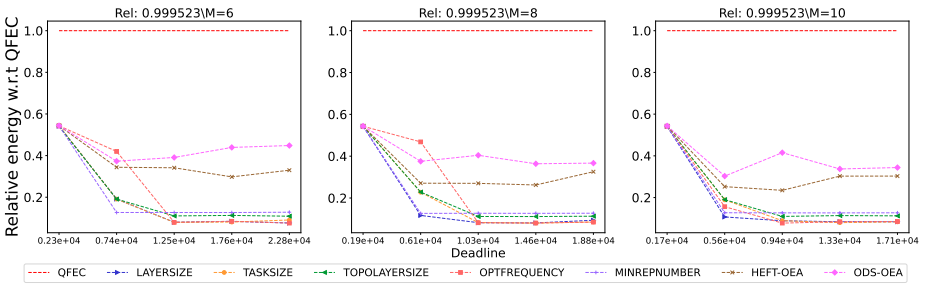

Figure 2375: Assessing the performance of different processor number on the BWA workflow.

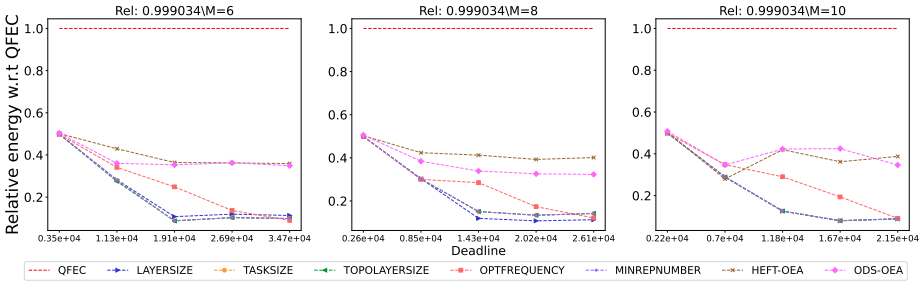

Figure 2376: Assessing the performance of different processor number on the Cholesky workflow.

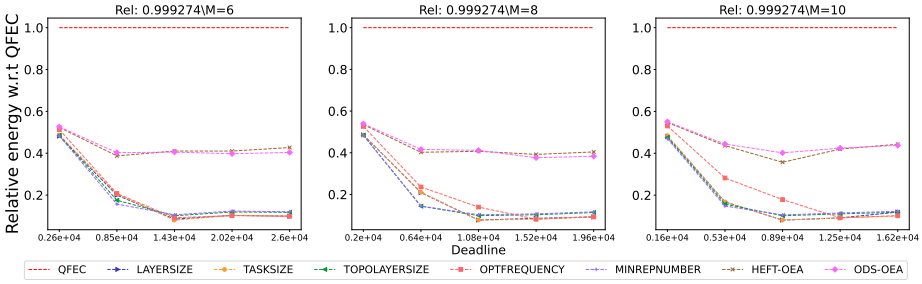

Figure 2377: Assessing the performance of different processor number on the Cycles workflow.

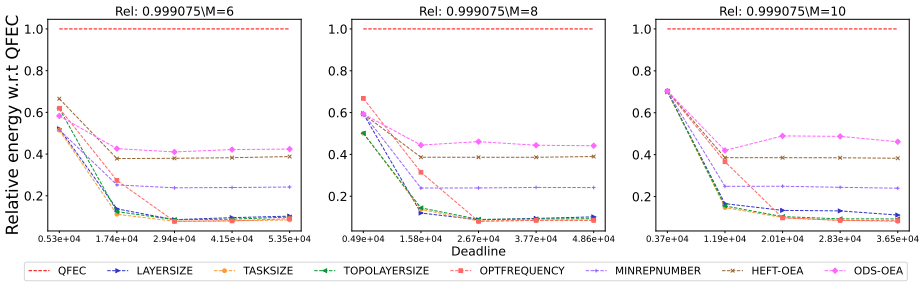

Figure 2378: Assessing the performance of different processor number on the Epigenomics workflow.

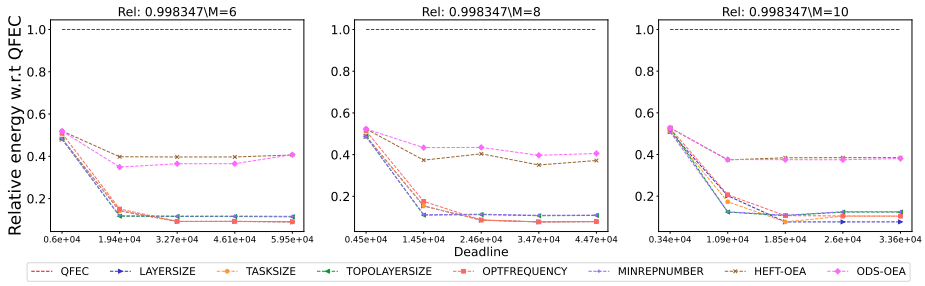

Figure 2379: Assessing the performance of different processor number on the Genome workflow.

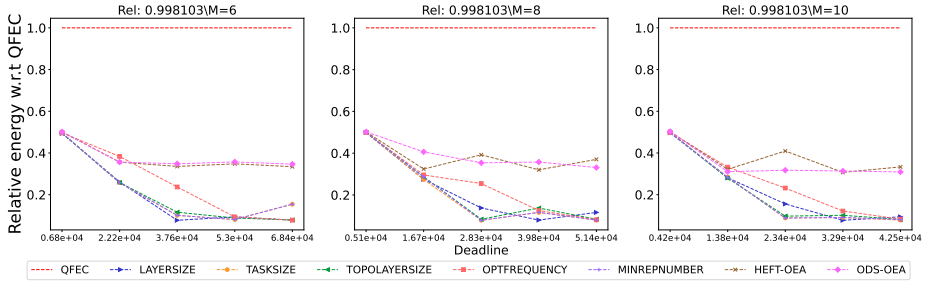

Figure 2380: Assessing the performance of different processor number on the LU.

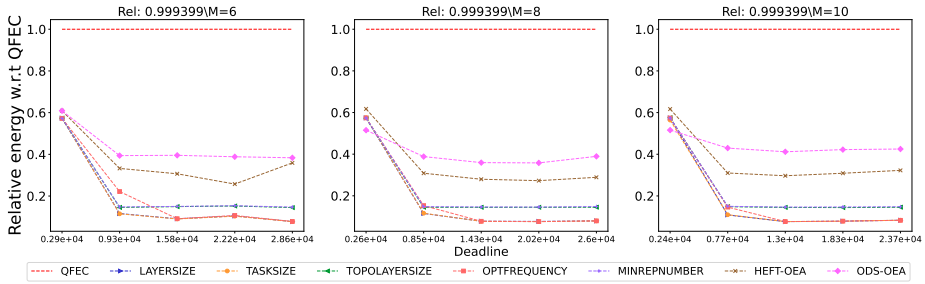

Figure 2381: Assessing the performance of different processor number on the Montage workflow.

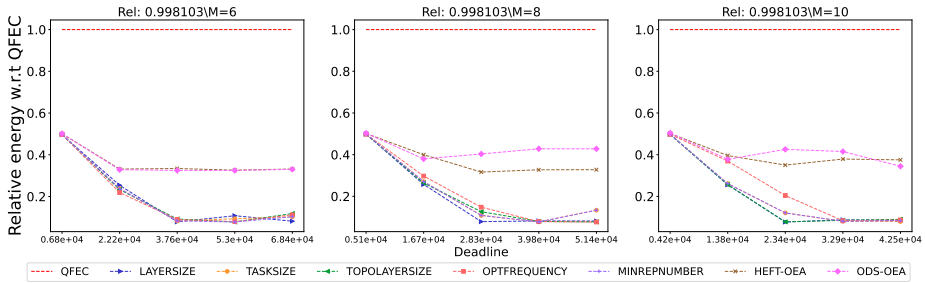

Figure 2382: Assessing the performance of different processor number on the QR.

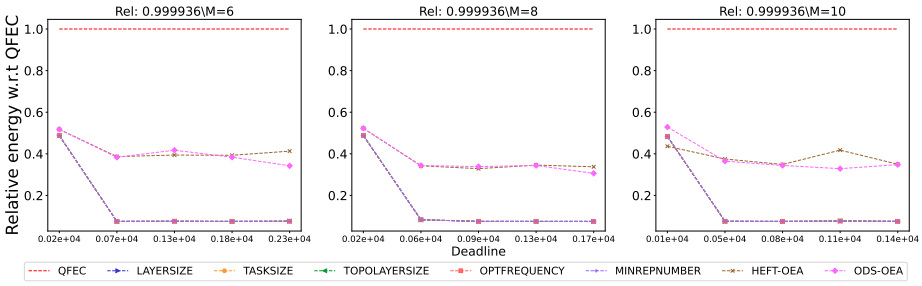

Figure 2383: Assessing the performance of different processor number on the Seismology workflow.

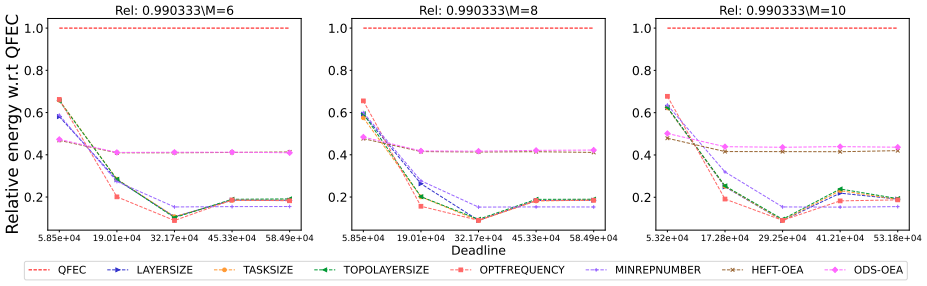

Figure 2384: Assessing the performance of different processor number on the SoyKB workflow.

H.4.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

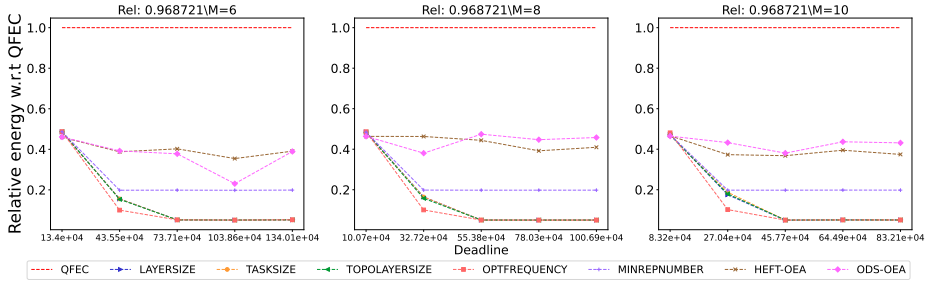

Figure 2385: Assessing the performance of different processor number on the BLAST workflow.

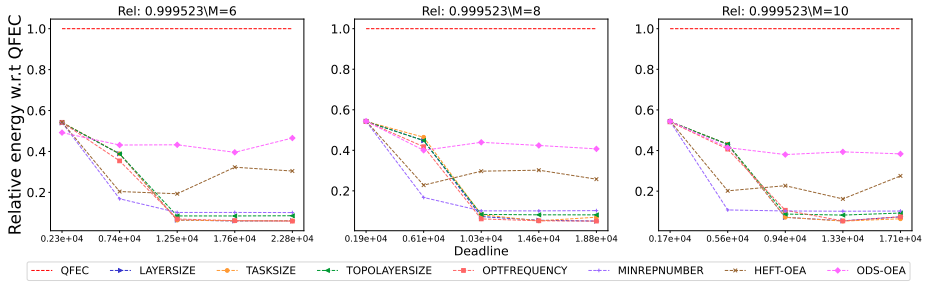

Figure 2386: Assessing the performance of different processor number on the BWA workflow.

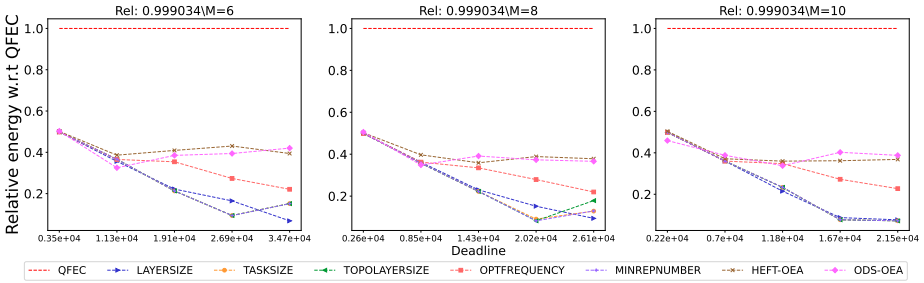

Figure 2387: Assessing the performance of different processor number on the Cholesky workflow.

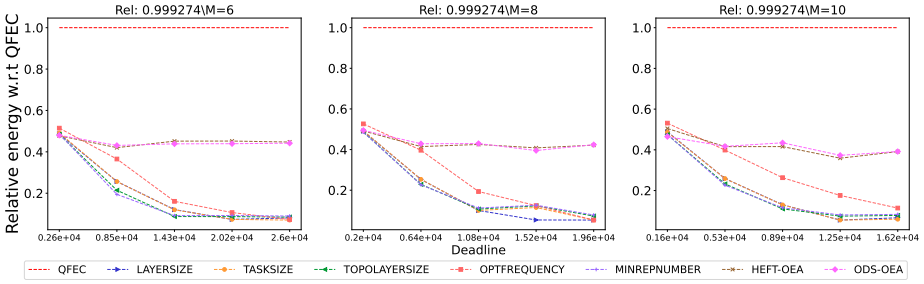

Figure 2388: Assessing the performance of different processor number on the Cycles workflow.

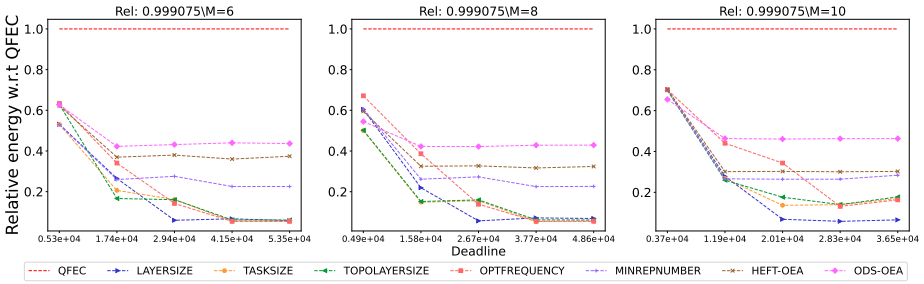

Figure 2389: Assessing the performance of different processor number on the Epigenomics workflow.

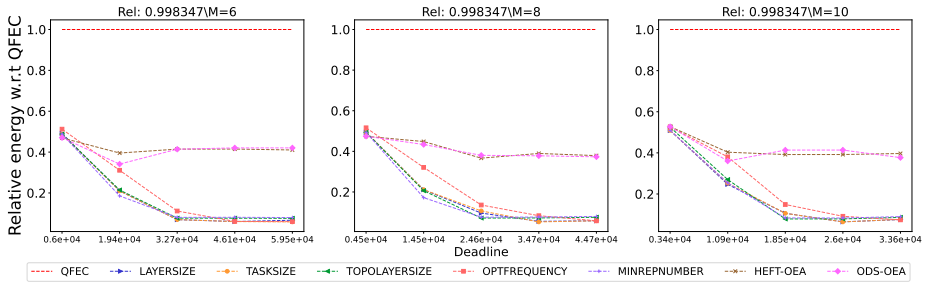

Figure 2390: Assessing the performance of different processor number on the Genome workflow.

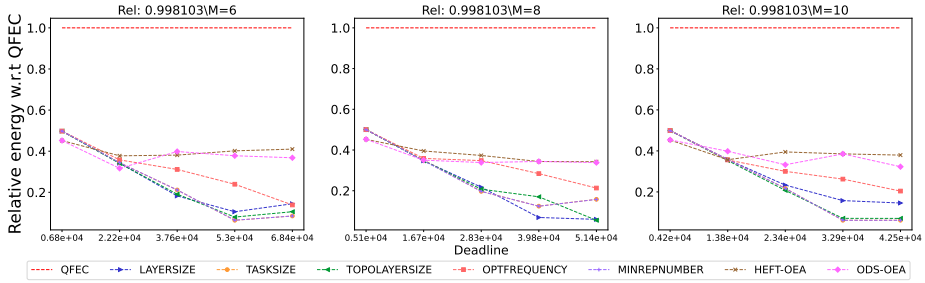

Figure 2391: Assessing the performance of different processor number on the LU.

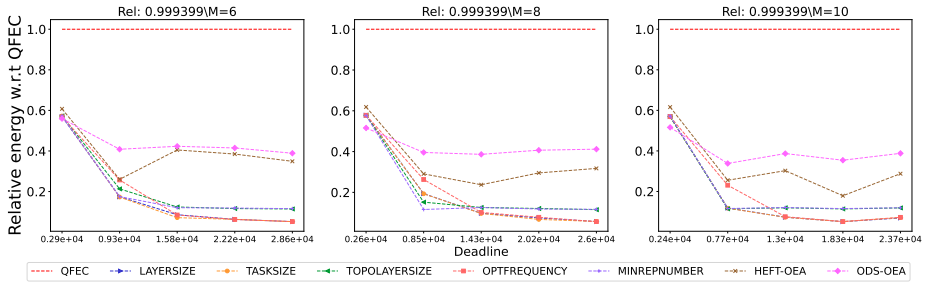

Figure 2392: Assessing the performance of different processor number on the Montage workflow.

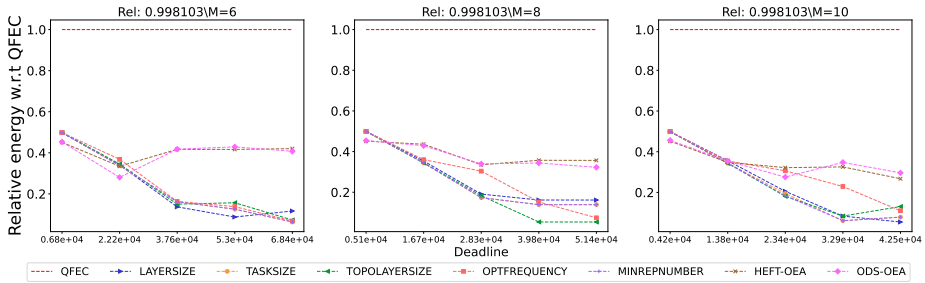

Figure 2393: Assessing the performance of different processor number on the QR.

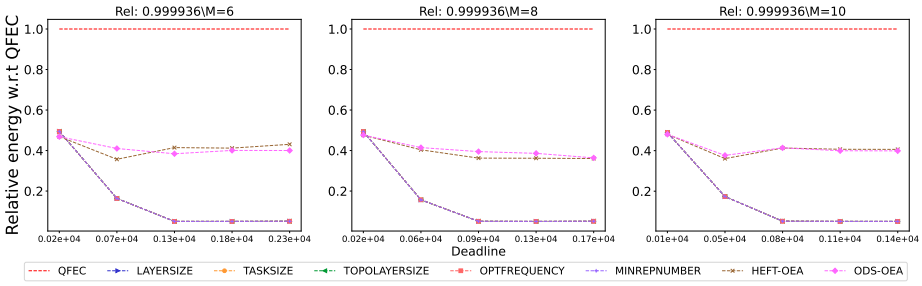

Figure 2394: Assessing the performance of different processor number on the Seismology workflow.

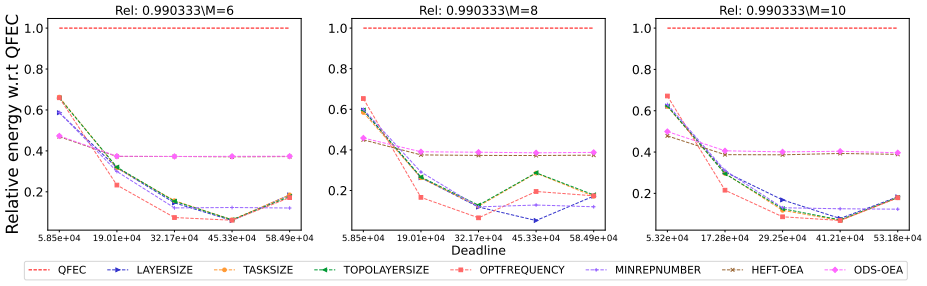

Figure 2395: Assessing the performance of different processor number on the SoyKB workflow.

H.4.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

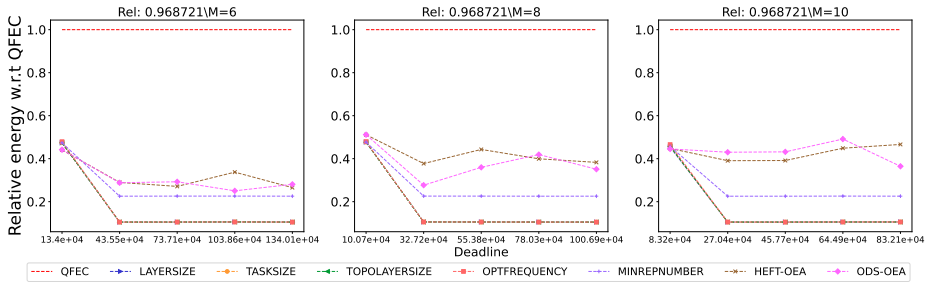

Figure 2396: Assessing the performance of different processor number on the BLAST workflow.

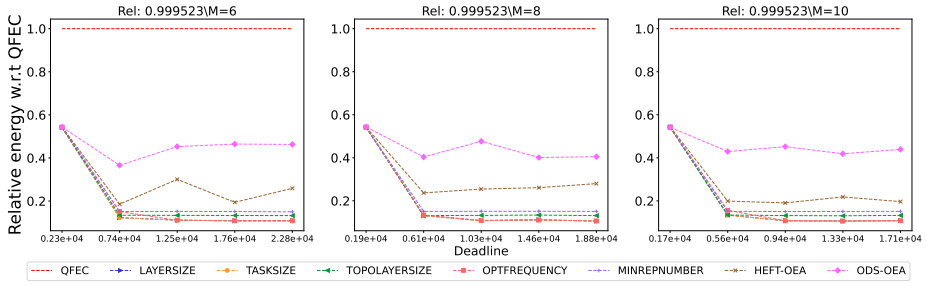

Figure 2397: Assessing the performance of different processor number on the BWA workflow.

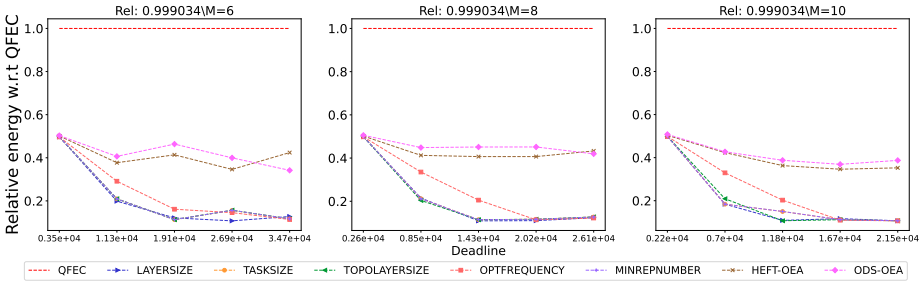

Figure 2398: Assessing the performance of different processor number on the Cholesky workflow.

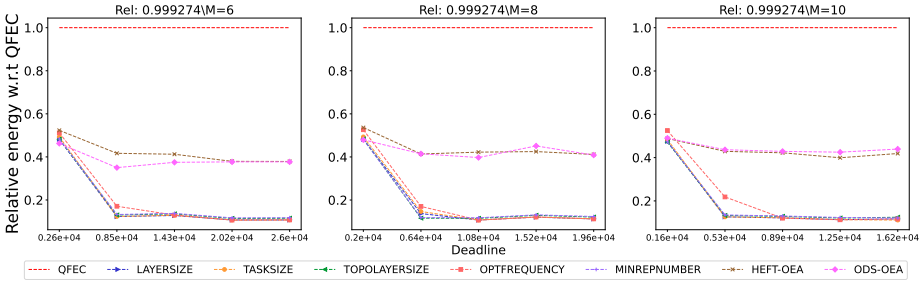

Figure 2399: Assessing the performance of different processor number on the Cycles workflow.

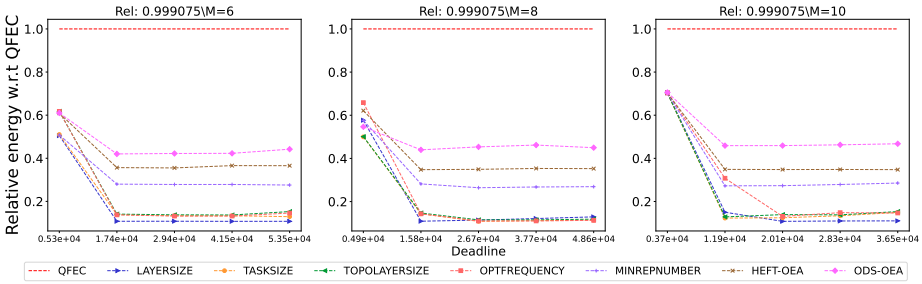

Figure 2400: Assessing the performance of different processor number on the Epigenomics workflow.

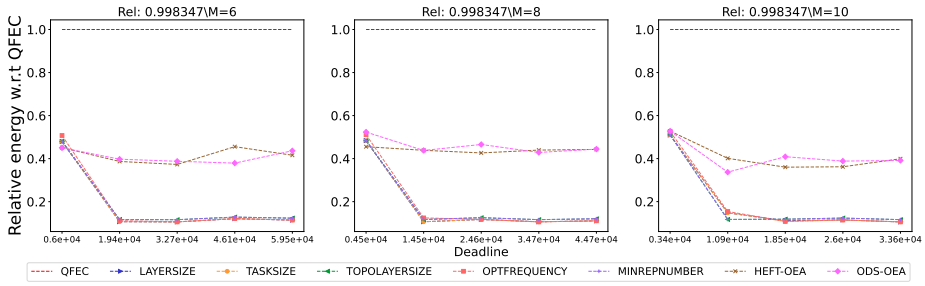

Figure 2401: Assessing the performance of different processor number on the Genome workflow.

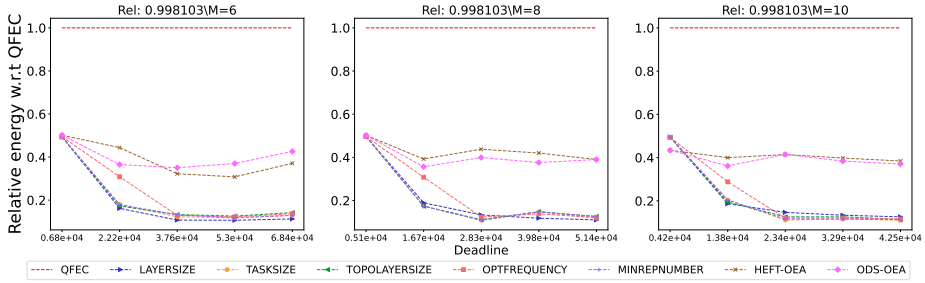

Figure 2402: Assessing the performance of different processor number on the LU.

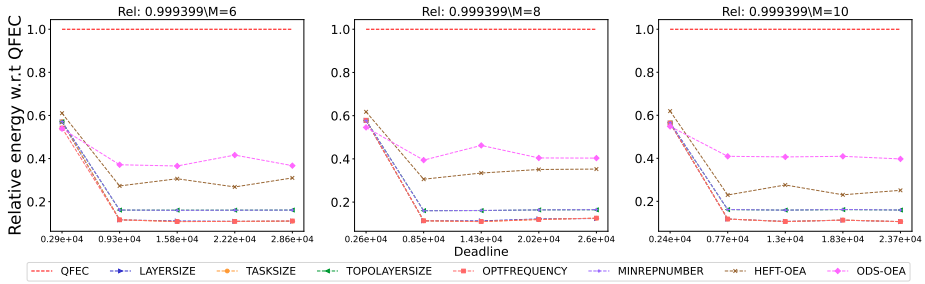

Figure 2403: Assessing the performance of different processor number on the Montage workflow.

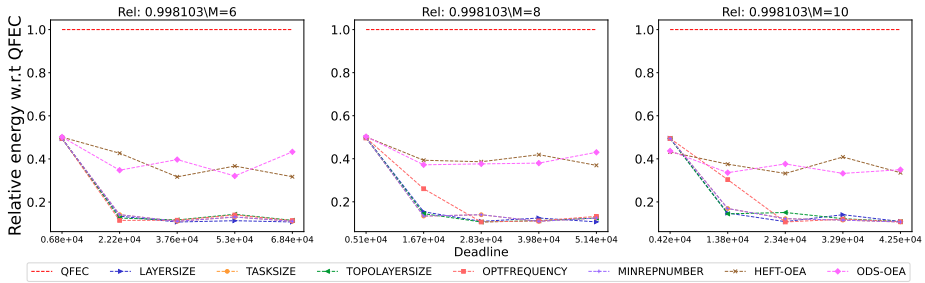

Figure 2404: Assessing the performance of different processor number on the QR.

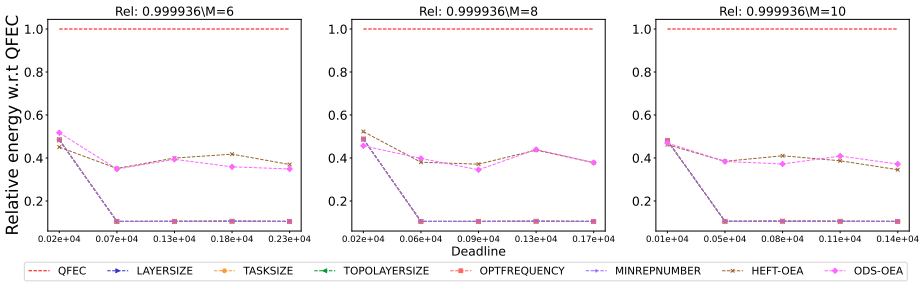

Figure 2405: Assessing the performance of different processor number on the Seismology workflow.

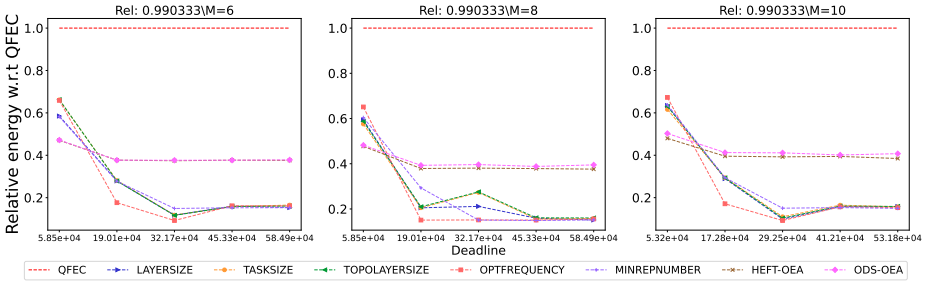

Figure 2406: Assessing the performance of different processor number on the SoyKB workflow.

H.4.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

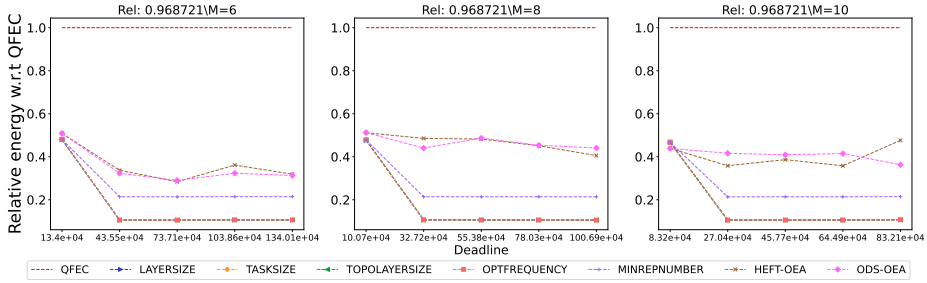

Figure 2407: Assessing the performance of different processor number on the BLAST workflow.

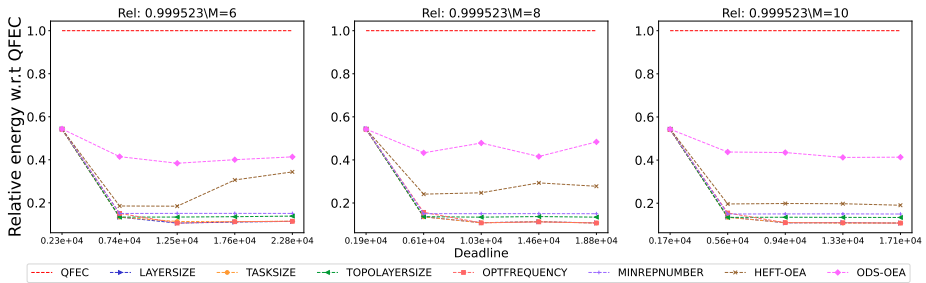

Figure 2408: Assessing the performance of different processor number on the BWA workflow.

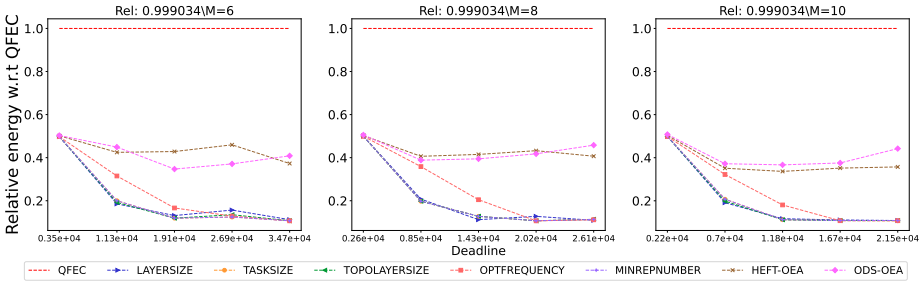

Figure 2409: Assessing the performance of different processor number on the Cholesky workflow.

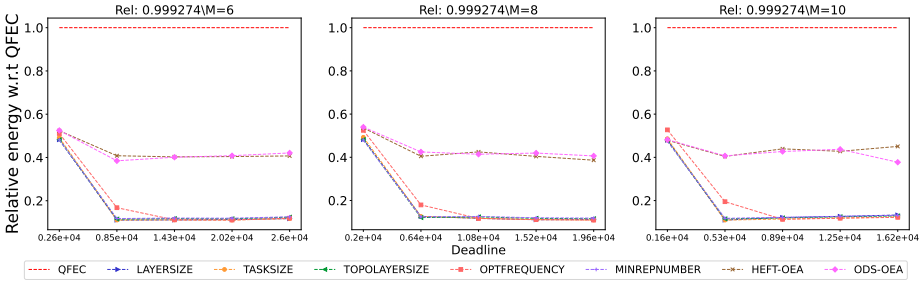

Figure 2410: Assessing the performance of different processor number on the Cycles workflow.

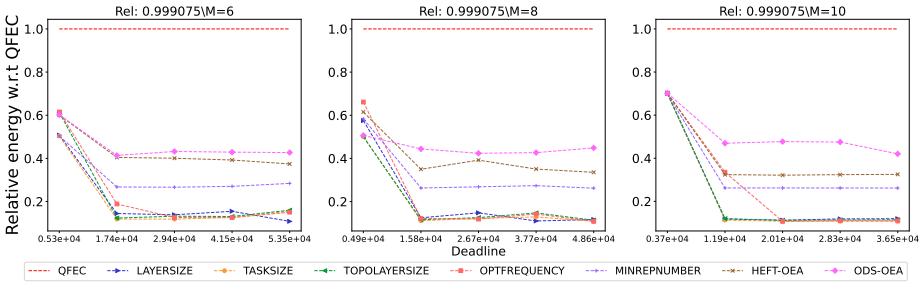

Figure 2411: Assessing the performance of different processor number on the Epigenomics workflow.

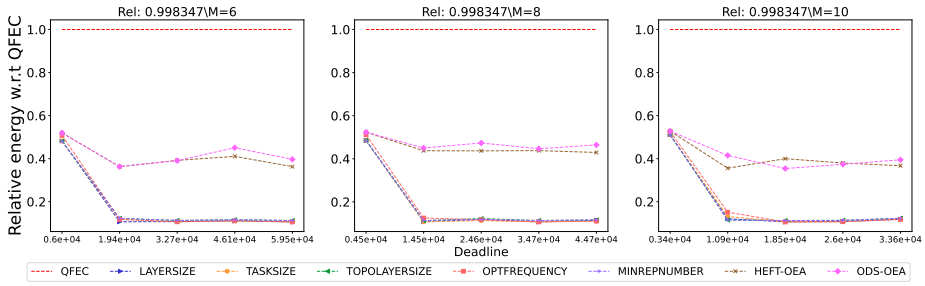

Figure 2412: Assessing the performance of different processor number on the Genome workflow.

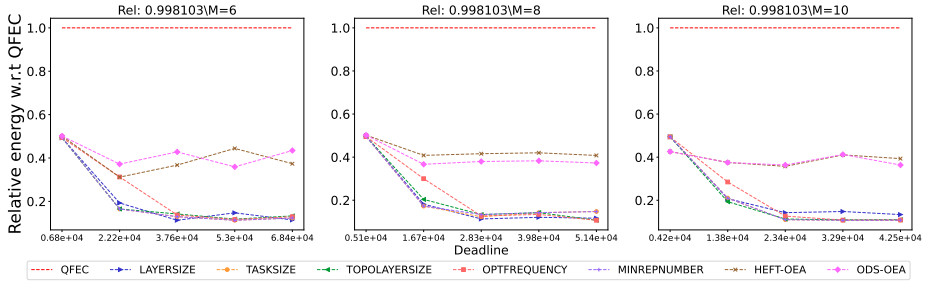

Figure 2413: Assessing the performance of different processor number on the LU.

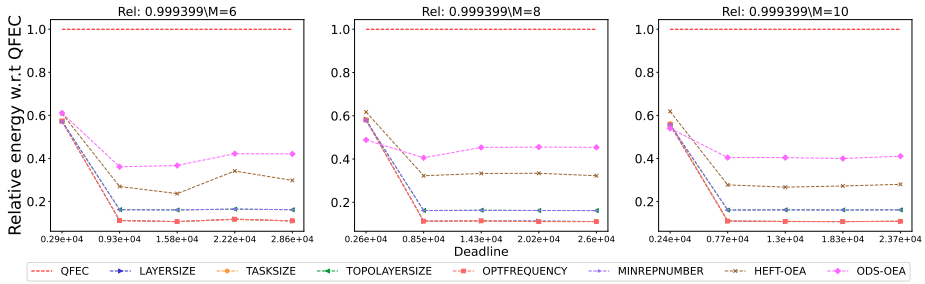

Figure 2414: Assessing the performance of different processor number on the Montage workflow.

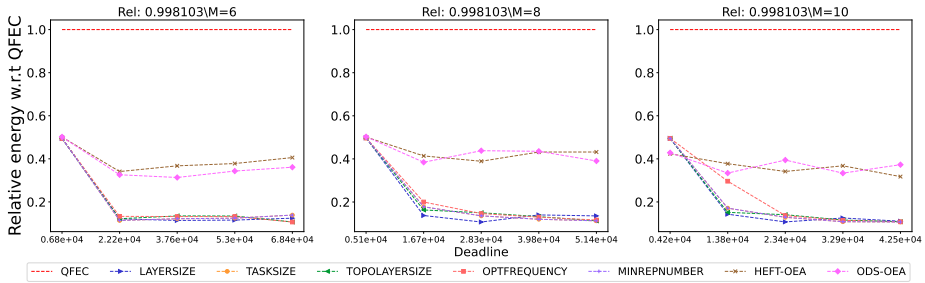

Figure 2415: Assessing the performance of different processor number on the QR.

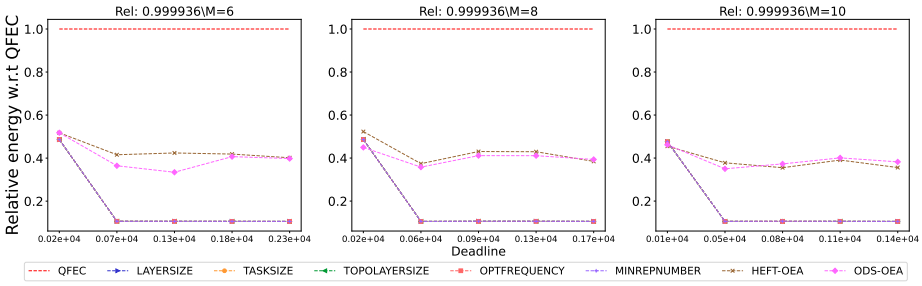

Figure 2416: Assessing the performance of different processor number on the Seismology workflow.

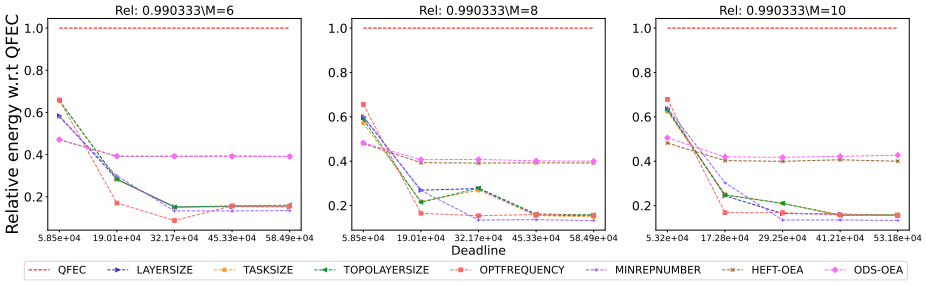

Figure 2417: Assessing the performance of different processor number on the SoyKB workflow.

H.5 $BC/WC = 0.5$

H.5.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

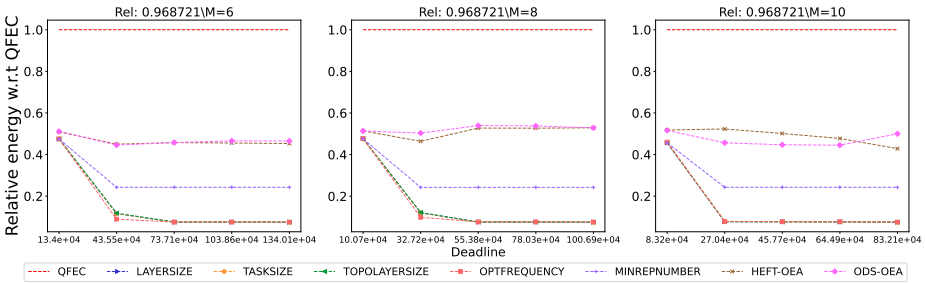

Figure 2418: Assessing the performance of different processor number on the BLAST workflow.

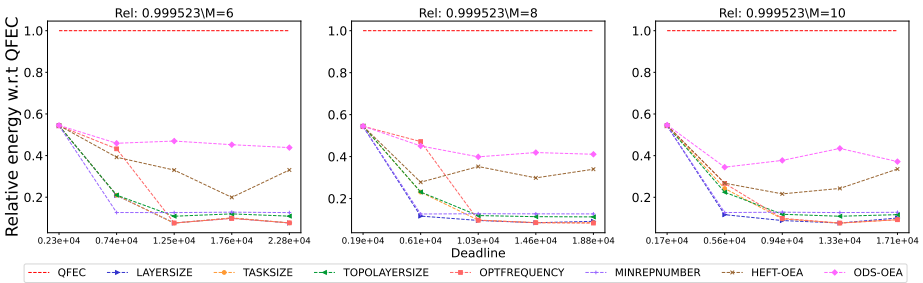

Figure 2419: Assessing the performance of different processor number on the BWA workflow.

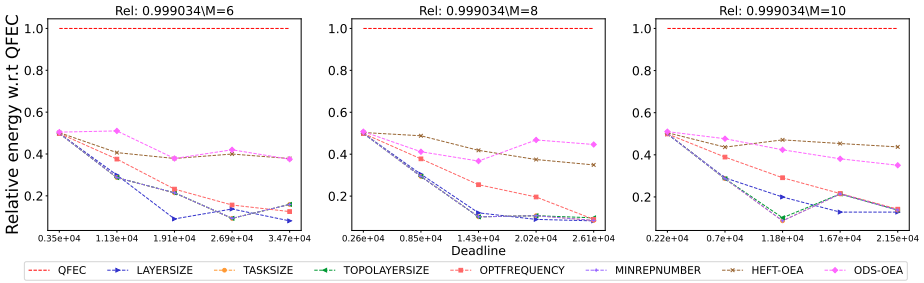

Figure 2420: Assessing the performance of different processor number on the Cholesky workflow.

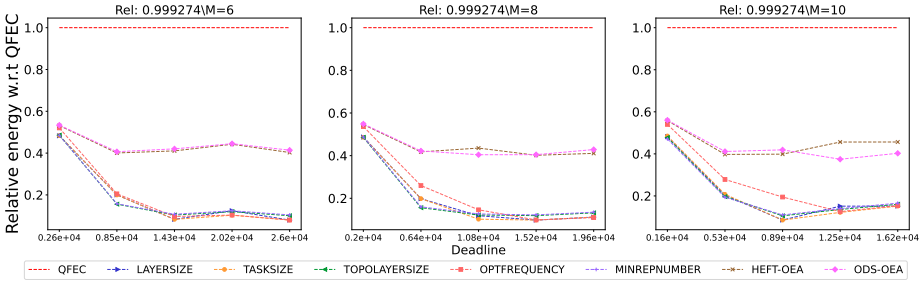

Figure 2421: Assessing the performance of different processor number on the Cycles workflow.

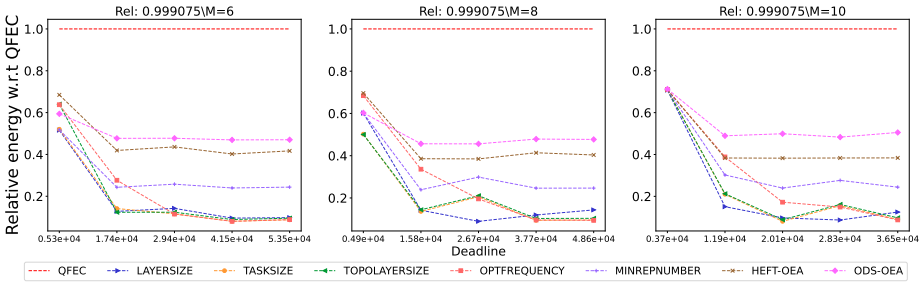

Figure 2422: Assessing the performance of different processor number on the Epigenomics workflow.

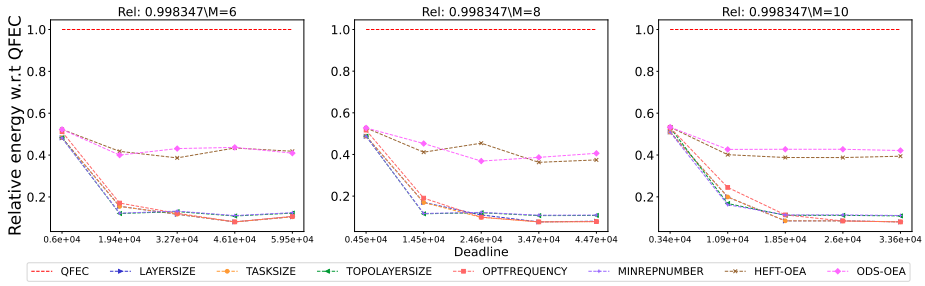

Figure 2423: Assessing the performance of different processor number on the Genome workflow.

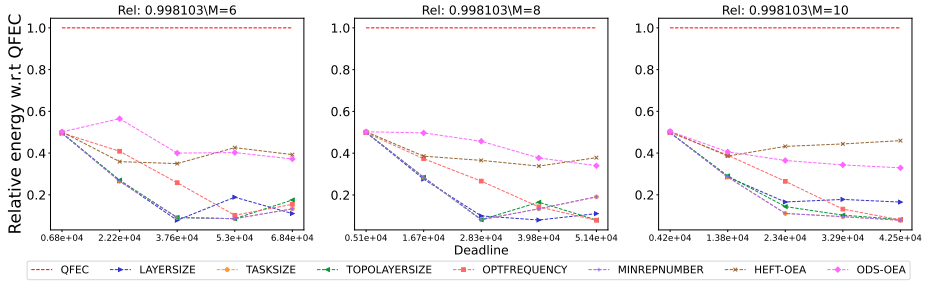

Figure 2424: Assessing the performance of different processor number on the LU.

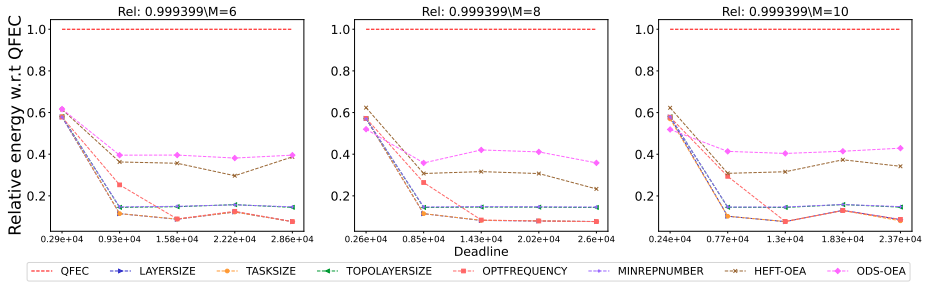

Figure 2425: Assessing the performance of different processor number on the Montage workflow.

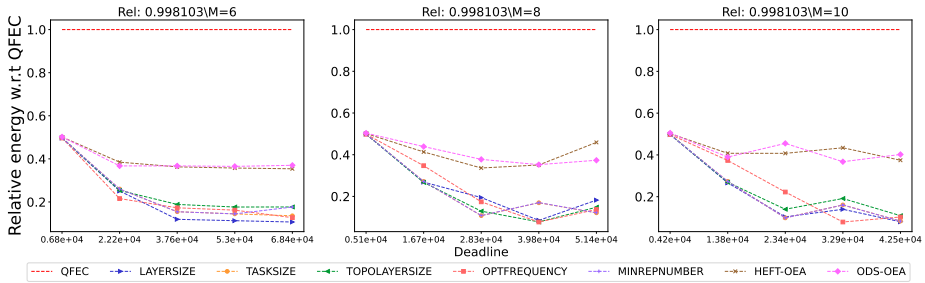

Figure 2426: Assessing the performance of different processor number on the QR.

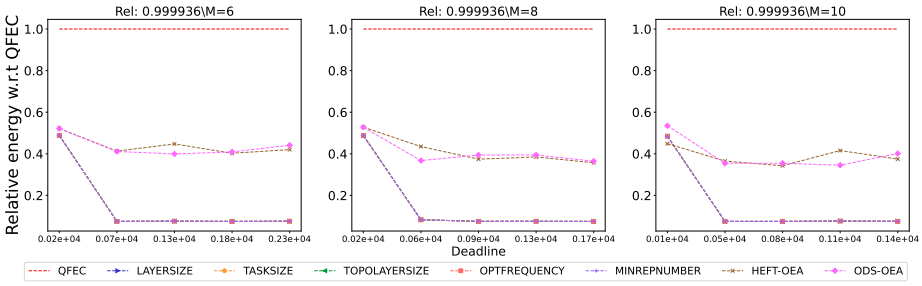

Figure 2427: Assessing the performance of different processor number on the Seismology workflow.

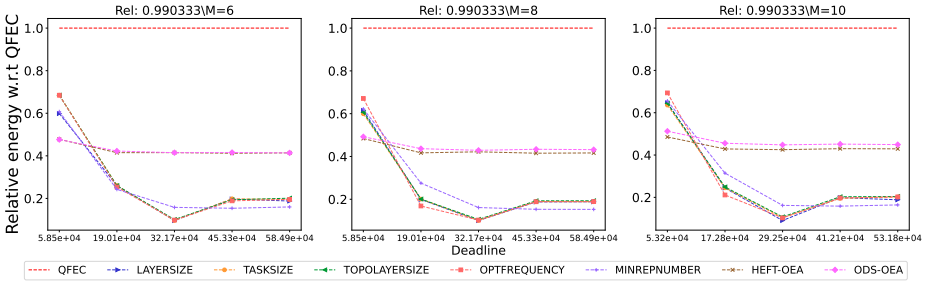

Figure 2428: Assessing the performance of different processor number on the SoyKB workflow.

H.5.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

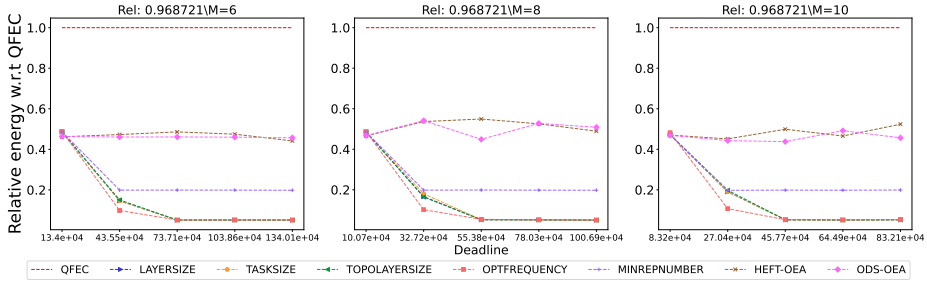

Figure 2429: Assessing the performance of different processor number on the BLAST workflow.

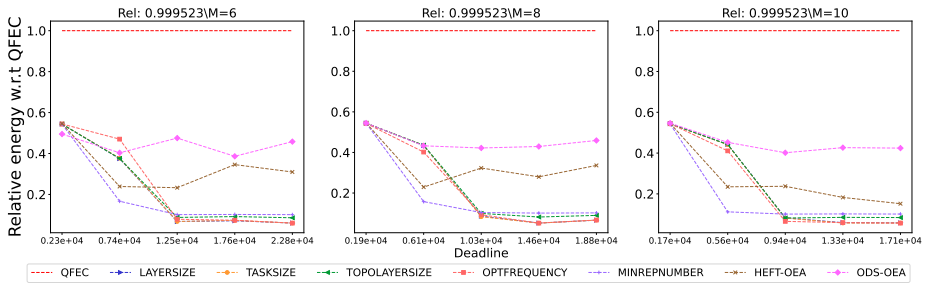

Figure 2430: Assessing the performance of different processor number on the BWA workflow.

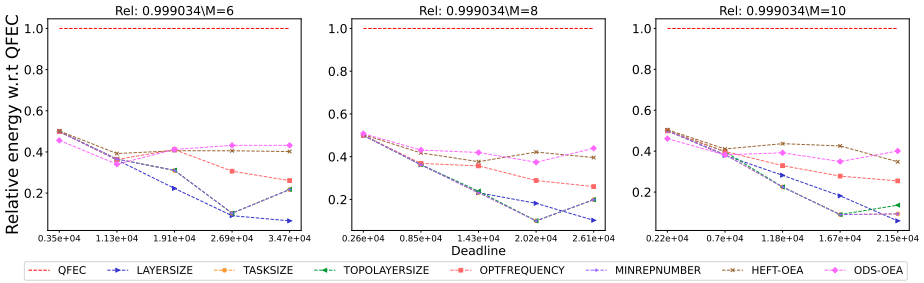

Figure 2431: Assessing the performance of different processor number on the Cholesky workflow.

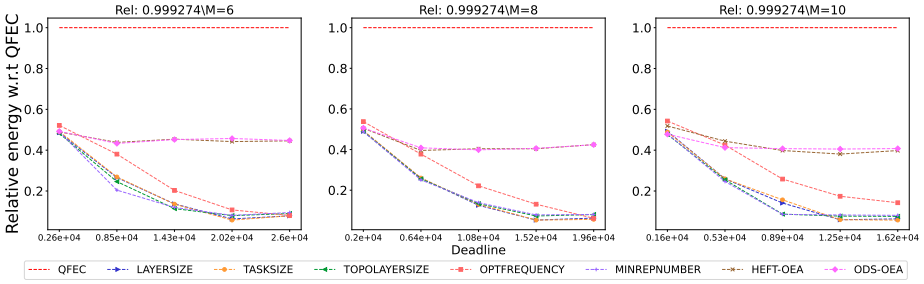

Figure 2432: Assessing the performance of different processor number on the Cycles workflow.

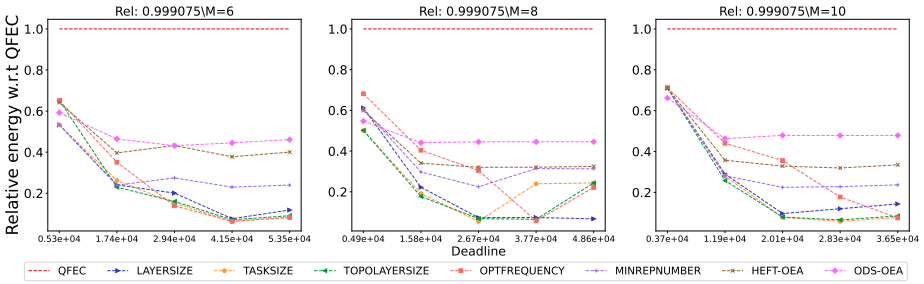

Figure 2433: Assessing the performance of different processor number on the Epigenomics workflow.

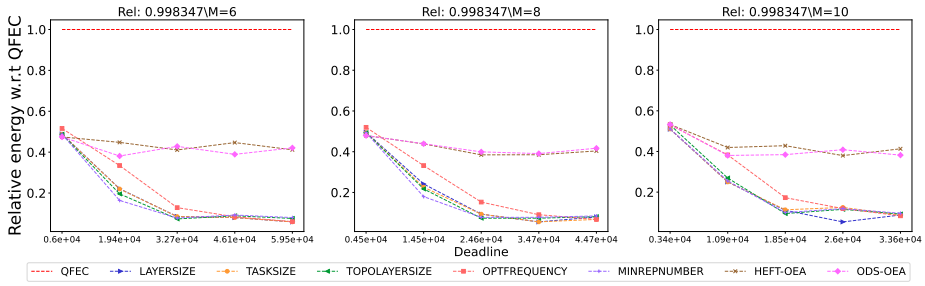

Figure 2434: Assessing the performance of different processor number on the Genome workflow.

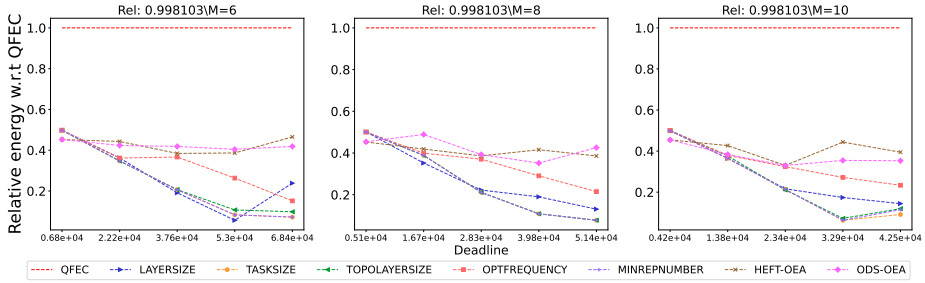

Figure 2435: Assessing the performance of different processor number on the LU.

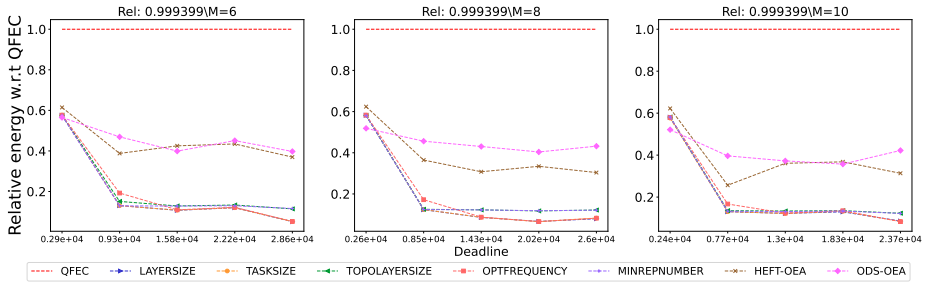

Figure 2436: Assessing the performance of different processor number on the Montage workflow.

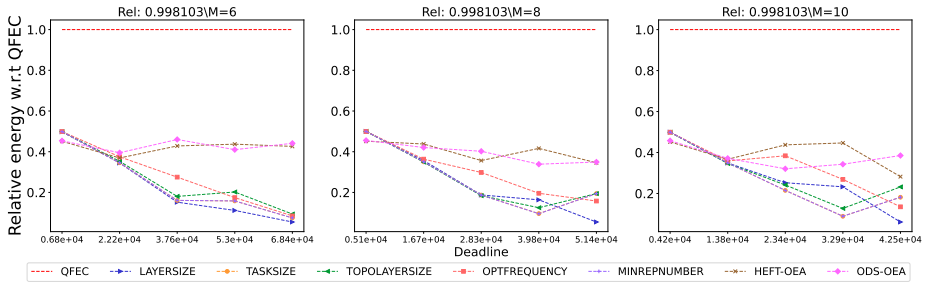

Figure 2437: Assessing the performance of different processor number on the QR.

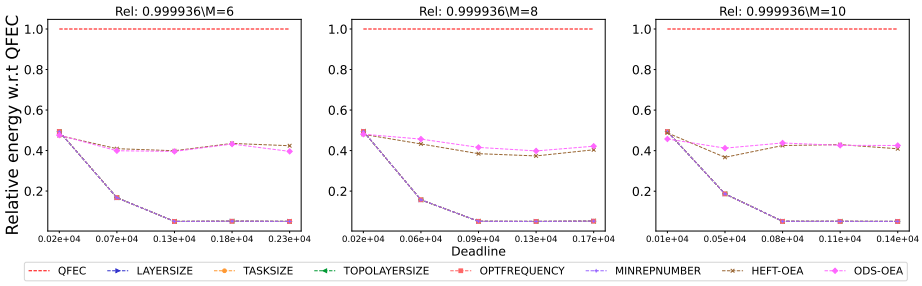

Figure 2438: Assessing the performance of different processor number on the Seismology workflow.

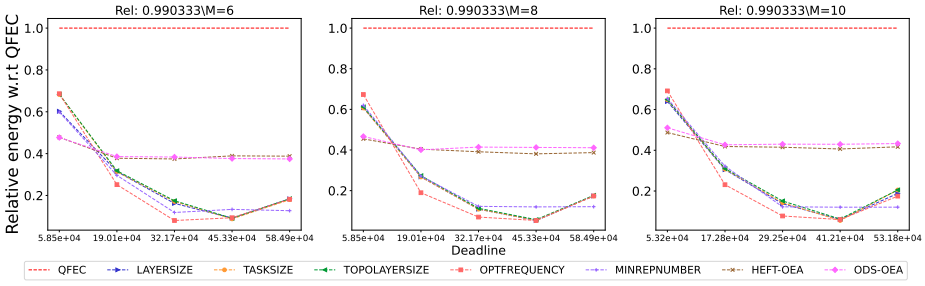

Figure 2439: Assessing the performance of different processor number on the SoyKB workflow.

H.5.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

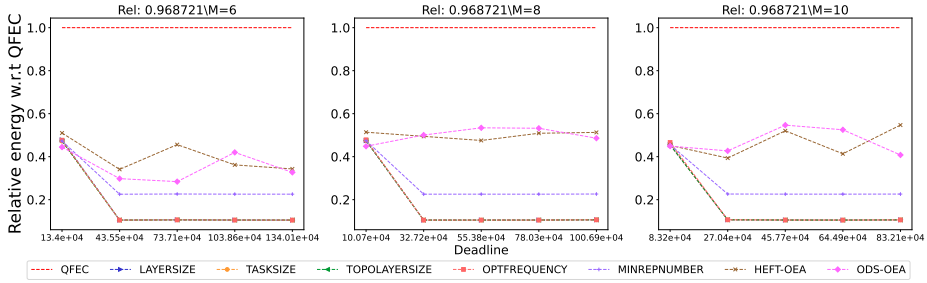

Figure 2440: Assessing the performance of different processor number on the BLAST workflow.

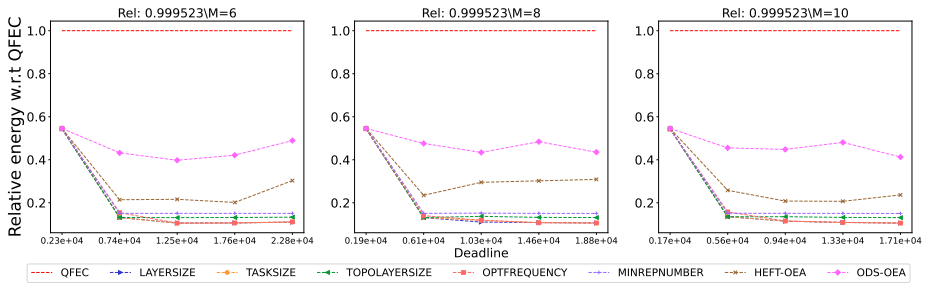

Figure 2441: Assessing the performance of different processor number on the BWA workflow.

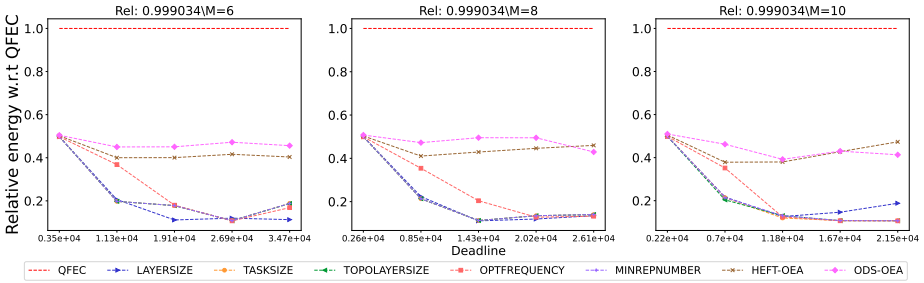

Figure 2442: Assessing the performance of different processor number on the Cholesky workflow.

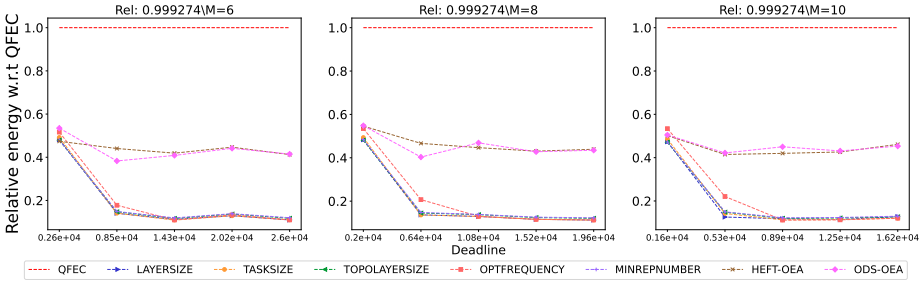

Figure 2443: Assessing the performance of different processor number on the Cycles workflow.

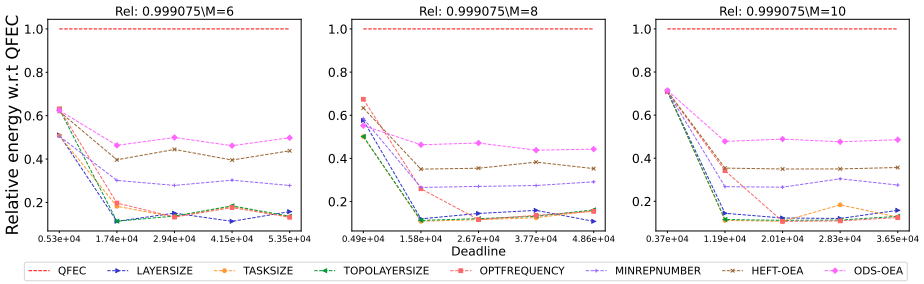

Figure 2444: Assessing the performance of different processor number on the Epigenomics workflow.

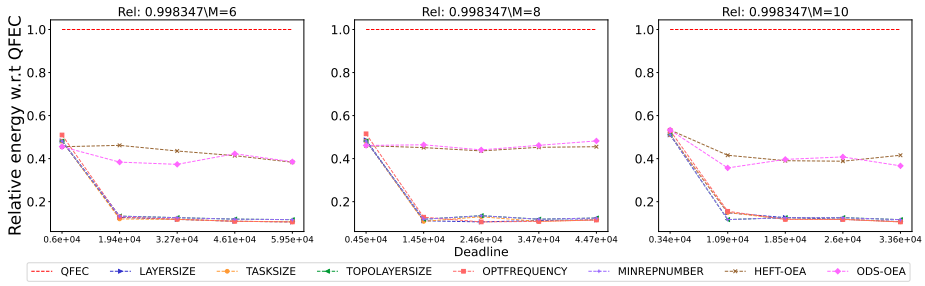

Figure 2445: Assessing the performance of different processor number on the Genome workflow.

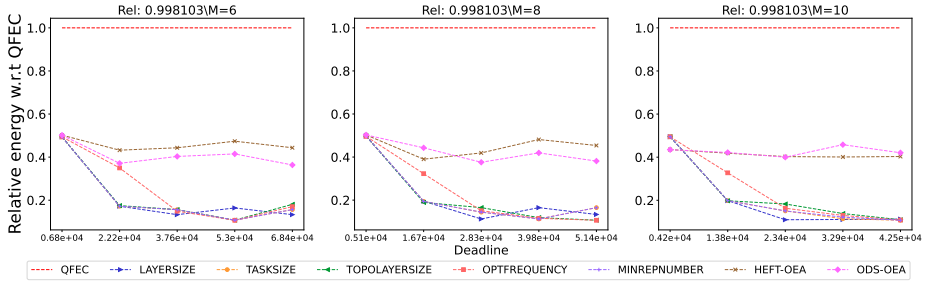

Figure 2446: Assessing the performance of different processor number on the LU.

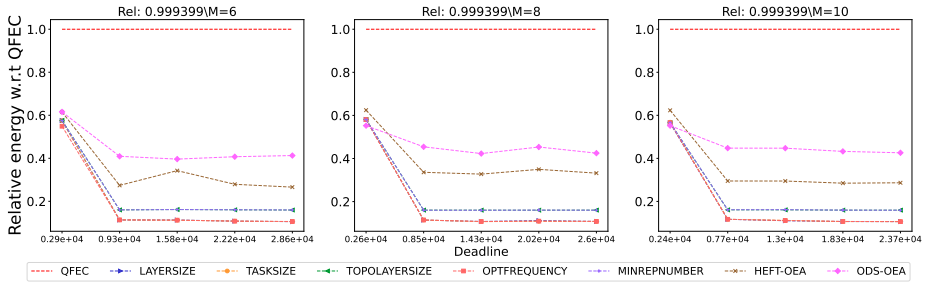

Figure 2447: Assessing the performance of different processor number on the Montage workflow.

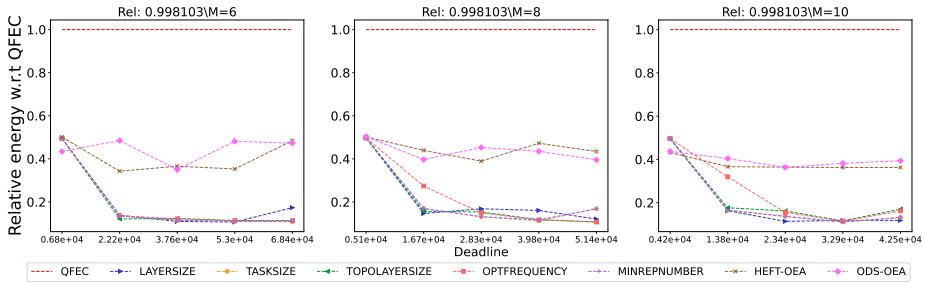

Figure 2448: Assessing the performance of different processor number on the QR.

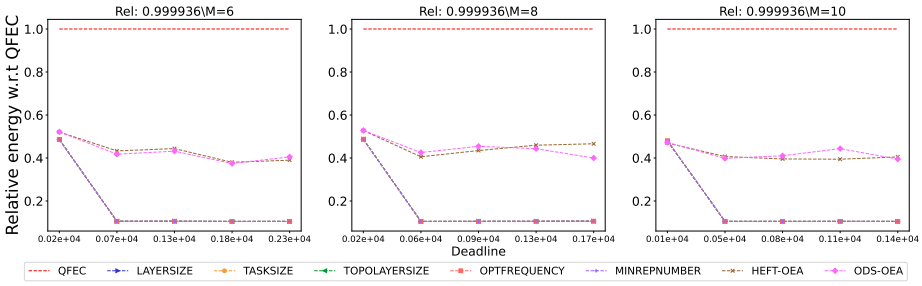

Figure 2449: Assessing the performance of different processor number on the Seismology workflow.

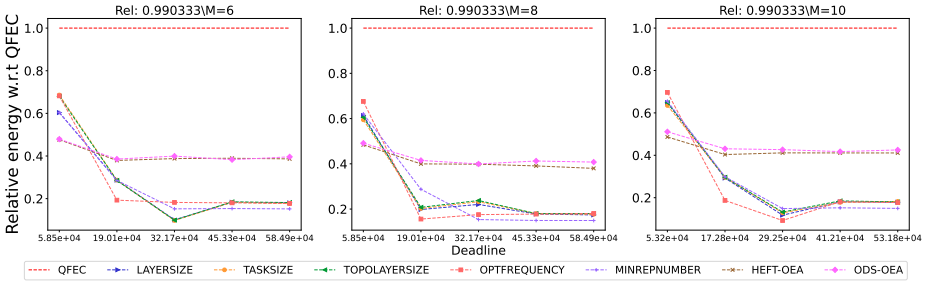

Figure 2450: Assessing the performance of different processor number on the SoyKB workflow.

H.5.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

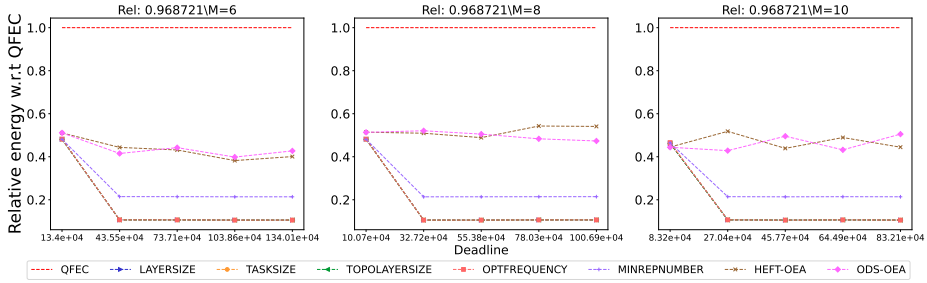

Figure 2451: Assessing the performance of different processor number on the BLAST workflow.

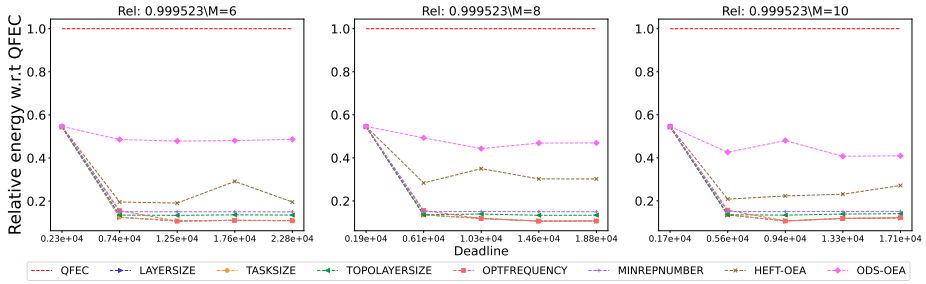

Figure 2452: Assessing the performance of different processor number on the BWA workflow.

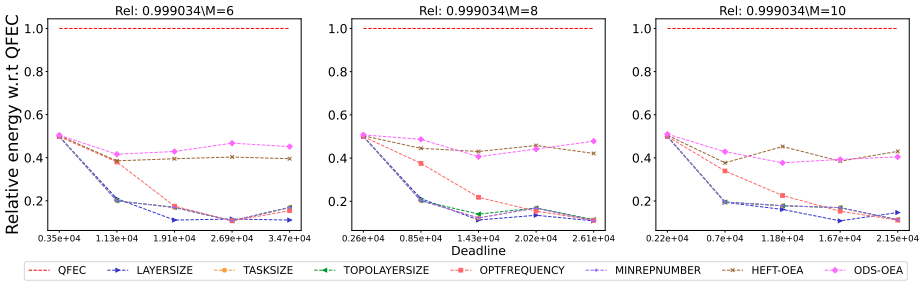

Figure 2453: Assessing the performance of different processor number on the Cholesky workflow.

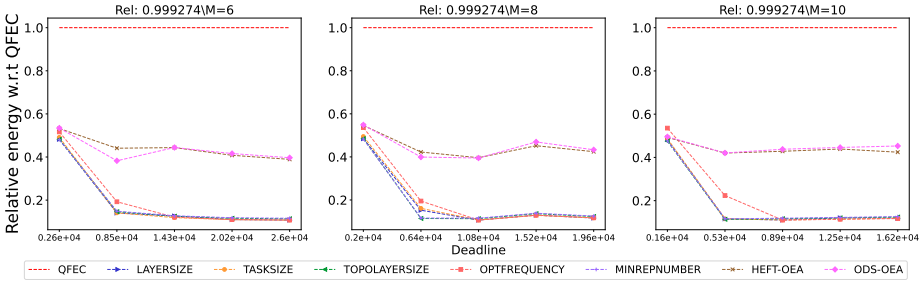

Figure 2454: Assessing the performance of different processor number on the Cycles workflow.

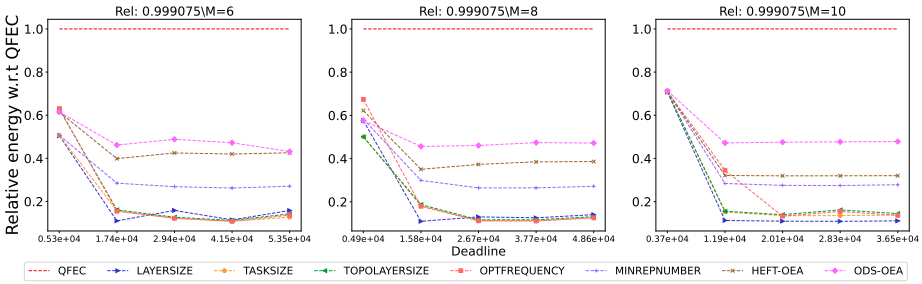

Figure 2455: Assessing the performance of different processor number on the Epigenomics workflow.

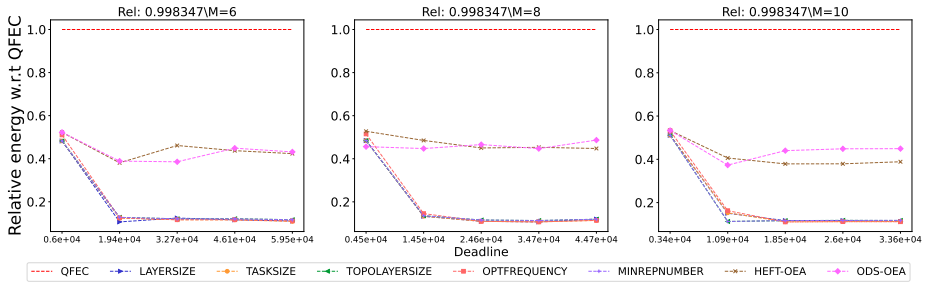

Figure 2456: Assessing the performance of different processor number on the Genome workflow.

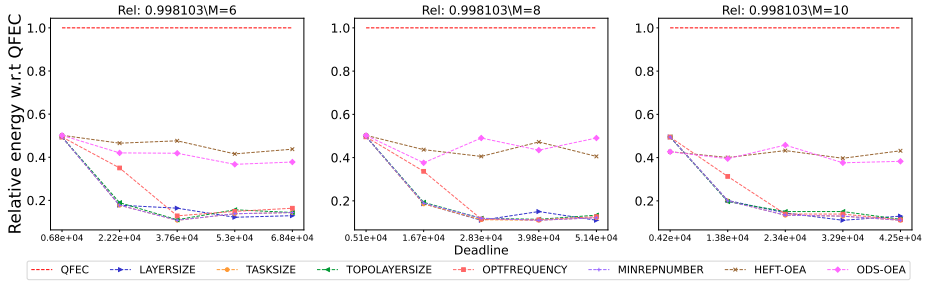

Figure 2457: Assessing the performance of different processor number on the LU.

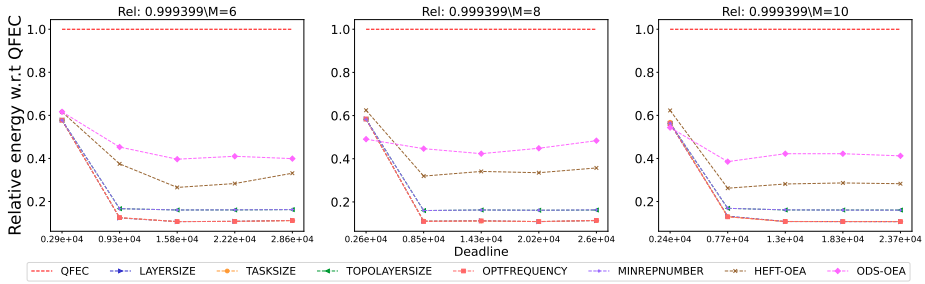

Figure 2458: Assessing the performance of different processor number on the Montage workflow.

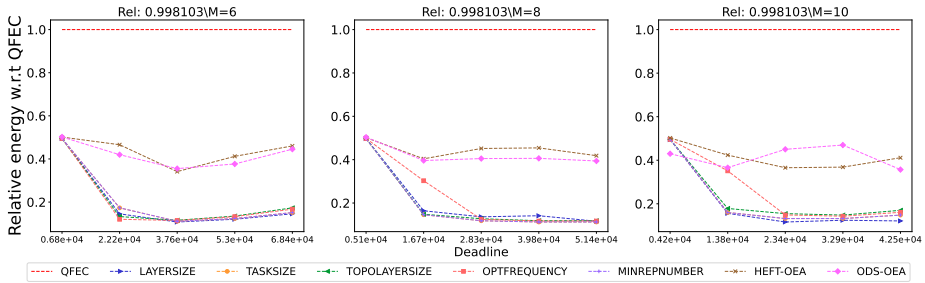

Figure 2459: Assessing the performance of different processor number on the QR.

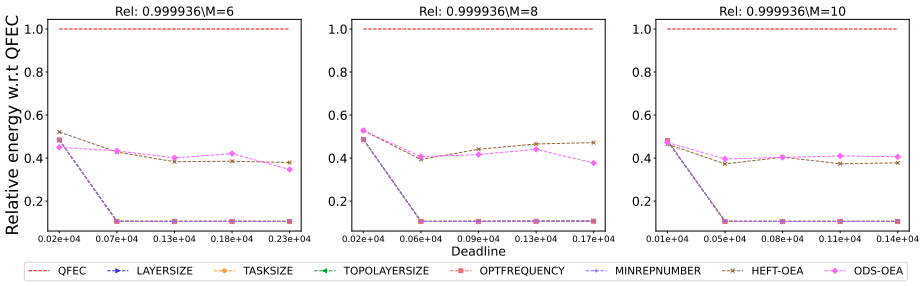

Figure 2460: Assessing the performance of different processor number on the Seismology workflow.

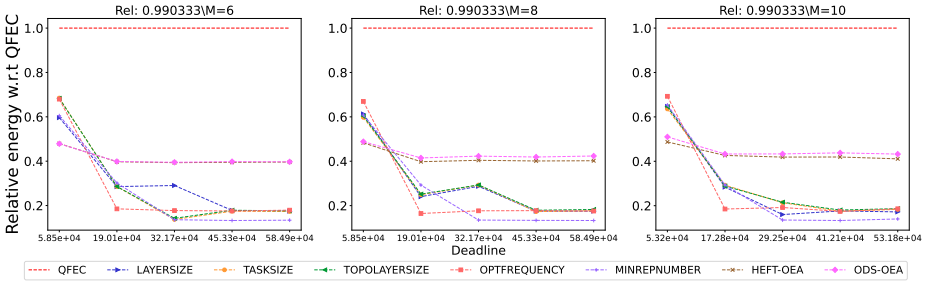

Figure 2461: Assessing the performance of different processor number on the SoyKB workflow.

H.6 $BC/WC = 0.6$

H.6.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

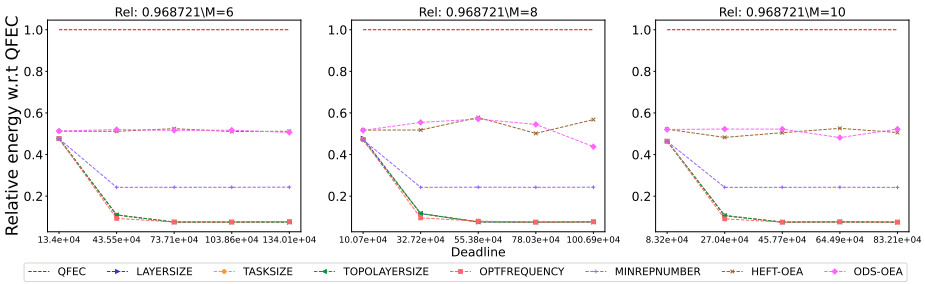

Figure 2462: Assessing the performance of different processor number on the BLAST workflow.

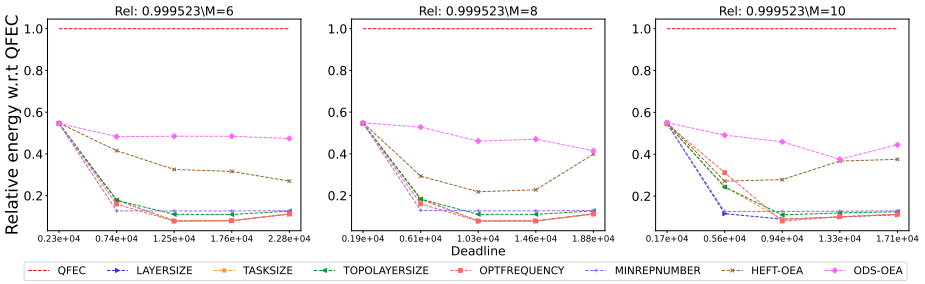

Figure 2463: Assessing the performance of different processor number on the BWA workflow.

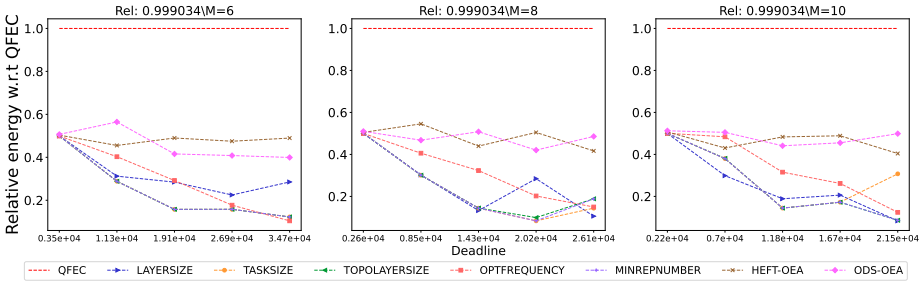

Figure 2464: Assessing the performance of different processor number on the Cholesky workflow.

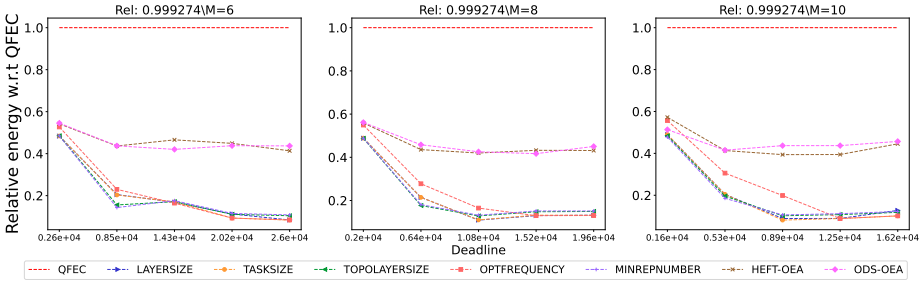

Figure 2465: Assessing the performance of different processor number on the Cycles workflow.

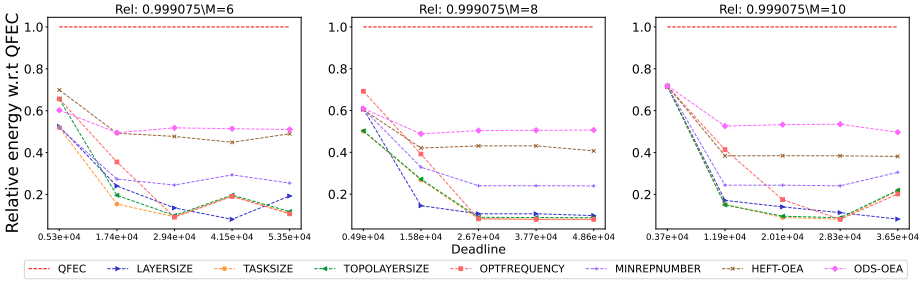

Figure 2466: Assessing the performance of different processor number on the Epigenomics workflow.

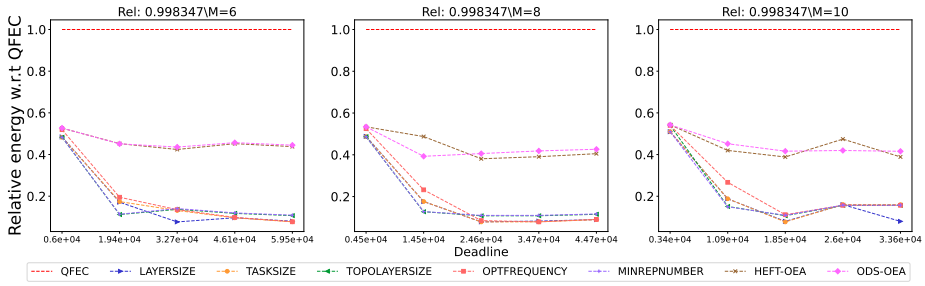

Figure 2467: Assessing the performance of different processor number on the Genome workflow.

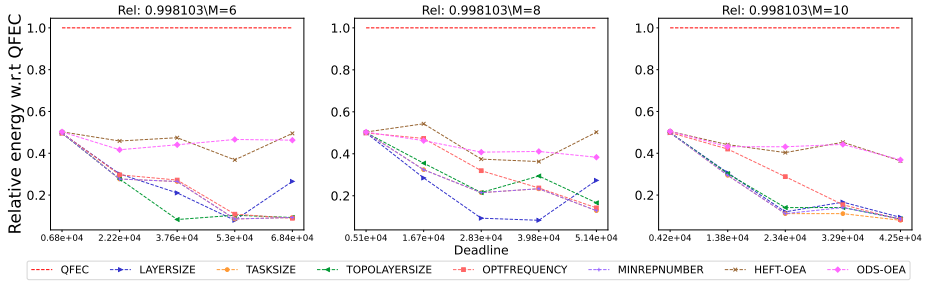

Figure 2468: Assessing the performance of different processor number on the LU.

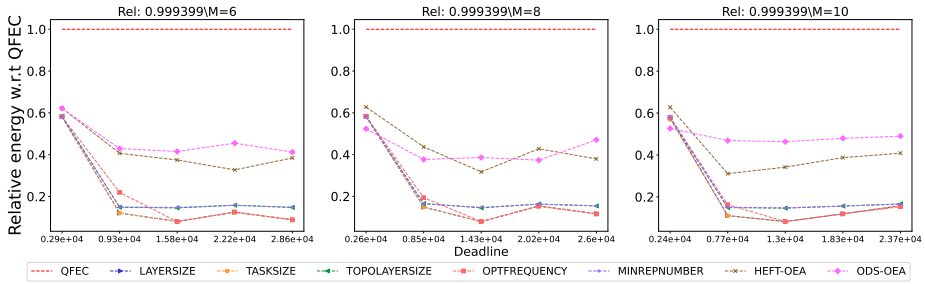

Figure 2469: Assessing the performance of different processor number on the Montage workflow.

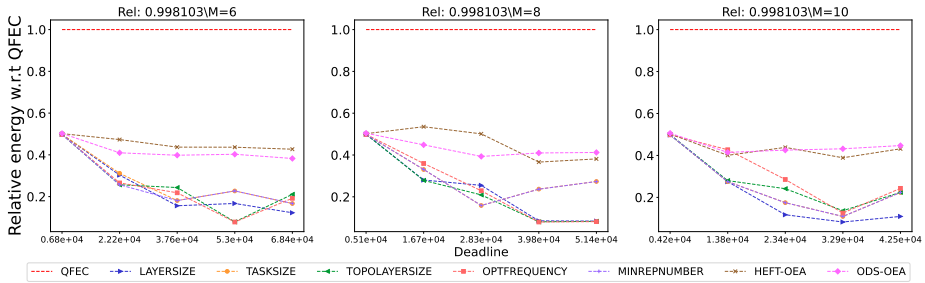

Figure 2470: Assessing the performance of different processor number on the QR.

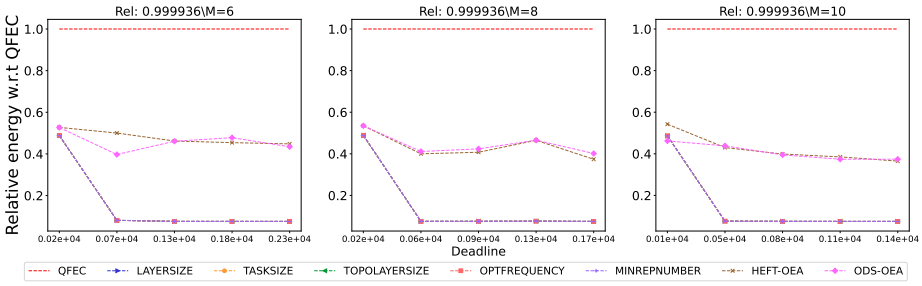

Figure 2471: Assessing the performance of different processor number on the Seismology workflow.

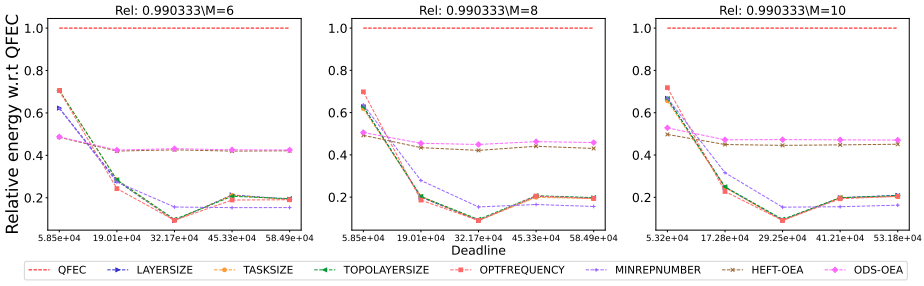

Figure 2472: Assessing the performance of different processor number on the SoyKB workflow.

H.6.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

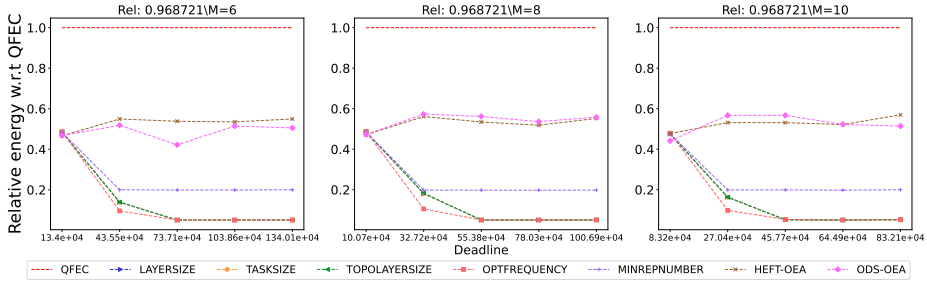

Figure 2473: Assessing the performance of different processor number on the BLAST workflow.

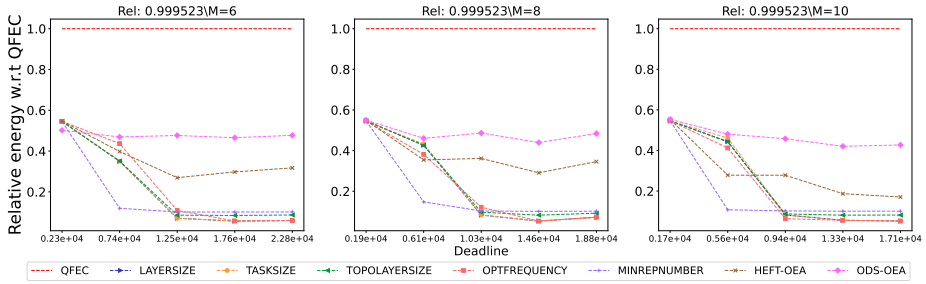

Figure 2474: Assessing the performance of different processor number on the BWA workflow.

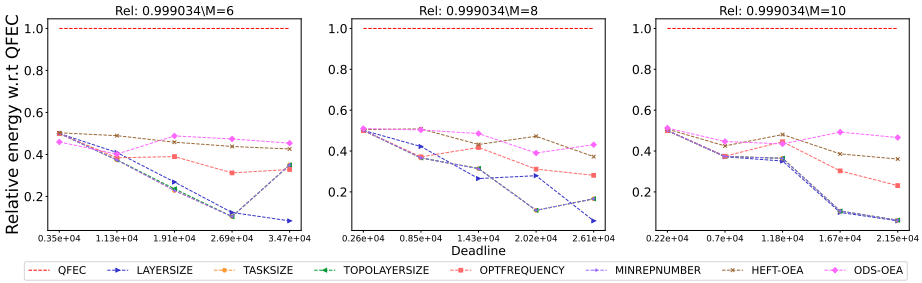

Figure 2475: Assessing the performance of different processor number on the Cholesky workflow.

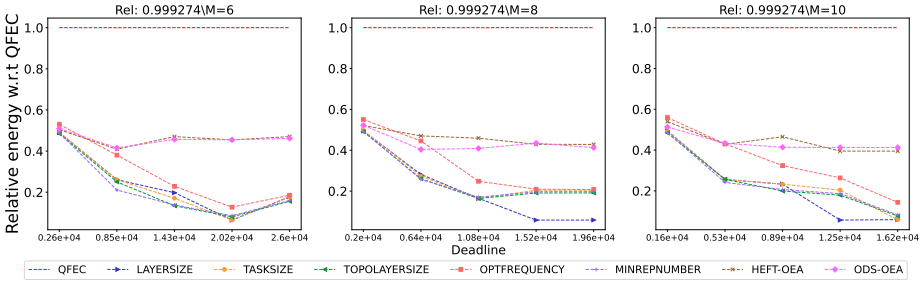

Figure 2476: Assessing the performance of different processor number on the Cycles workflow.

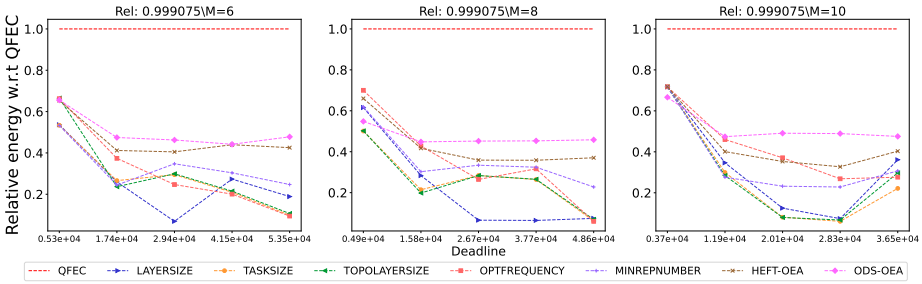

Figure 2477: Assessing the performance of different processor number on the Epigenomics workflow.

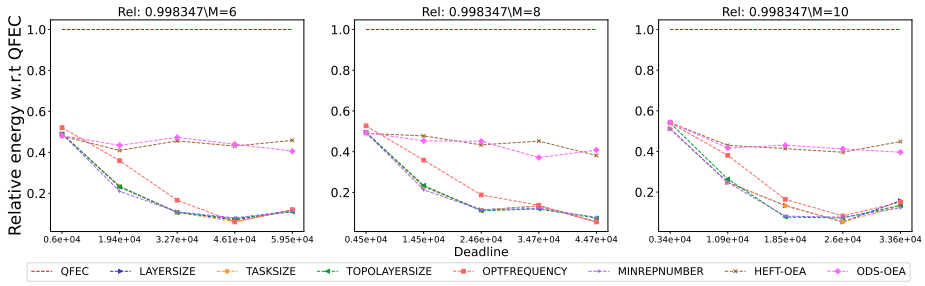

Figure 2478: Assessing the performance of different processor number on the Genome workflow.

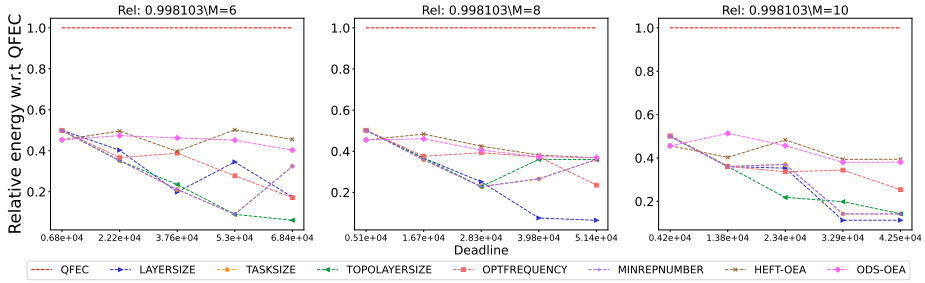

Figure 2479: Assessing the performance of different processor number on the LU.

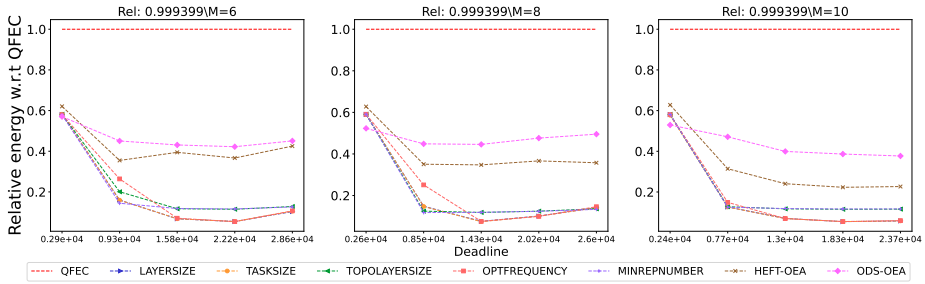

Figure 2480: Assessing the performance of different processor number on the Montage workflow.

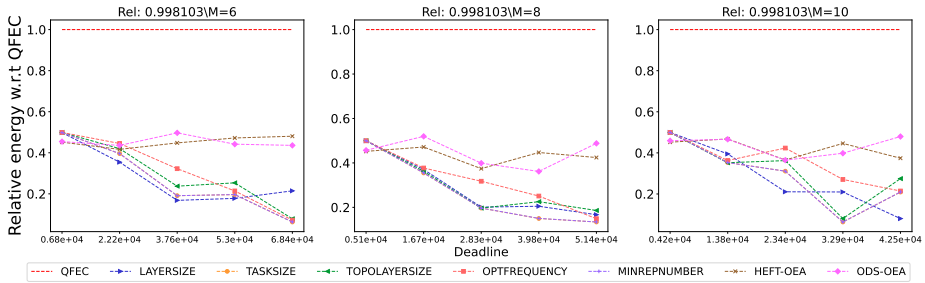

Figure 2481: Assessing the performance of different processor number on the QR.

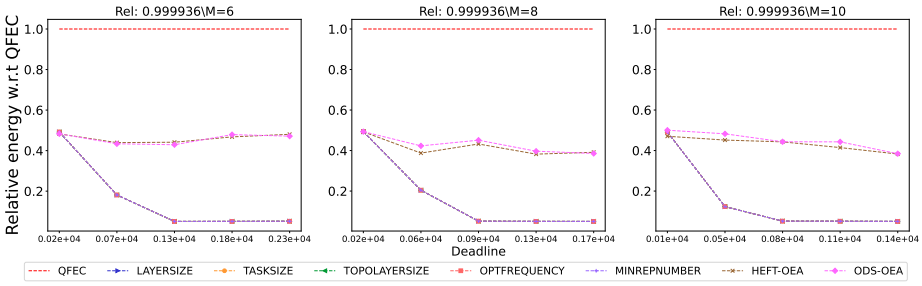

Figure 2482: Assessing the performance of different processor number on the Seismology workflow.

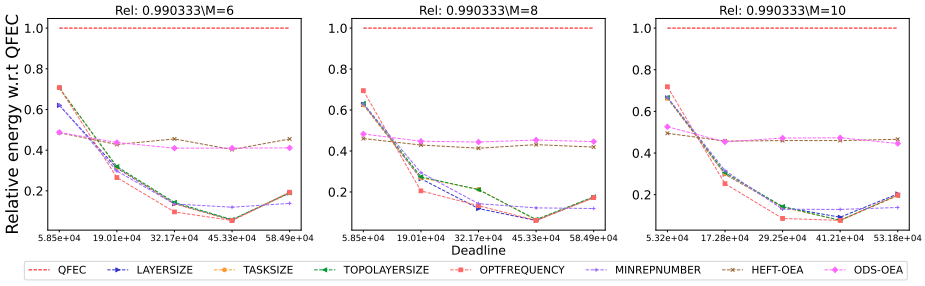

Figure 2483: Assessing the performance of different processor number on the SoyKB workflow.

H.6.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

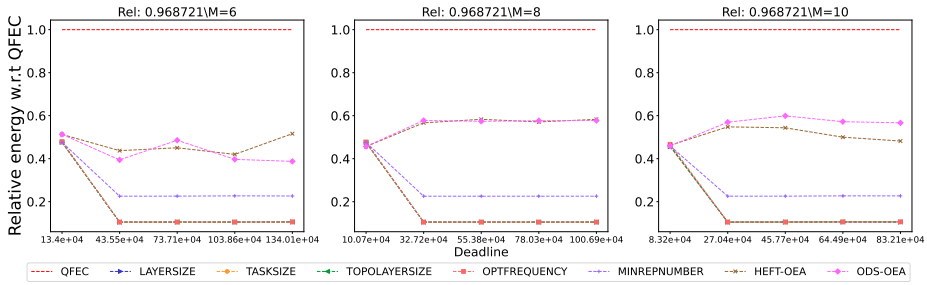

Figure 2484: Assessing the performance of different processor number on the BLAST workflow.

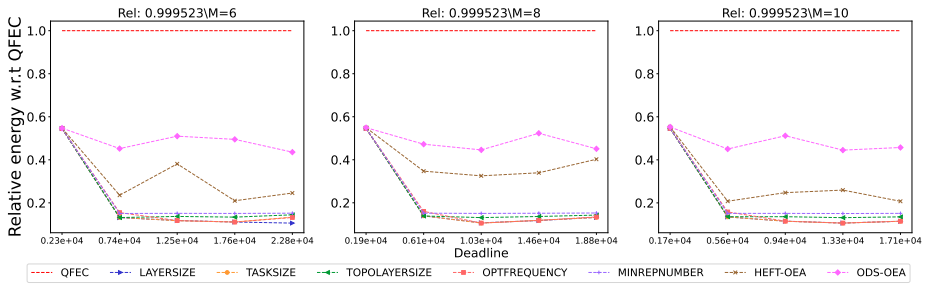

Figure 2485: Assessing the performance of different processor number on the BWA workflow.

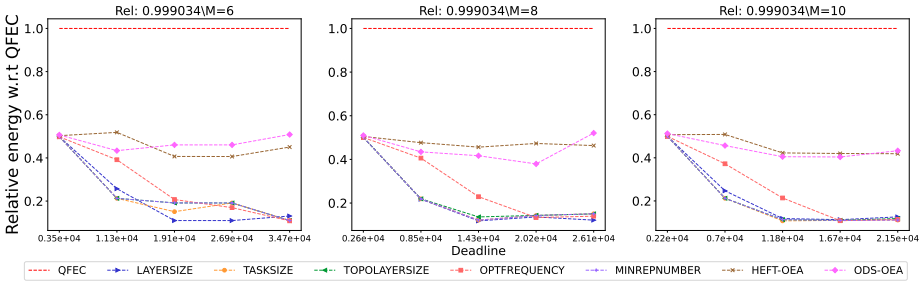

Figure 2486: Assessing the performance of different processor number on the Cholesky workflow.

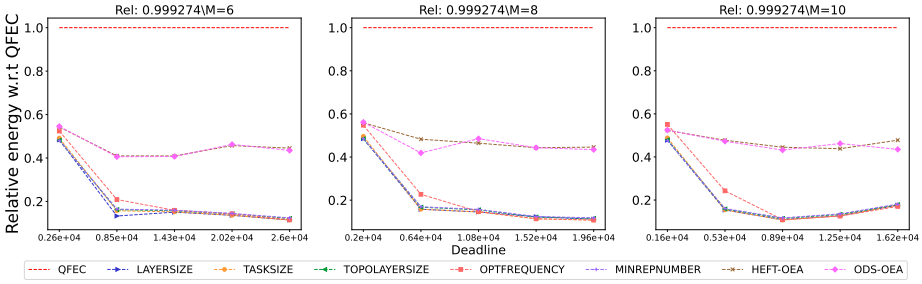

Figure 2487: Assessing the performance of different processor number on the Cycles workflow.

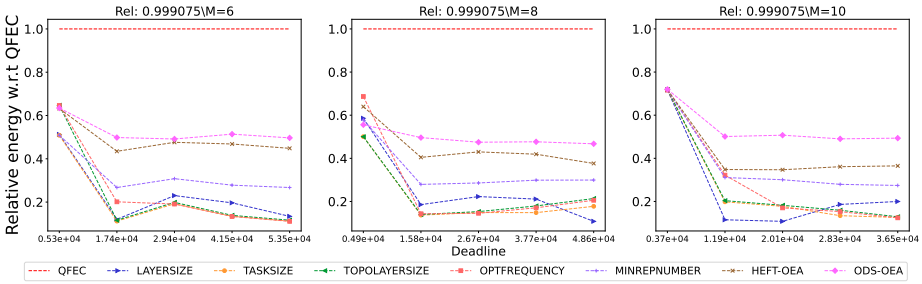

Figure 2488: Assessing the performance of different processor number on the Epigenomics workflow.

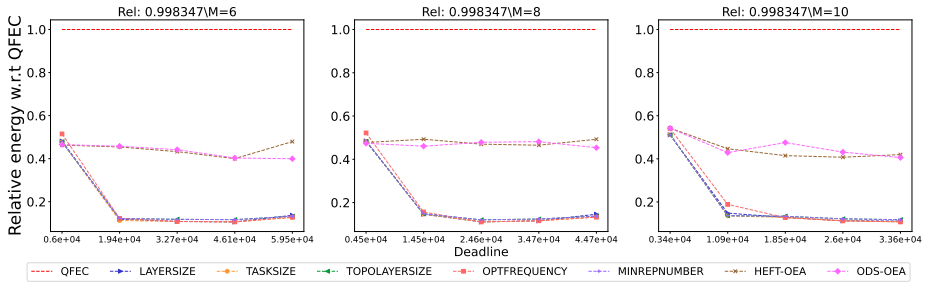

Figure 2489: Assessing the performance of different processor number on the Genome workflow.

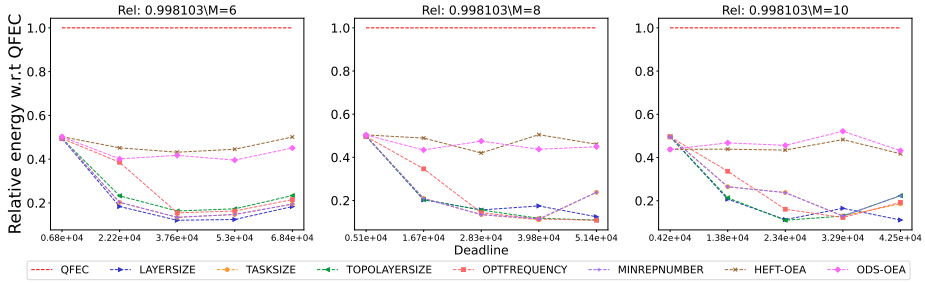

Figure 2490: Assessing the performance of different processor number on the LU.

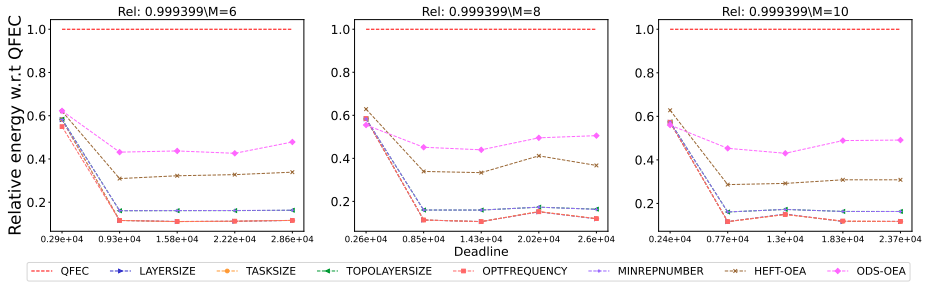

Figure 2491: Assessing the performance of different processor number on the Montage workflow.

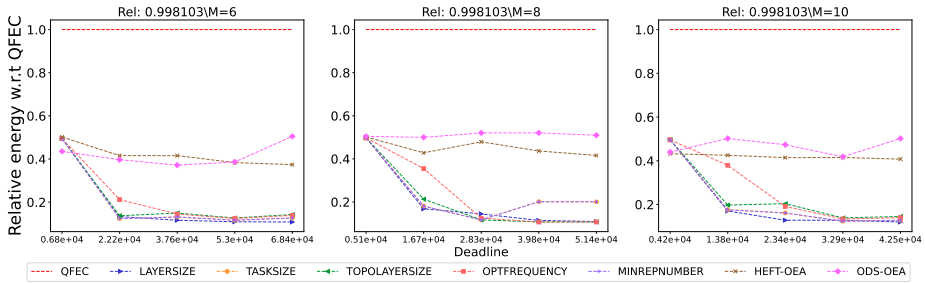

Figure 2492: Assessing the performance of different processor number on the QR.

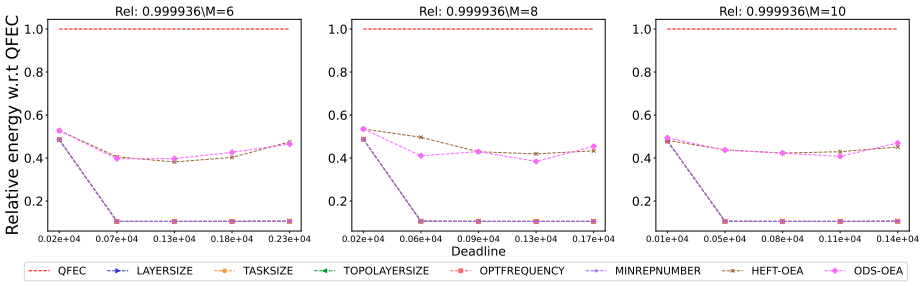

Figure 2493: Assessing the performance of different processor number on the Seismology workflow.

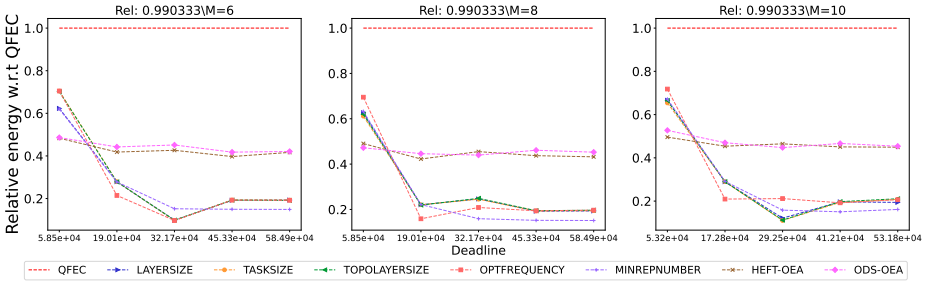

Figure 2494: Assessing the performance of different processor number on the SoyKB workflow.

H.6.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

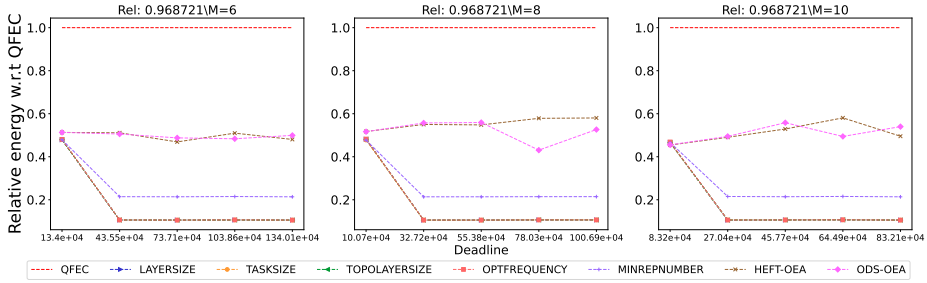

Figure 2495: Assessing the performance of different processor number on the BLAST workflow.

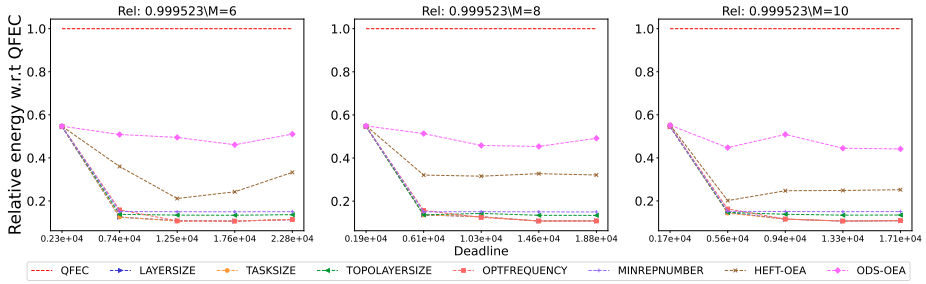

Figure 2496: Assessing the performance of different processor number on the BWA workflow.

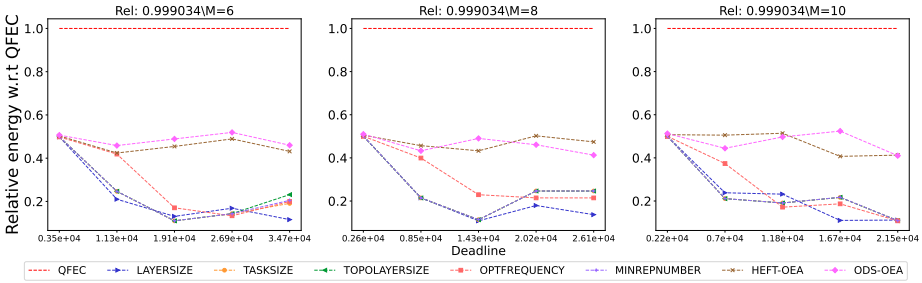

Figure 2497: Assessing the performance of different processor number on the Cholesky workflow.

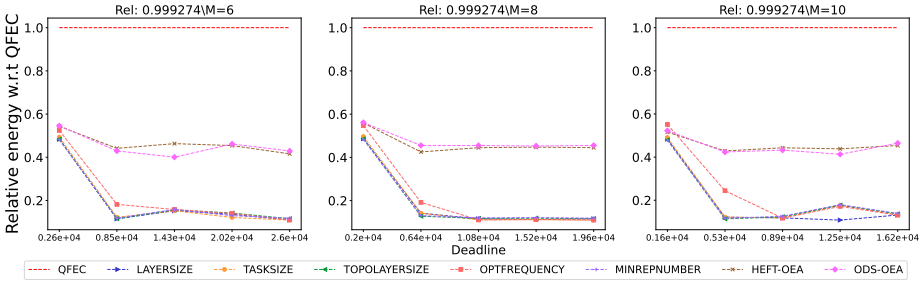

Figure 2498: Assessing the performance of different processor number on the Cycles workflow.

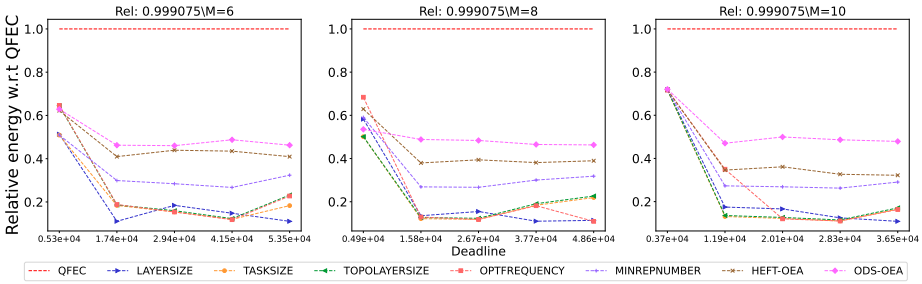

Figure 2499: Assessing the performance of different processor number on the Epigenomics workflow.

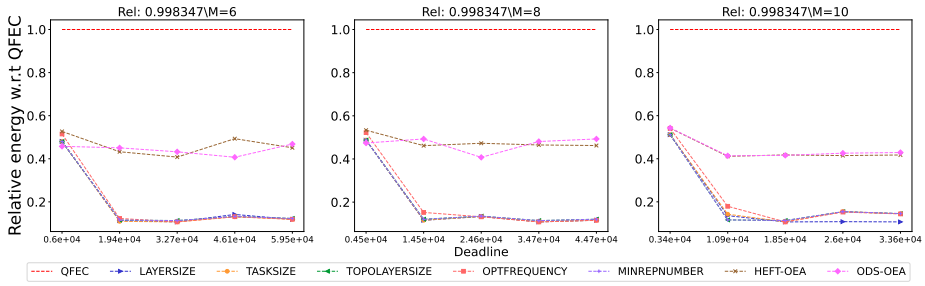

Figure 2500: Assessing the performance of different processor number on the Genome workflow.

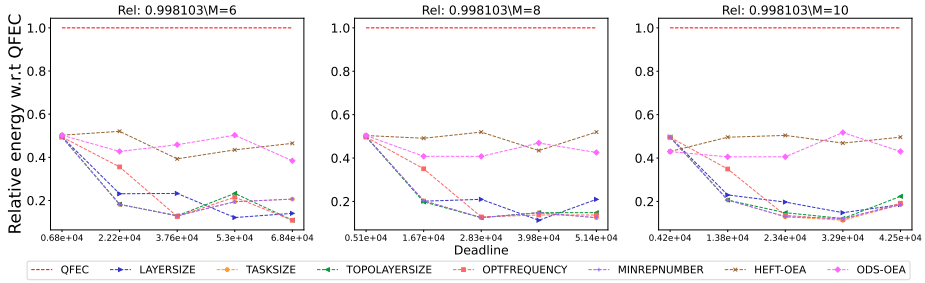

Figure 2501: Assessing the performance of different processor number on the LU.

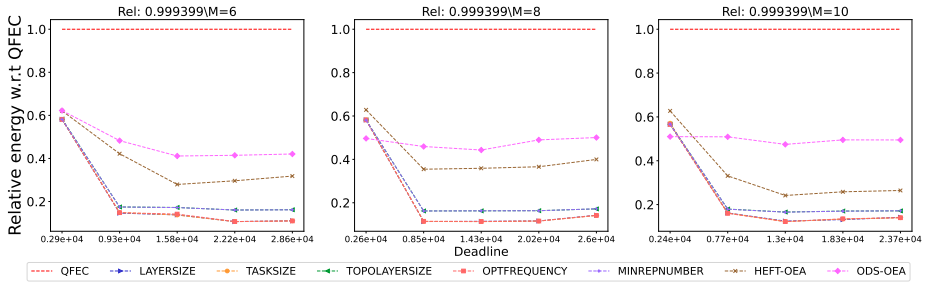

Figure 2502: Assessing the performance of different processor number on the Montage workflow.

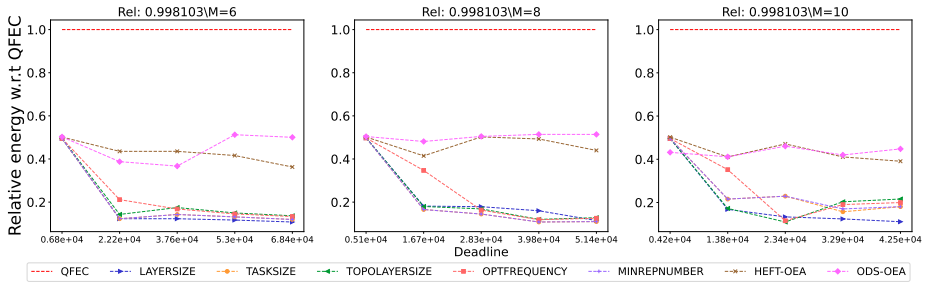

Figure 2503: Assessing the performance of different processor number on the QR.

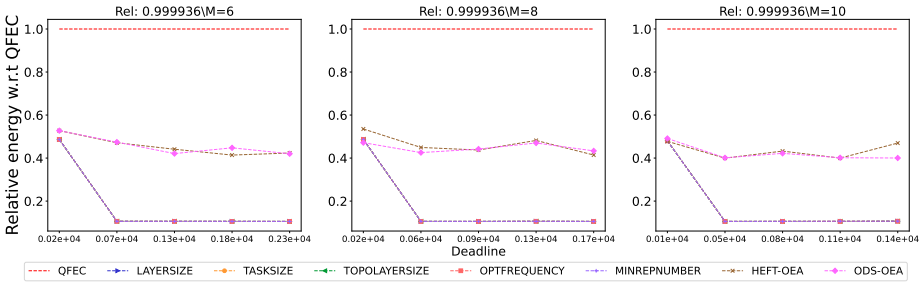

Figure 2504: Assessing the performance of different processor number on the Seismology workflow.

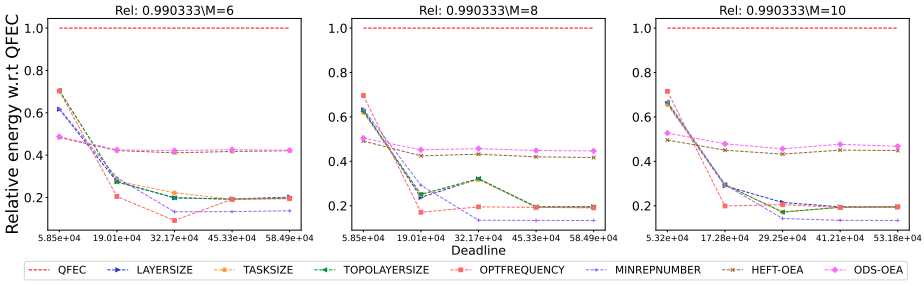

Figure 2505: Assessing the performance of different processor number on the SoyKB workflow.

H.7 $BC/WC = 0.7$

H.7.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

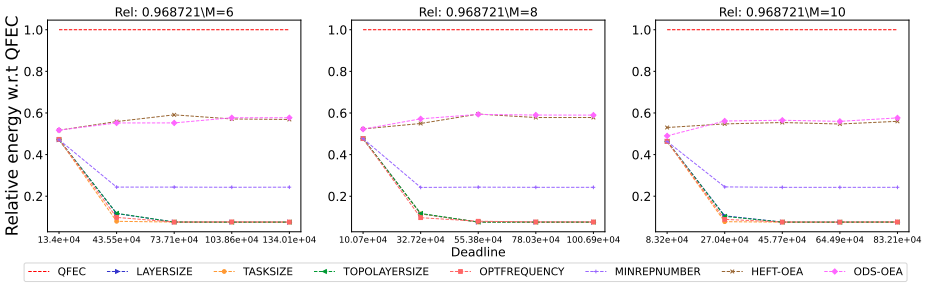

Figure 2506: Assessing the performance of different processor number on the BLAST workflow.

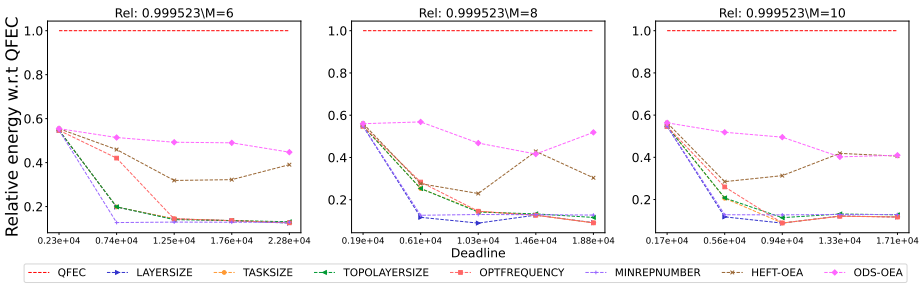

Figure 2507: Assessing the performance of different processor number on the BWA workflow.

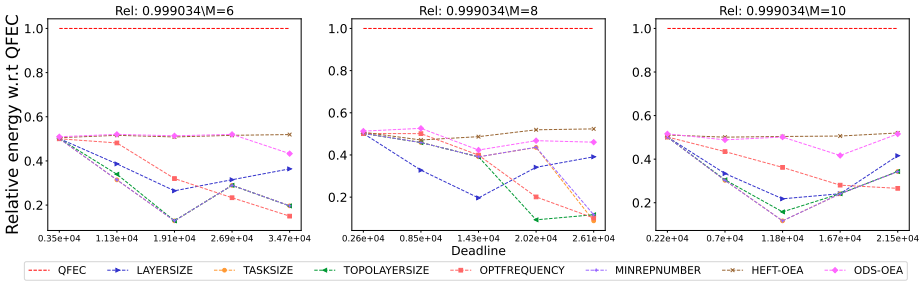

Figure 2508: Assessing the performance of different processor number on the Cholesky workflow.

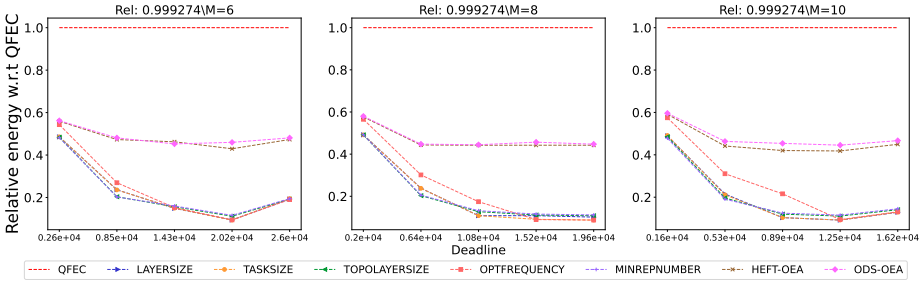

Figure 2509: Assessing the performance of different processor number on the Cycles workflow.

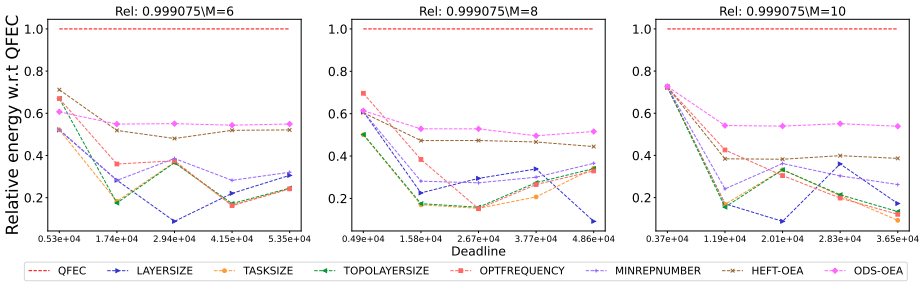

Figure 2510: Assessing the performance of different processor number on the Epigenomics workflow.

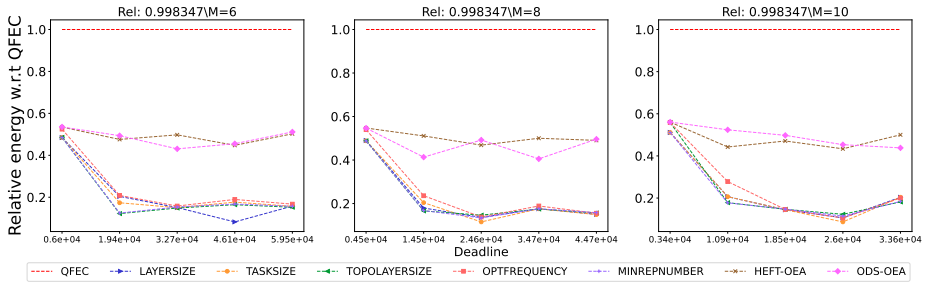

Figure 2511: Assessing the performance of different processor number on the Genome workflow.

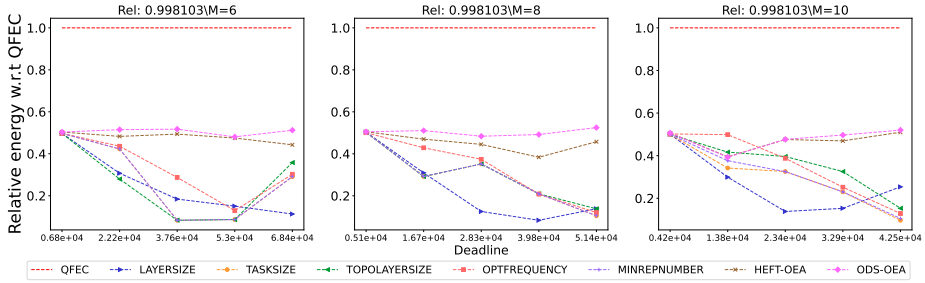

Figure 2512: Assessing the performance of different processor number on the LU.

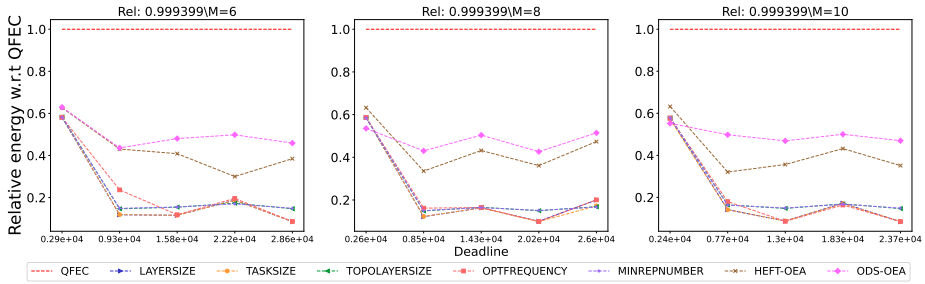

Figure 2513: Assessing the performance of different processor number on the Montage workflow.

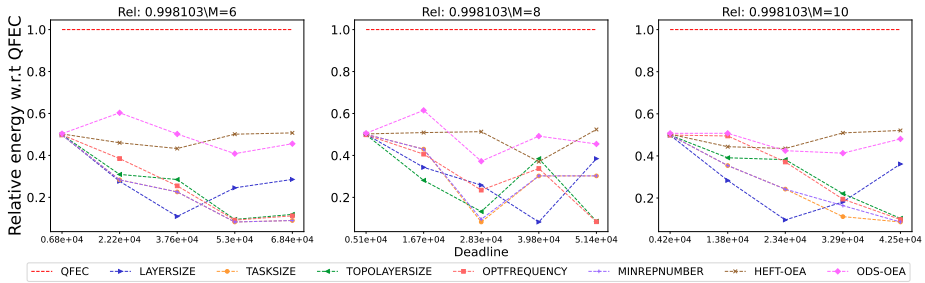

Figure 2514: Assessing the performance of different processor number on the QR.

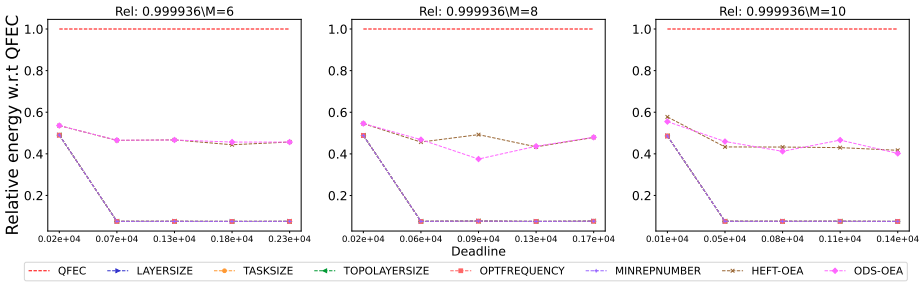

Figure 2515: Assessing the performance of different processor number on the Seismology workflow.

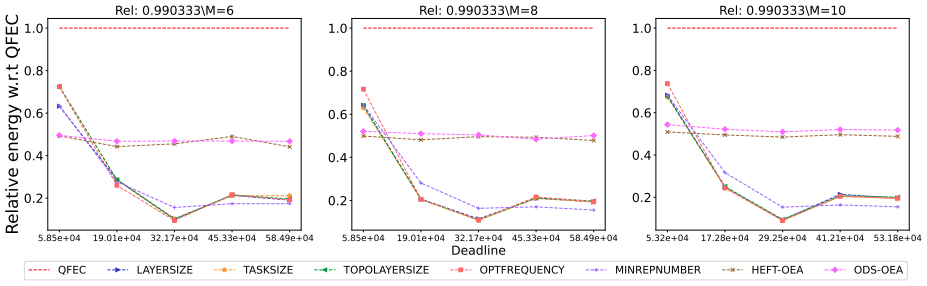

Figure 2516: Assessing the performance of different processor number on the SoyKB workflow.

H.7.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

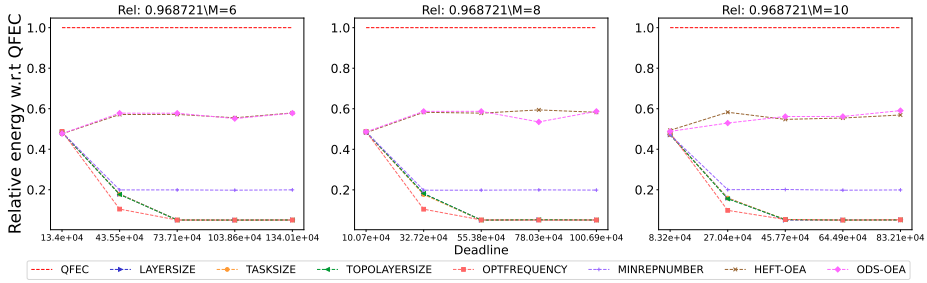

Figure 2517: Assessing the performance of different processor number on the BLAST workflow.

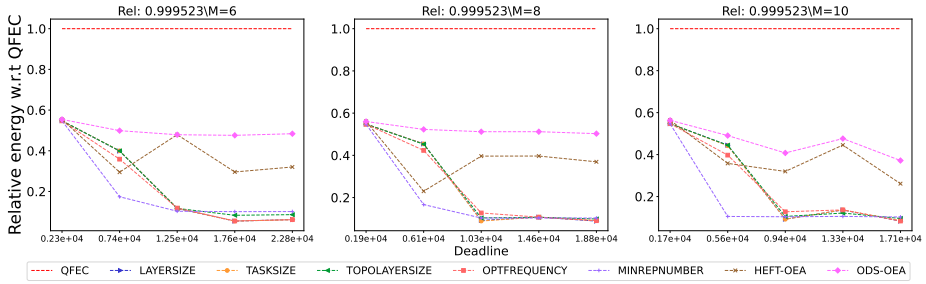

Figure 2518: Assessing the performance of different processor number on the BWA workflow.

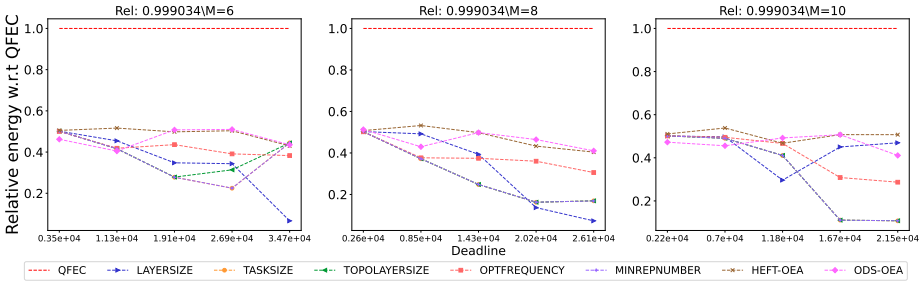

Figure 2519: Assessing the performance of different processor number on the Cholesky workflow.

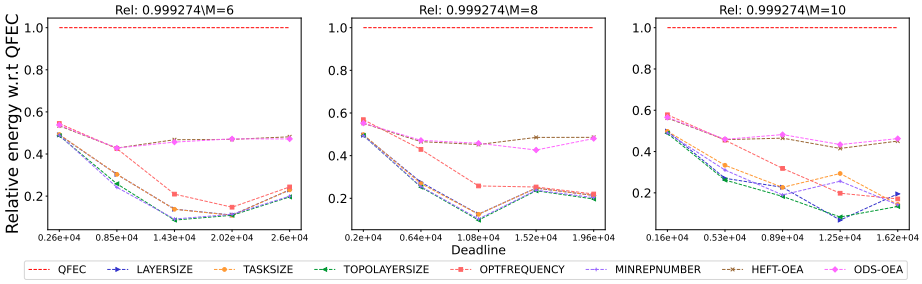

Figure 2520: Assessing the performance of different processor number on the Cycles workflow.

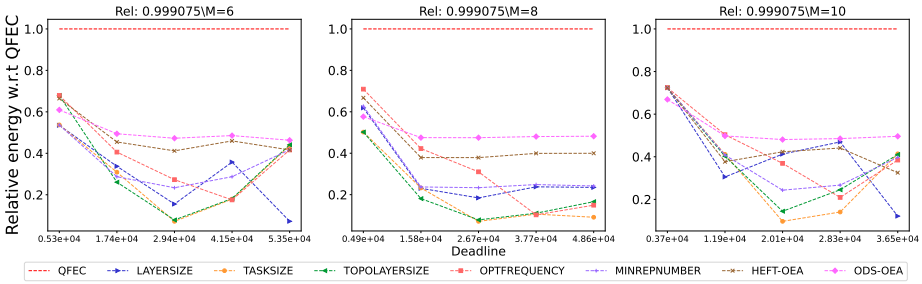

Figure 2521: Assessing the performance of different processor number on the Epigenomics workflow.

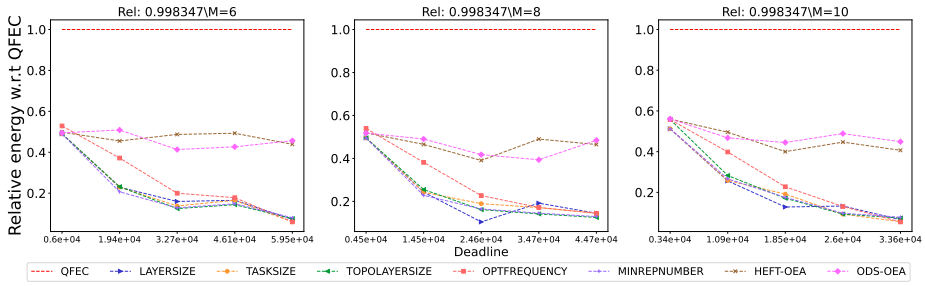

Figure 2522: Assessing the performance of different processor number on the Genome workflow.

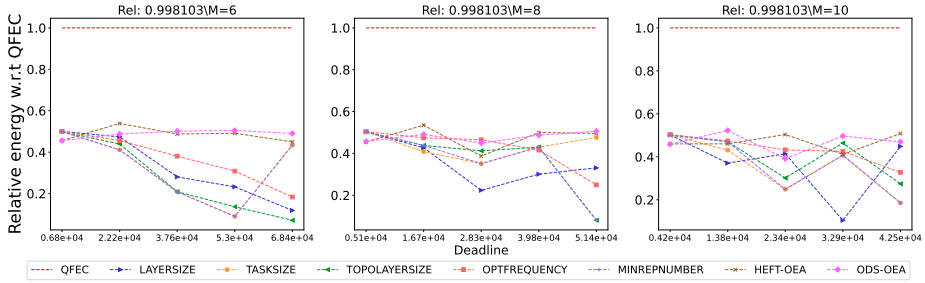

Figure 2523: Assessing the performance of different processor number on the LU.

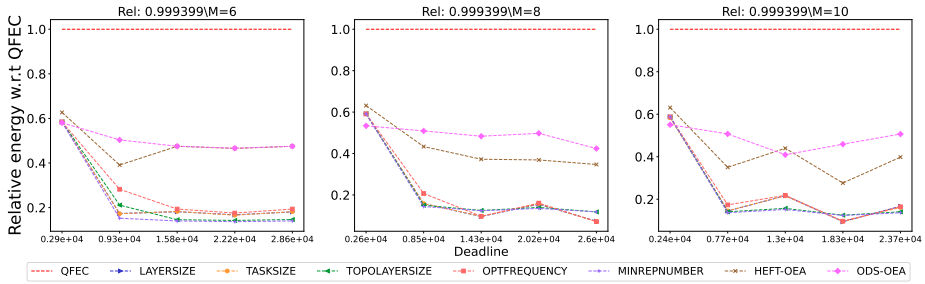

Figure 2524: Assessing the performance of different processor number on the Montage workflow.

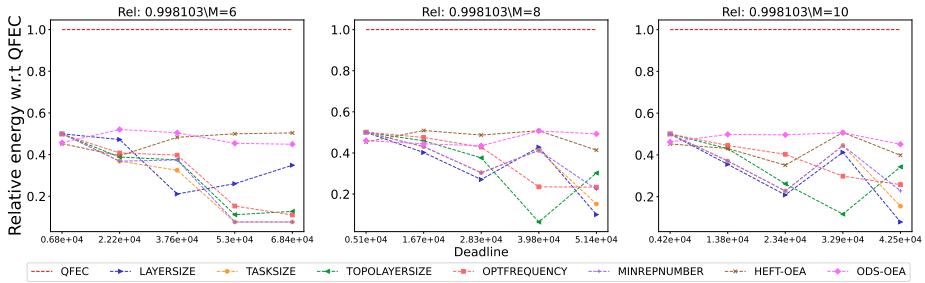

Figure 2525: Assessing the performance of different processor number on the QR.

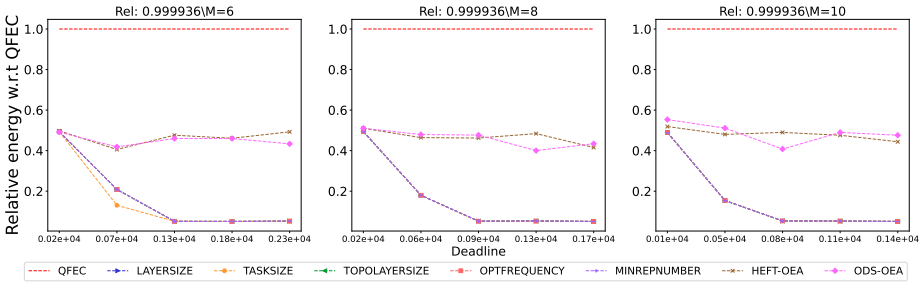

Figure 2526: Assessing the performance of different processor number on the Seismology workflow.

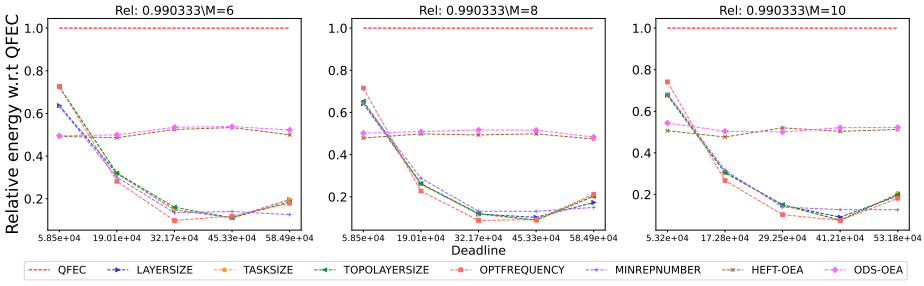

Figure 2527: Assessing the performance of different processor number on the SoyKB workflow.

H.7.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

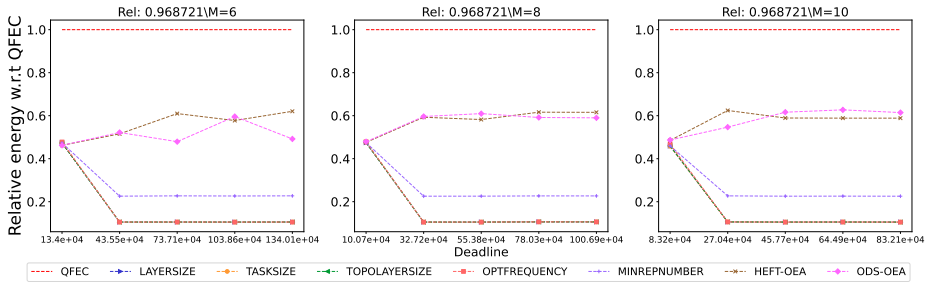

Figure 2528: Assessing the performance of different processor number on the BLAST workflow.

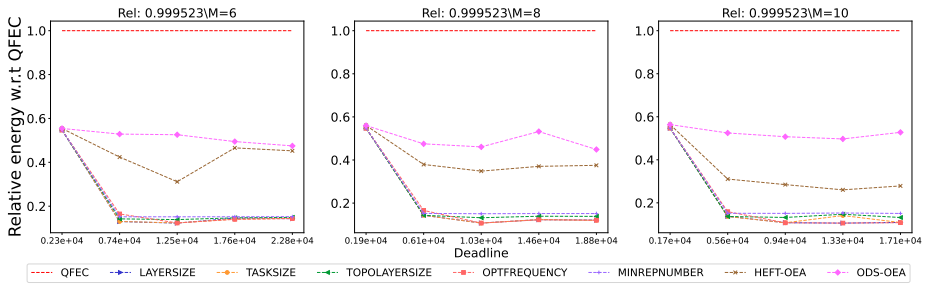

Figure 2529: Assessing the performance of different processor number on the BWA workflow.

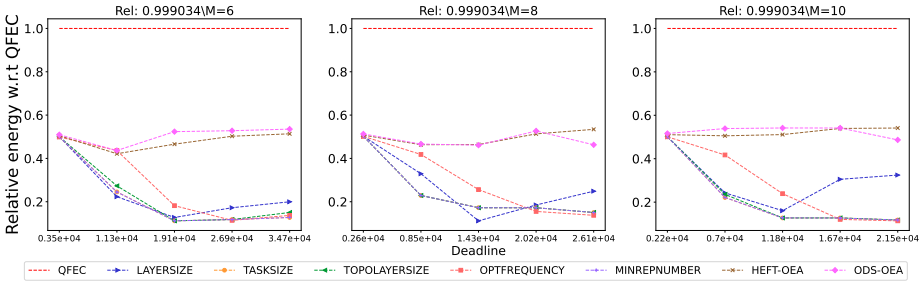

Figure 2530: Assessing the performance of different processor number on the Cholesky workflow.

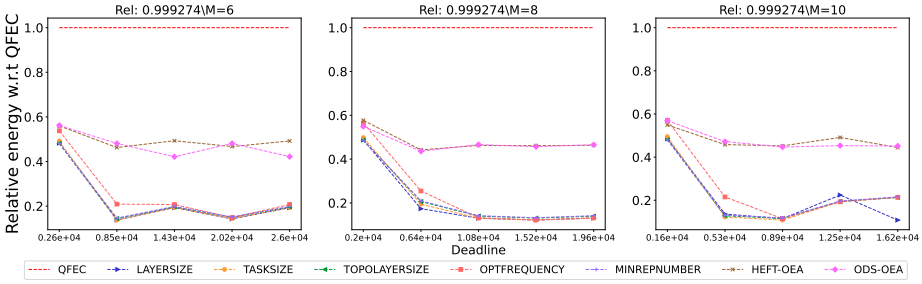

Figure 2531: Assessing the performance of different processor number on the Cycles workflow.

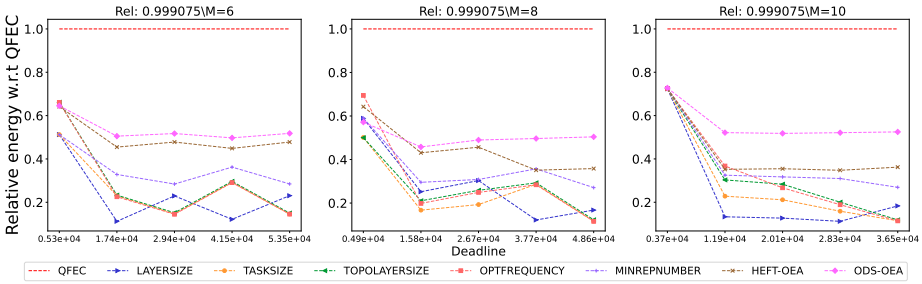

Figure 2532: Assessing the performance of different processor number on the Epigenomics workflow.

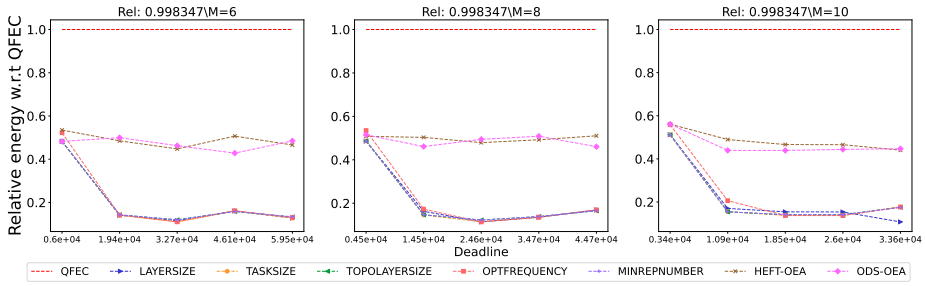

Figure 2533: Assessing the performance of different processor number on the Genome workflow.

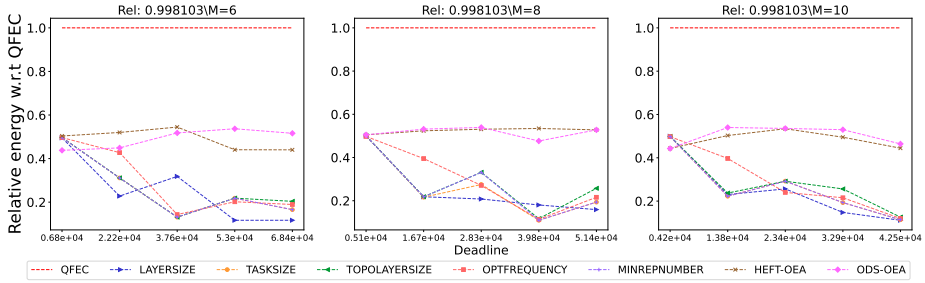

Figure 2534: Assessing the performance of different processor number on the LU.

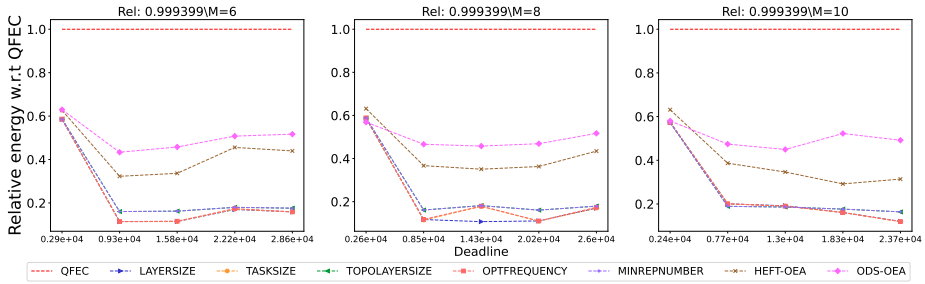

Figure 2535: Assessing the performance of different processor number on the Montage workflow.

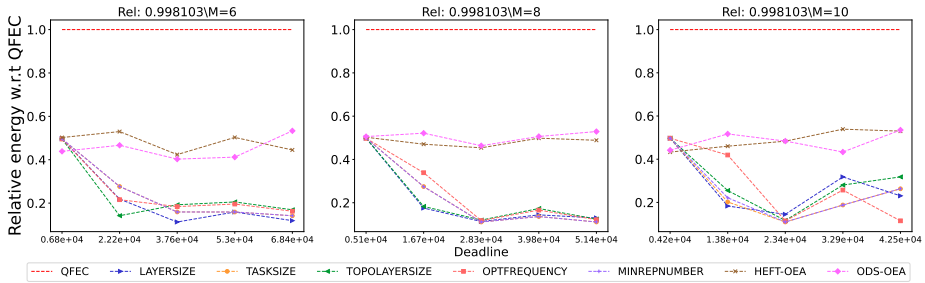

Figure 2536: Assessing the performance of different processor number on the QR.

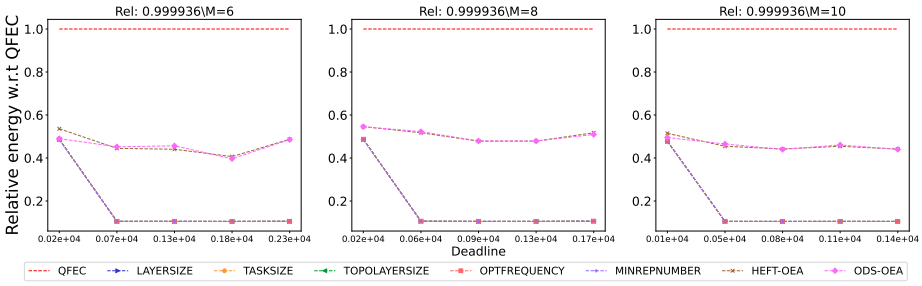

Figure 2537: Assessing the performance of different processor number on the Seismology workflow.

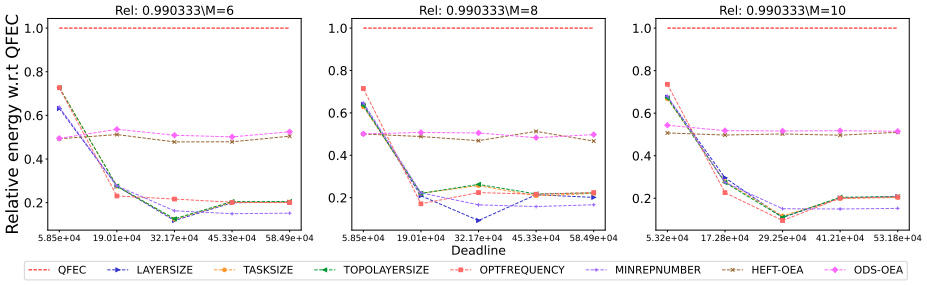

Figure 2538: Assessing the performance of different processor number on the SoyKB workflow.

H.7.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

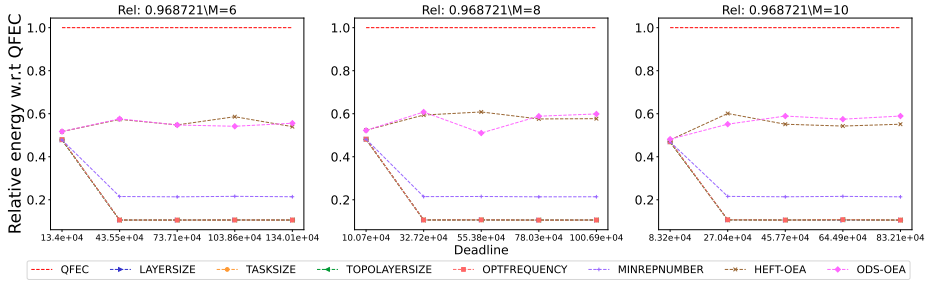

Figure 2539: Assessing the performance of different processor number on the BLAST workflow.

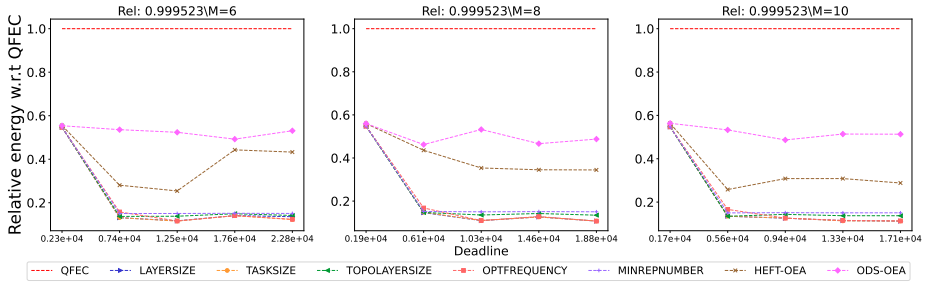

Figure 2540: Assessing the performance of different processor number on the BWA workflow.

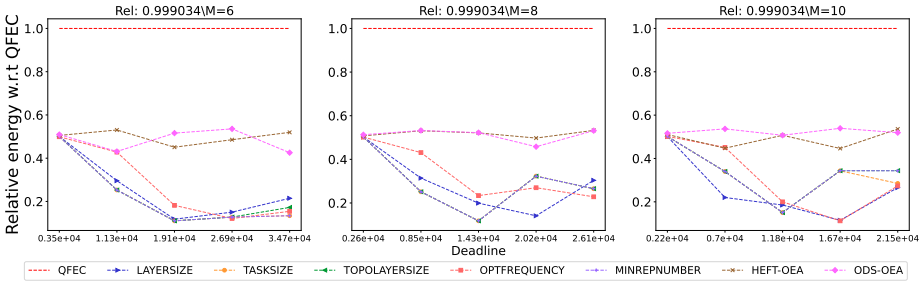

Figure 2541: Assessing the performance of different processor number on the Cholesky workflow.

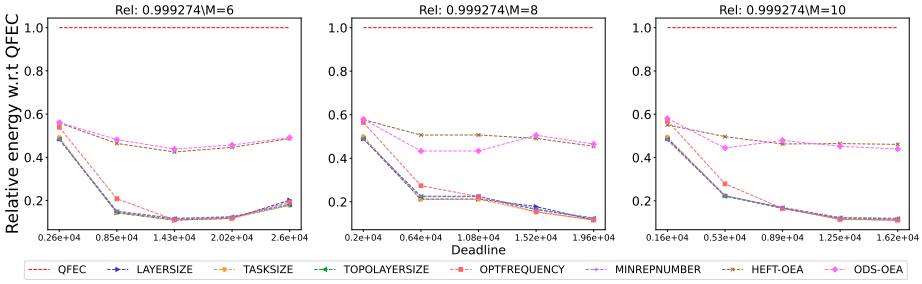

Figure 2542: Assessing the performance of different processor number on the Cycles workflow.

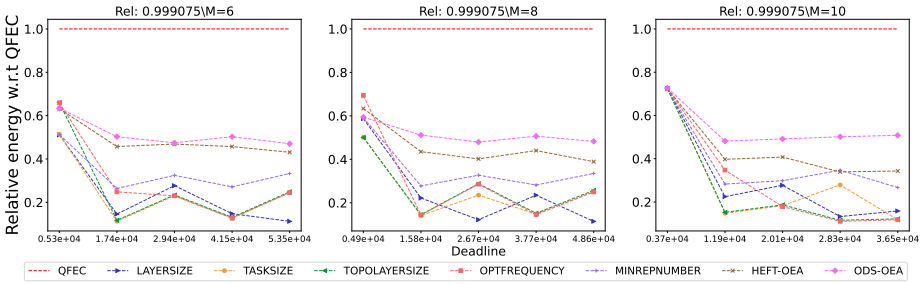

Figure 2543: Assessing the performance of different processor number on the Epigenomics workflow.

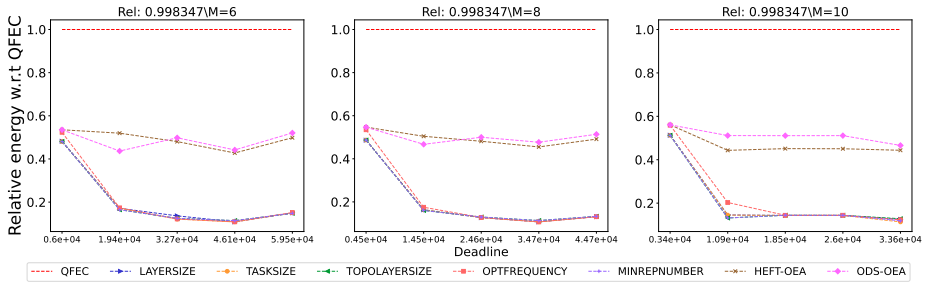

Figure 2544: Assessing the performance of different processor number on the Genome workflow.

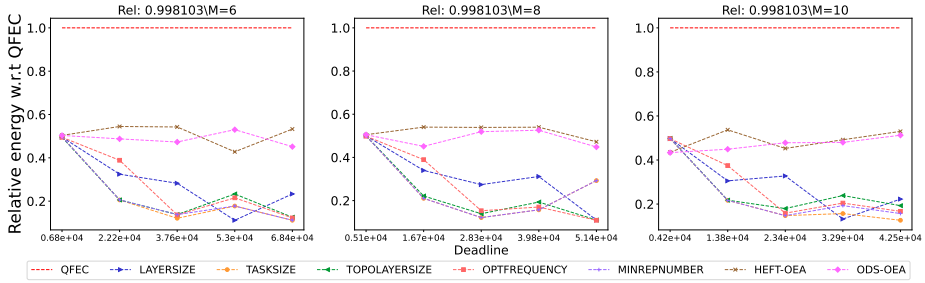

Figure 2545: Assessing the performance of different processor number on the LU.

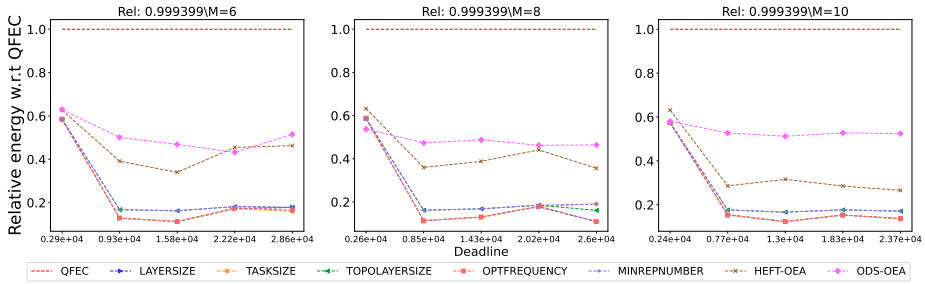

Figure 2546: Assessing the performance of different processor number on the Montage workflow.

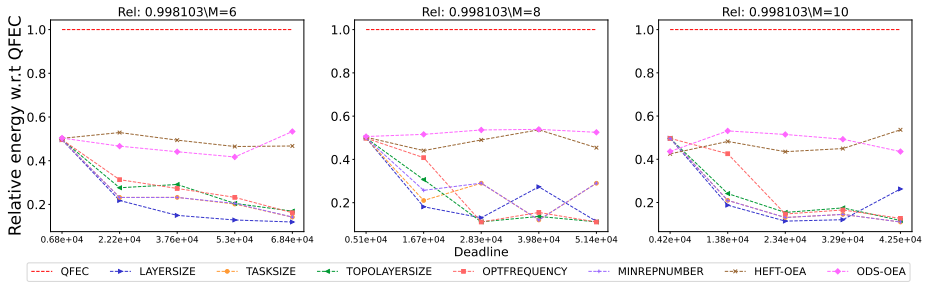

Figure 2547: Assessing the performance of different processor number on the QR.

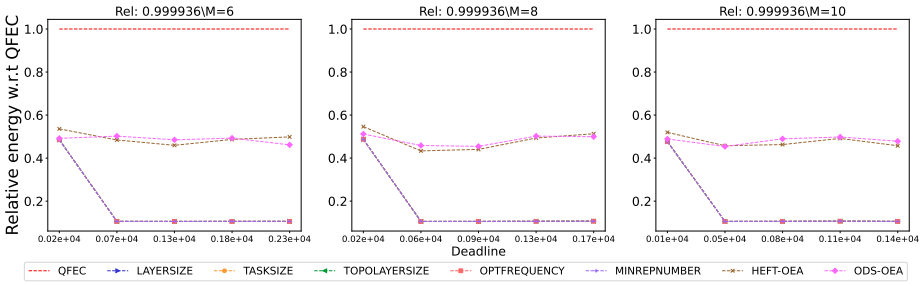

Figure 2548: Assessing the performance of different processor number on the Seismology workflow.

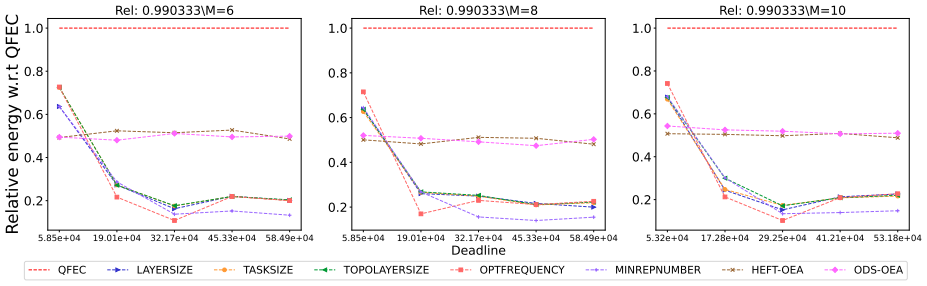

Figure 2549: Assessing the performance of different processor number on the SoyKB workflow.

H.8 $BC/WC = 0.8$

H.8.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

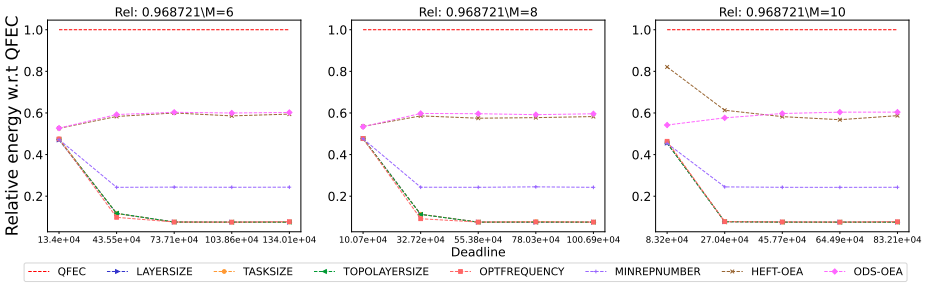

Figure 2550: Assessing the performance of different processor number on the BLAST workflow.

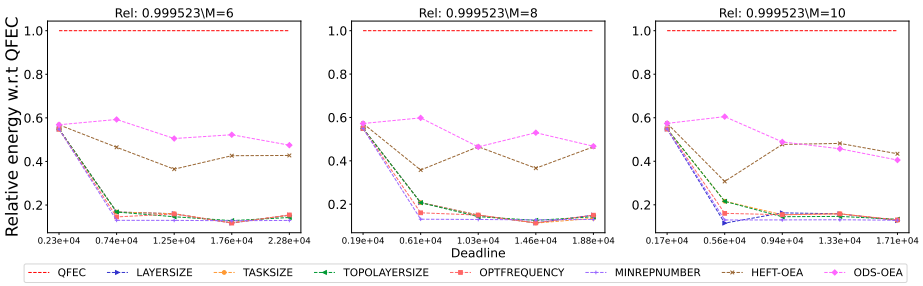

Figure 2551: Assessing the performance of different processor number on the BWA workflow.

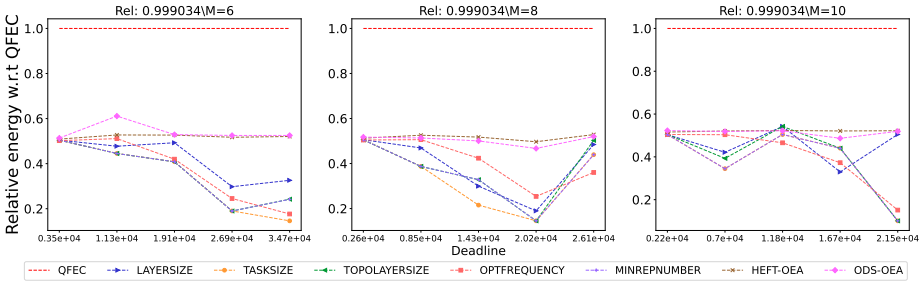

Figure 2552: Assessing the performance of different processor number on the Cholesky workflow.

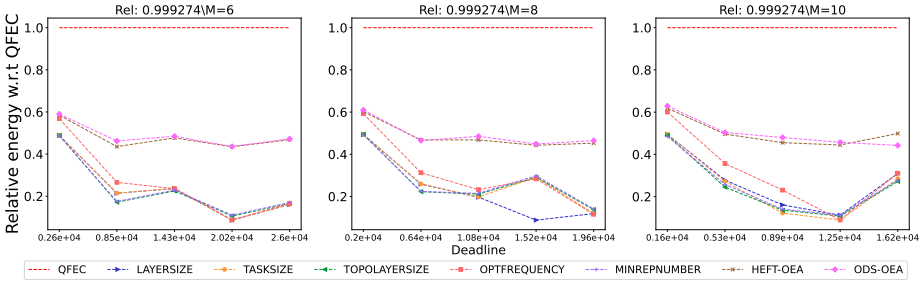

Figure 2553: Assessing the performance of different processor number on the Cycles workflow.

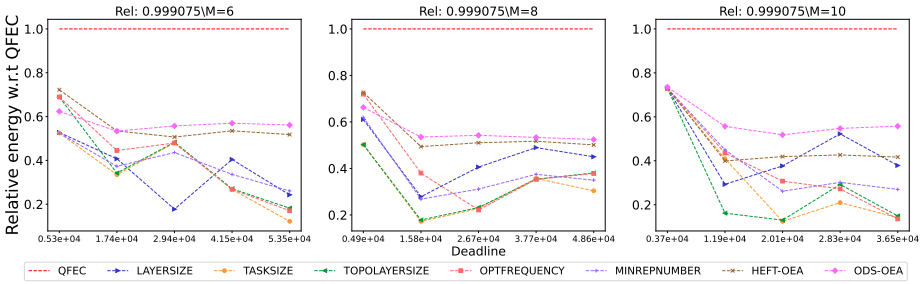

Figure 2554: Assessing the performance of different processor number on the Epigenomics workflow.

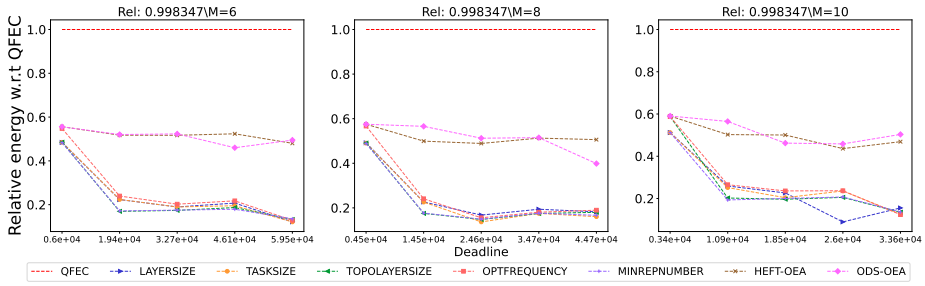

Figure 2555: Assessing the performance of different processor number on the Genome workflow.

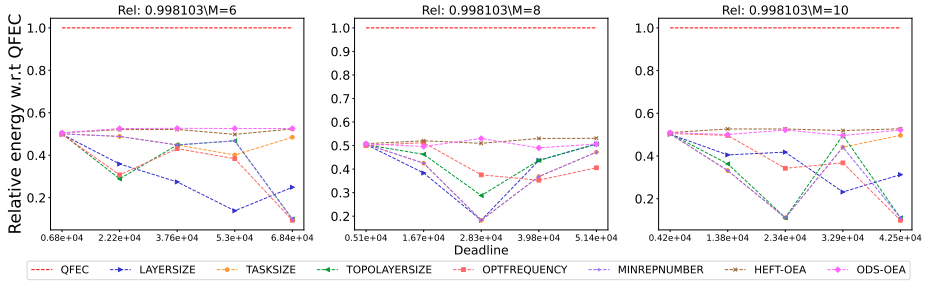

Figure 2556: Assessing the performance of different processor number on the LU.

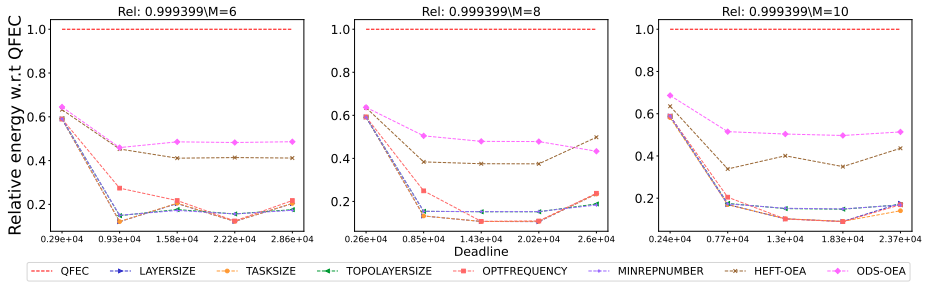

Figure 2557: Assessing the performance of different processor number on the Montage workflow.

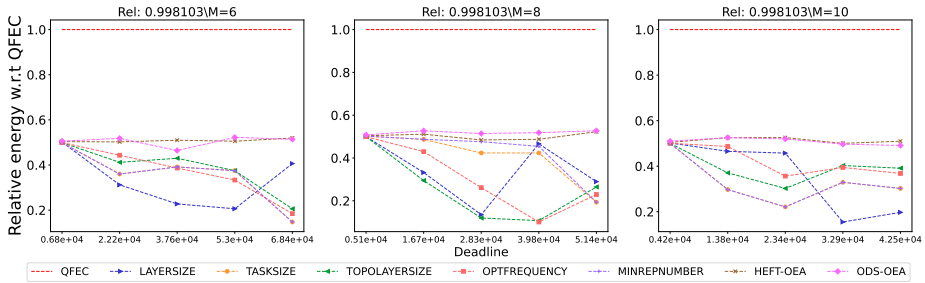

Figure 2558: Assessing the performance of different processor number on the QR.

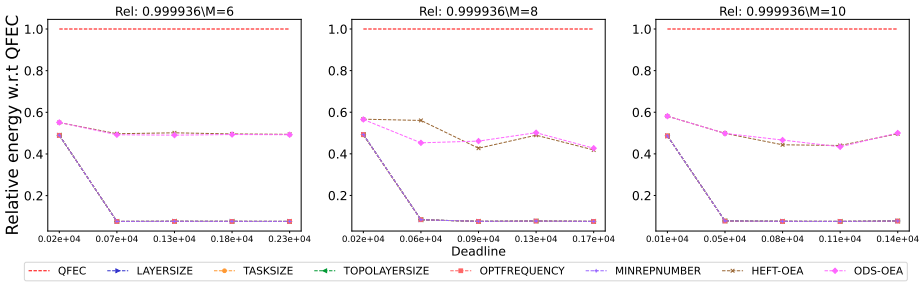

Figure 2559: Assessing the performance of different processor number on the Seismology workflow.

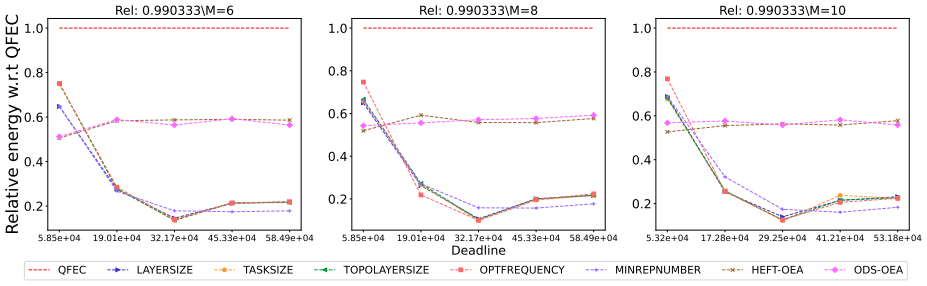

Figure 2560: Assessing the performance of different processor number on the SoyKB workflow.

H.8.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

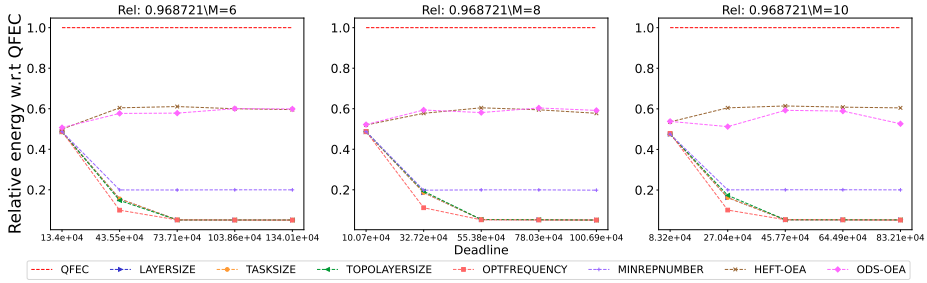

Figure 2561: Assessing the performance of different processor number on the BLAST workflow.

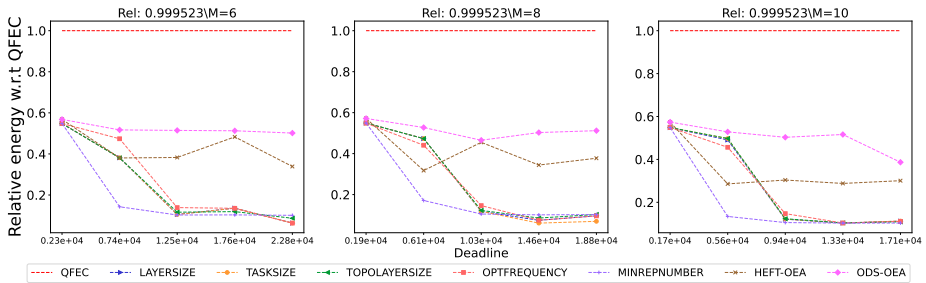

Figure 2562: Assessing the performance of different processor number on the BWA workflow.

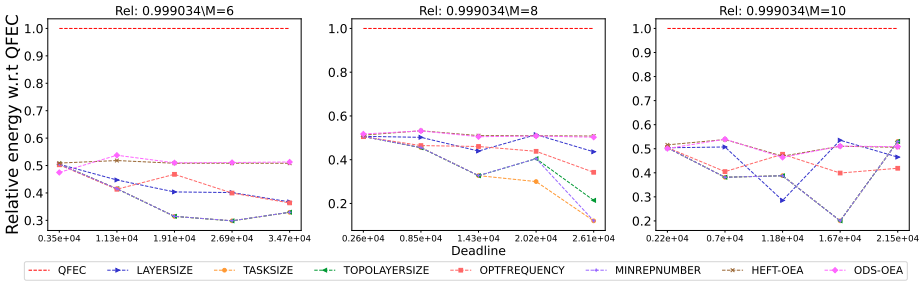

Figure 2563: Assessing the performance of different processor number on the Cholesky workflow.

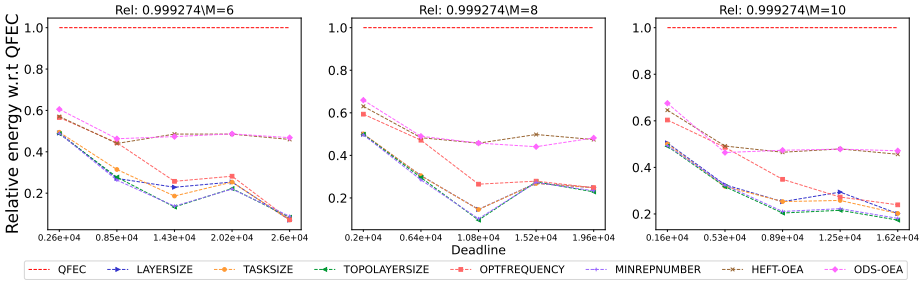

Figure 2564: Assessing the performance of different processor number on the Cycles workflow.

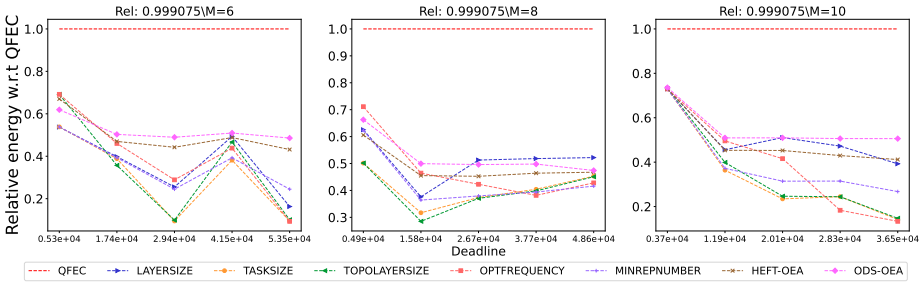

Figure 2565: Assessing the performance of different processor number on the Epigenomics workflow.

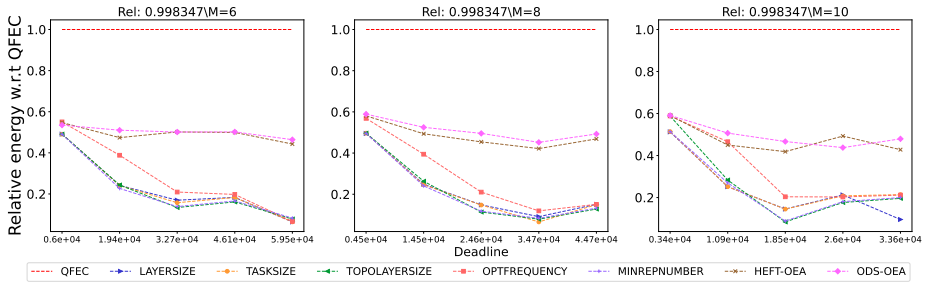

Figure 2566: Assessing the performance of different processor number on the Genome workflow.

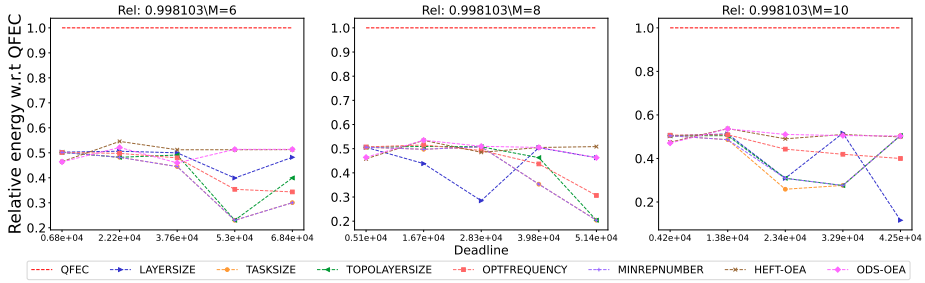

Figure 2567: Assessing the performance of different processor number on the LU.

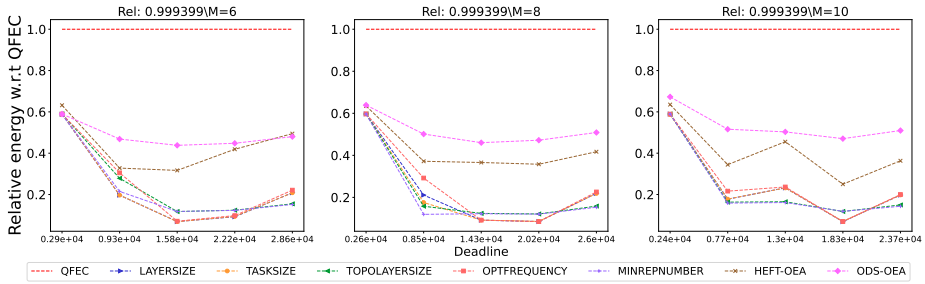

Figure 2568: Assessing the performance of different processor number on the Montage workflow.

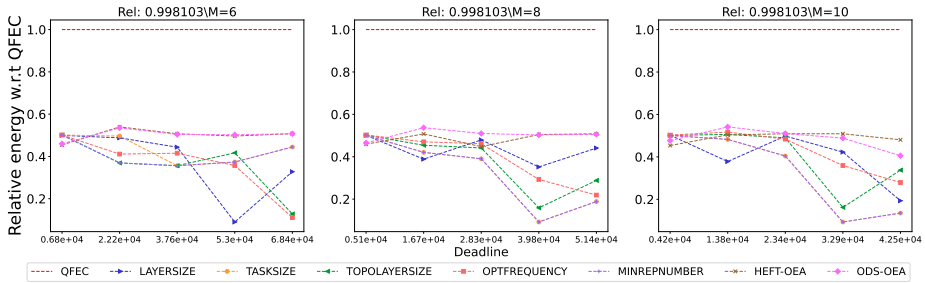

Figure 2569: Assessing the performance of different processor number on the QR.

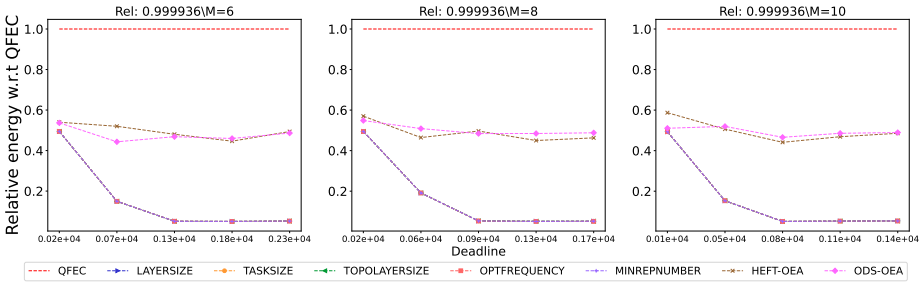

Figure 2570: Assessing the performance of different processor number on the Seismology workflow.

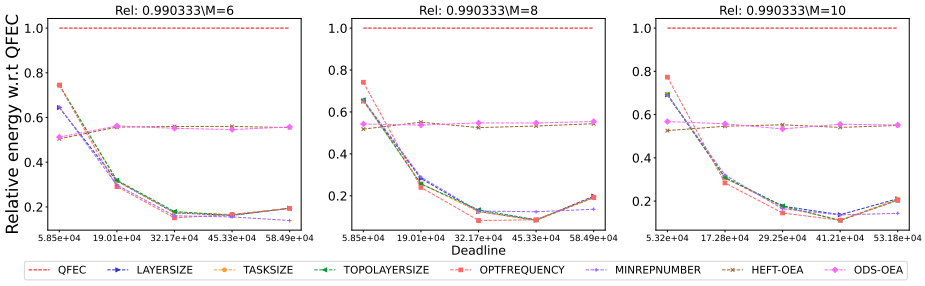

Figure 2571: Assessing the performance of different processor number on the SoyKB workflow.

H.8.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

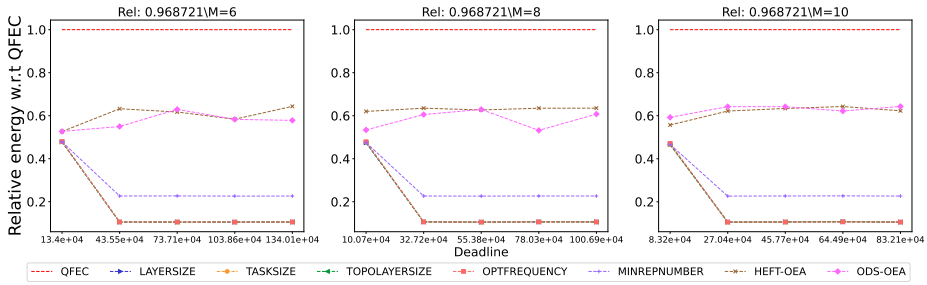

Figure 2572: Assessing the performance of different processor number on the BLAST workflow.

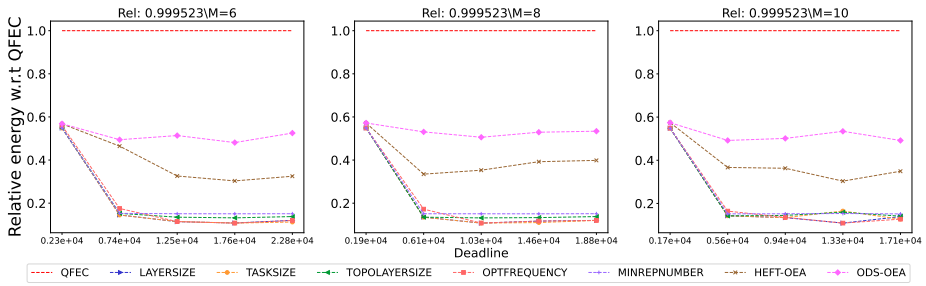

Figure 2573: Assessing the performance of different processor number on the BWA workflow.

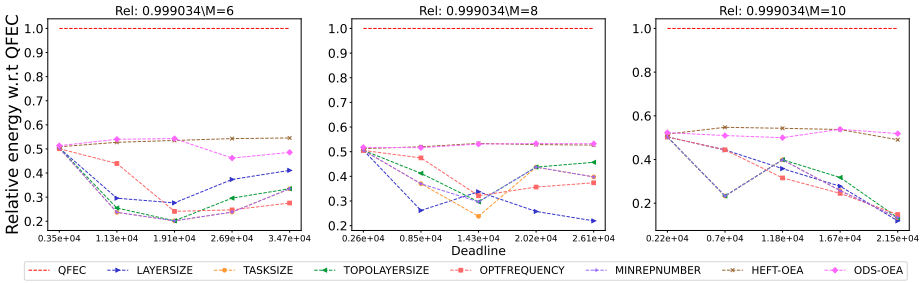

Figure 2574: Assessing the performance of different processor number on the Cholesky workflow.

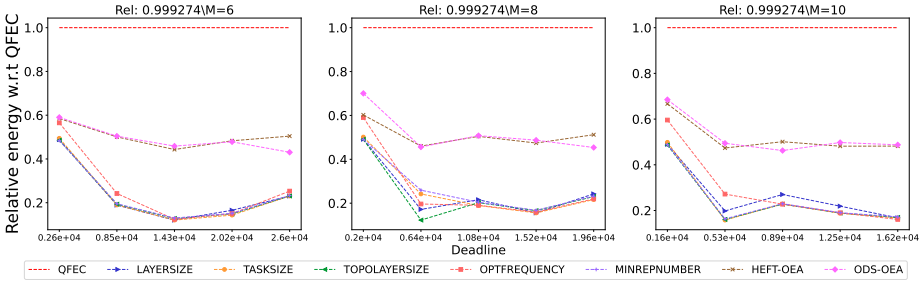

Figure 2575: Assessing the performance of different processor number on the Cycles workflow.

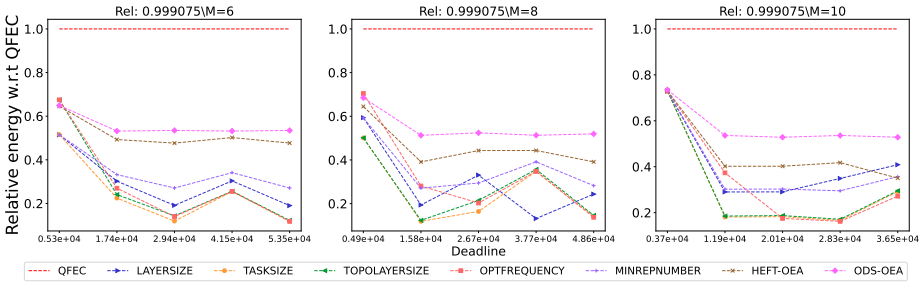

Figure 2576: Assessing the performance of different processor number on the Epigenomics workflow.

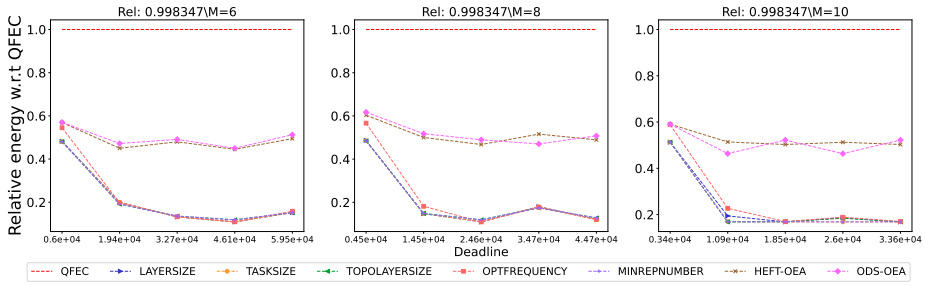

Figure 2577: Assessing the performance of different processor number on the Genome workflow.

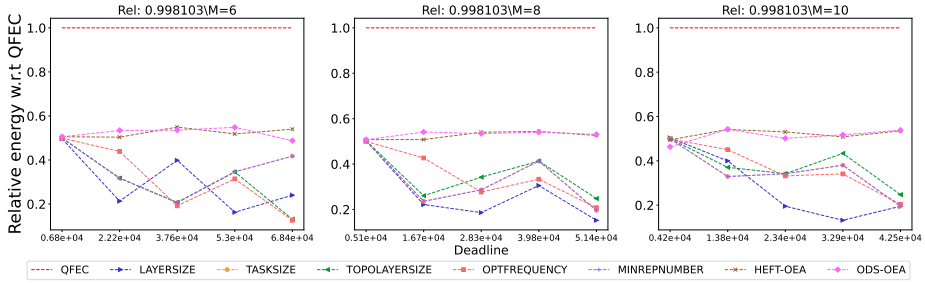

Figure 2578: Assessing the performance of different processor number on the LU.

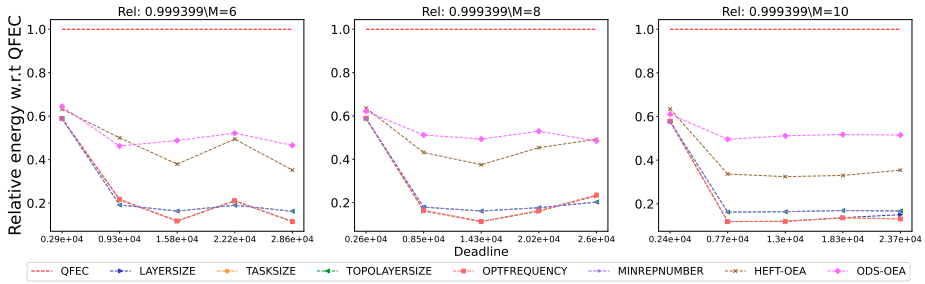

Figure 2579: Assessing the performance of different processor number on the Montage workflow.

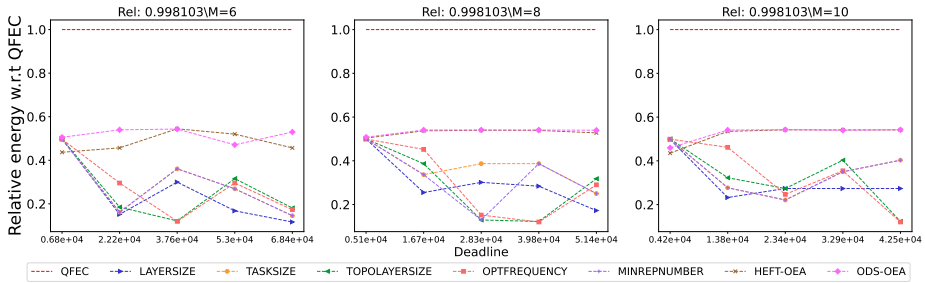

Figure 2580: Assessing the performance of different processor number on the QR.

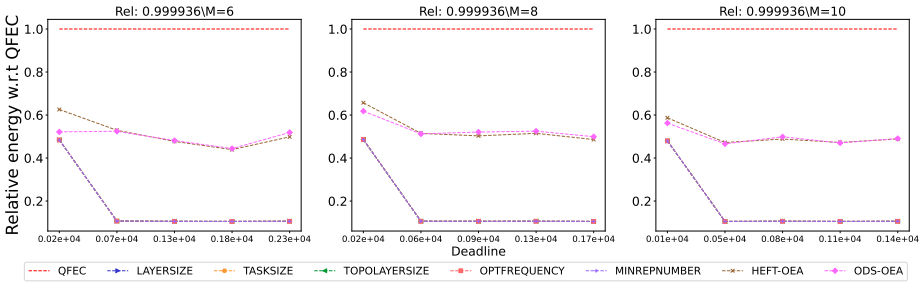

Figure 2581: Assessing the performance of different processor number on the Seismology workflow.

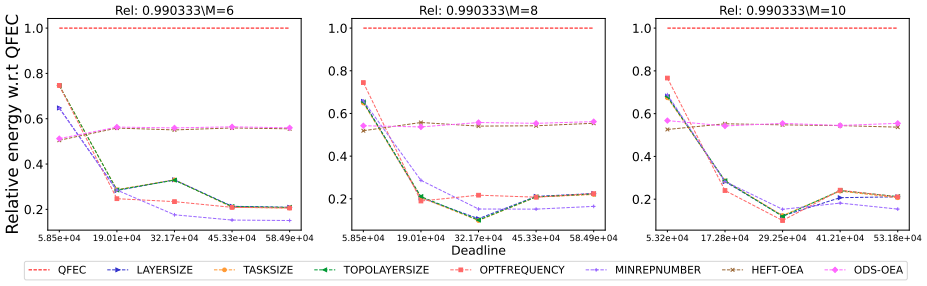

Figure 2582: Assessing the performance of different processor number on the SoyKB workflow.

H.8.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

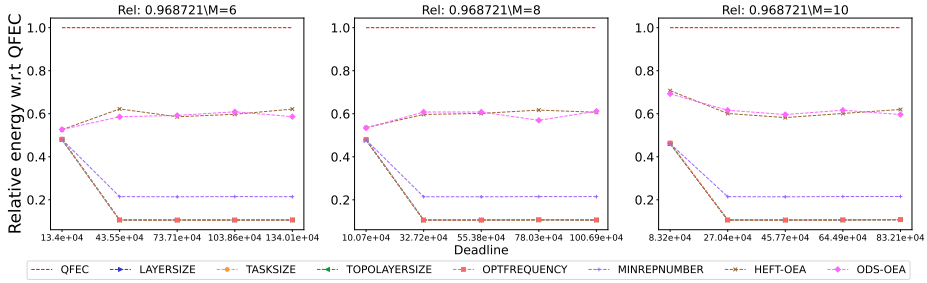

Figure 2583: Assessing the performance of different processor number on the BLAST workflow.

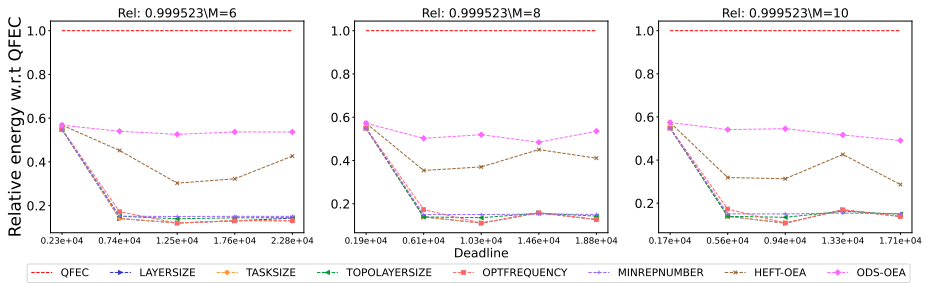

Figure 2584: Assessing the performance of different processor number on the BWA workflow.

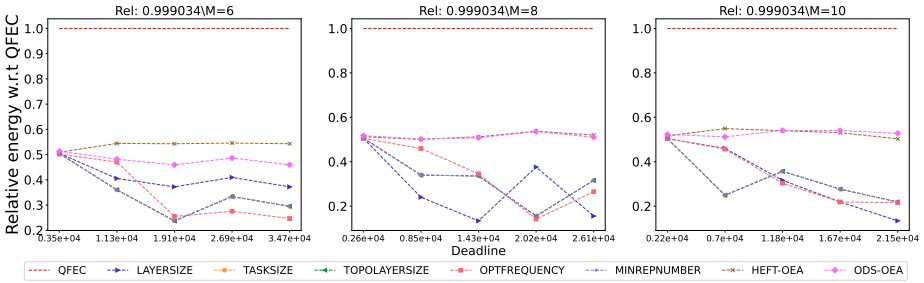

Figure 2585: Assessing the performance of different processor number on the Cholesky workflow.

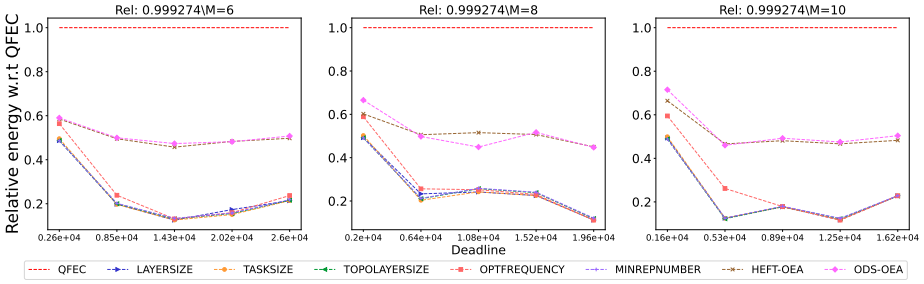

Figure 2586: Assessing the performance of different processor number on the Cycles workflow.

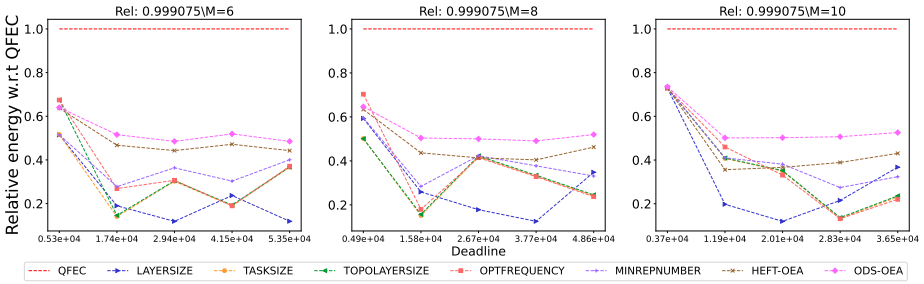

Figure 2587: Assessing the performance of different processor number on the Epigenomics workflow.

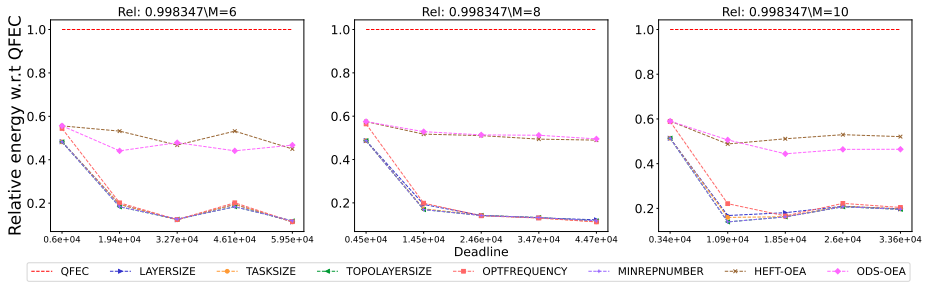

Figure 2588: Assessing the performance of different processor number on the Genome workflow.

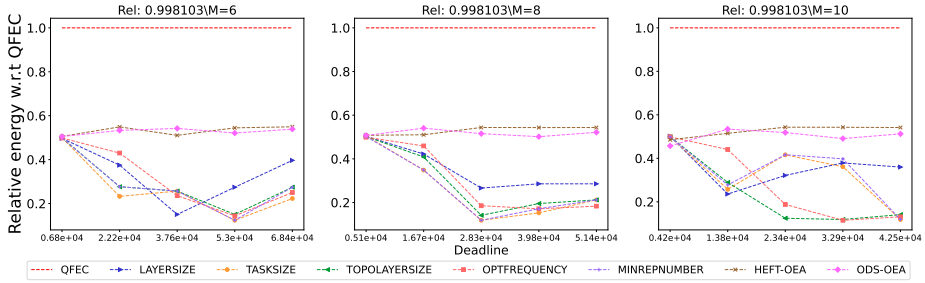

Figure 2589: Assessing the performance of different processor number on the LU.

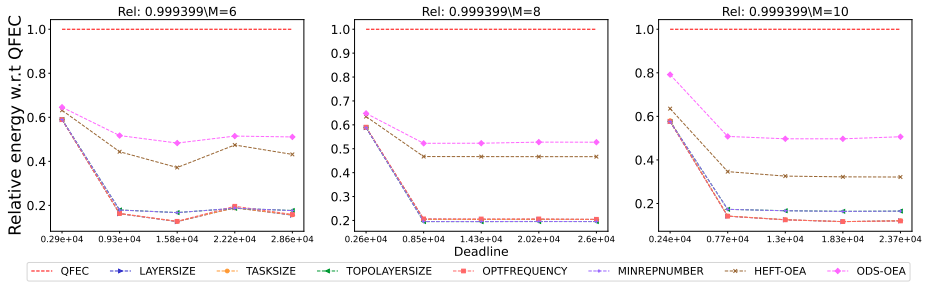

Figure 2590: Assessing the performance of different processor number on the Montage workflow.

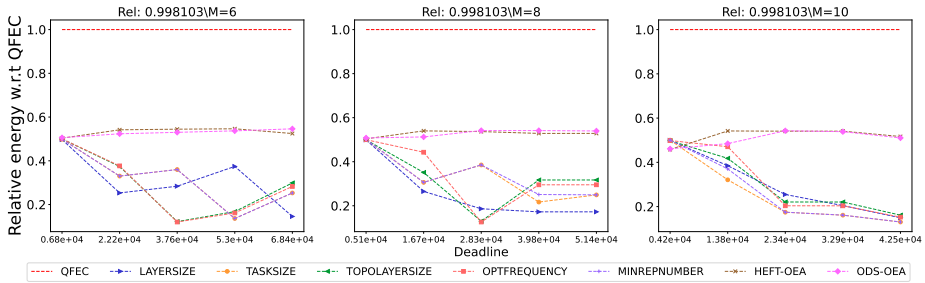

Figure 2591: Assessing the performance of different processor number on the QR.

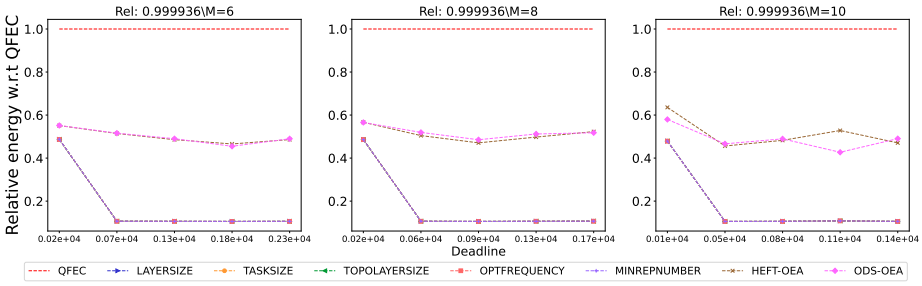

Figure 2592: Assessing the performance of different processor number on the Seismology workflow.

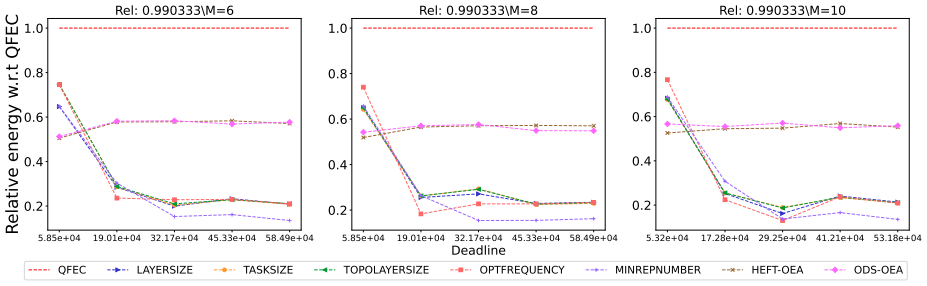

Figure 2593: Assessing the performance of different processor number on the SoyKB workflow.

H.9 $BC/WC = 0.9$

H.9.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

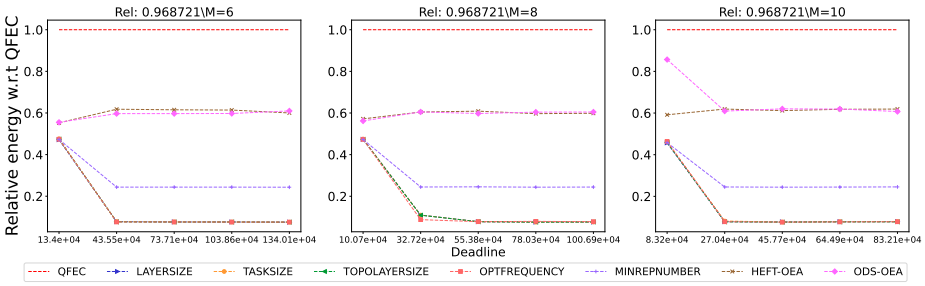

Figure 2594: Assessing the performance of different processor number on the BLAST workflow.

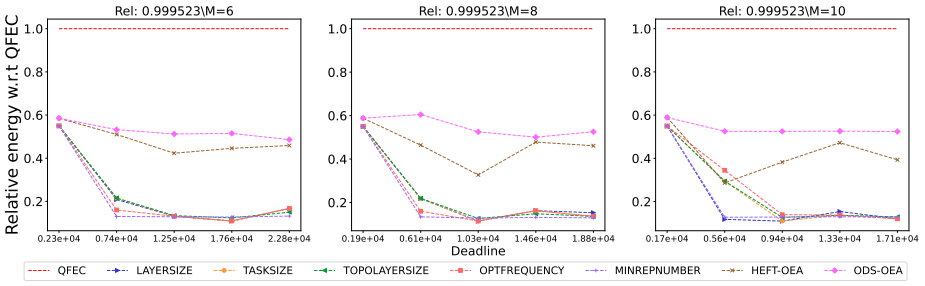

Figure 2595: Assessing the performance of different processor number on the BWA workflow.

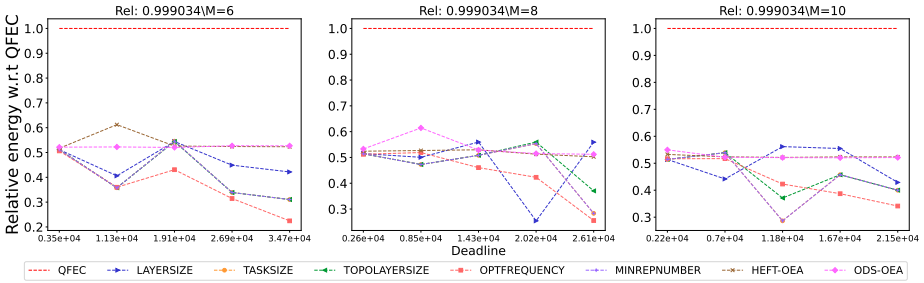

Figure 2596: Assessing the performance of different processor number on the Cholesky workflow.

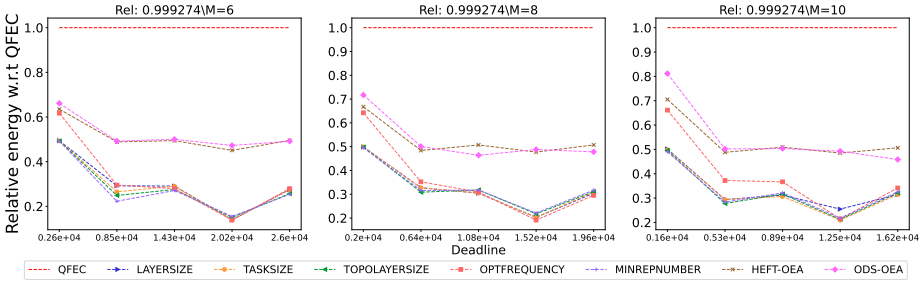

Figure 2597: Assessing the performance of different processor number on the Cycles workflow.

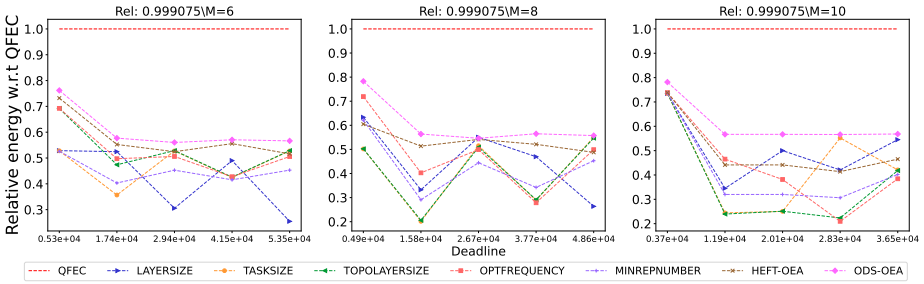

Figure 2598: Assessing the performance of different processor number on the Epigenomics workflow.

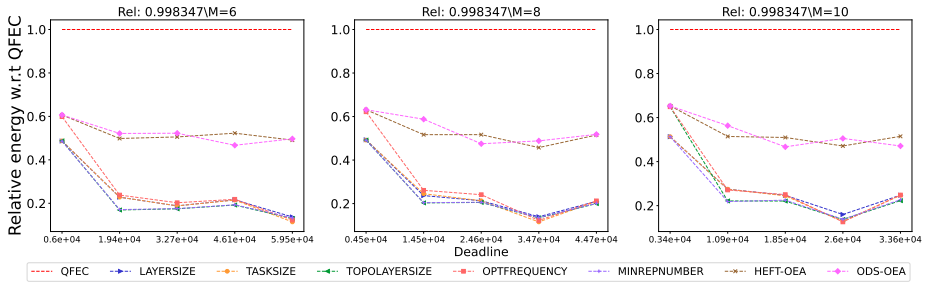

Figure 2599: Assessing the performance of different processor number on the Genome workflow.

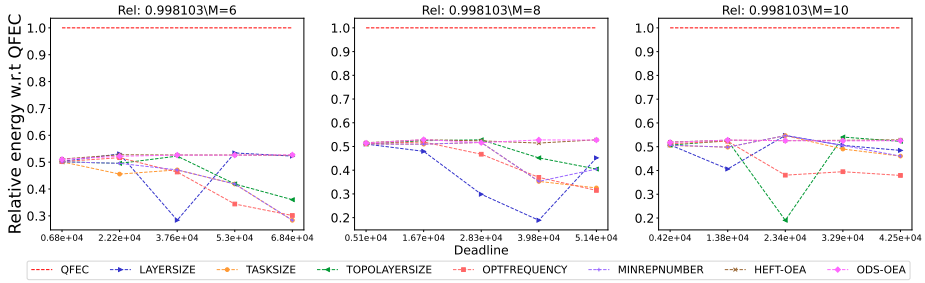

Figure 2600: Assessing the performance of different processor number on the LU.

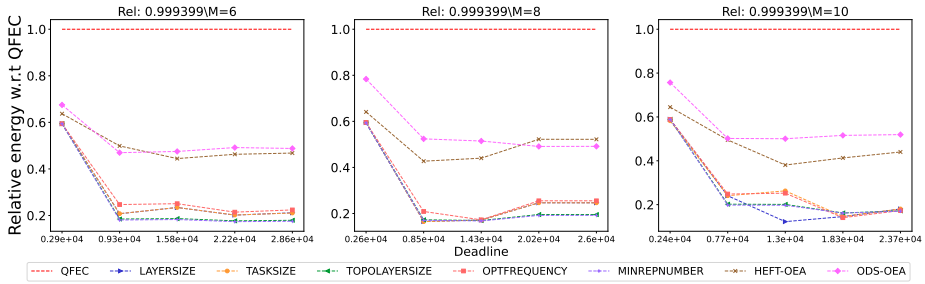

Figure 2601: Assessing the performance of different processor number on the Montage workflow.

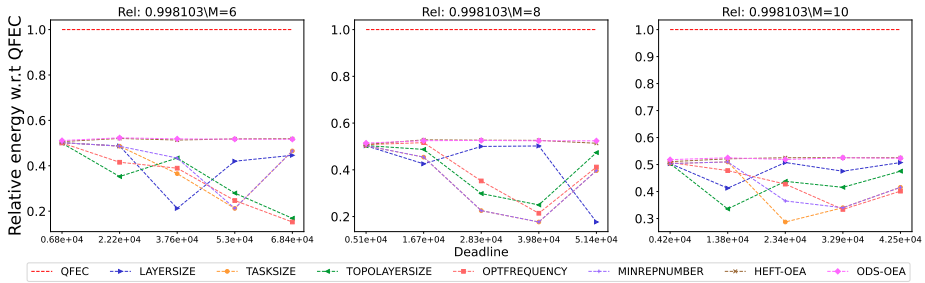

Figure 2602: Assessing the performance of different processor number on the QR.

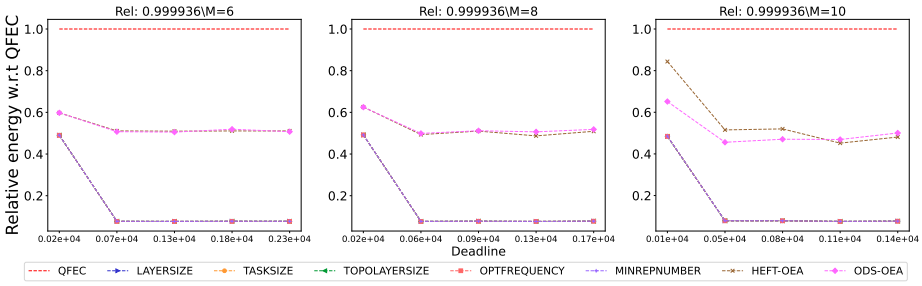

Figure 2603: Assessing the performance of different processor number on the Seismology workflow.

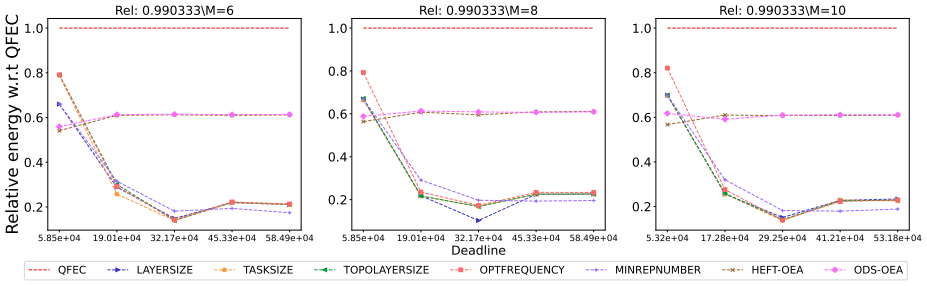

Figure 2604: Assessing the performance of different processor number on the SoyKB workflow.

H.9.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

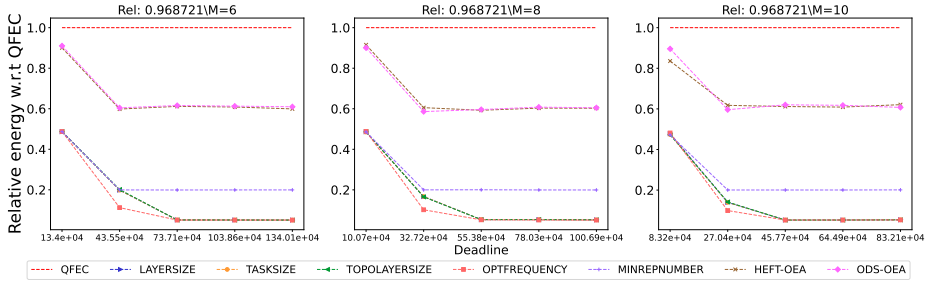

Figure 2605: Assessing the performance of different processor number on the BLAST workflow.

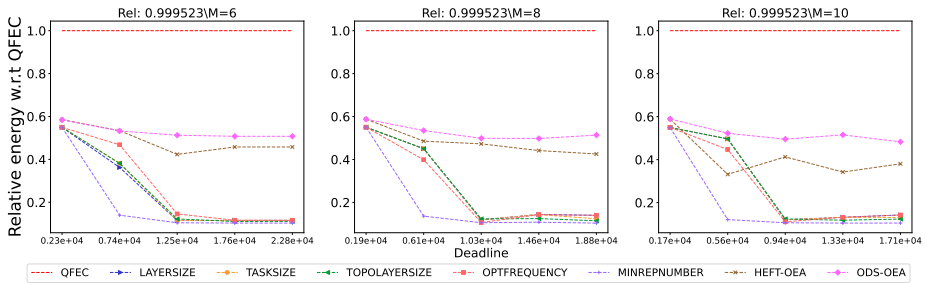

Figure 2606: Assessing the performance of different processor number on the BWA workflow.

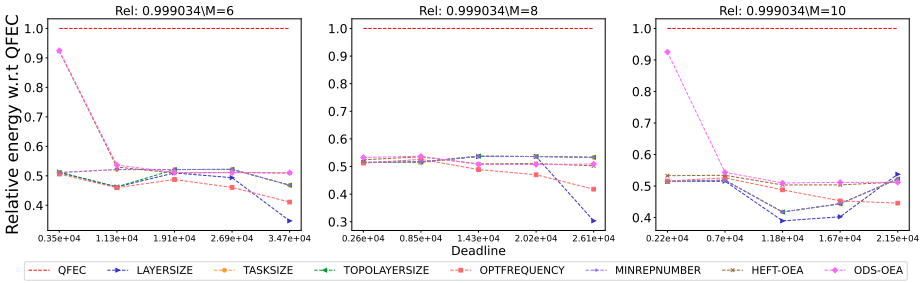

Figure 2607: Assessing the performance of different processor number on the Cholesky workflow.

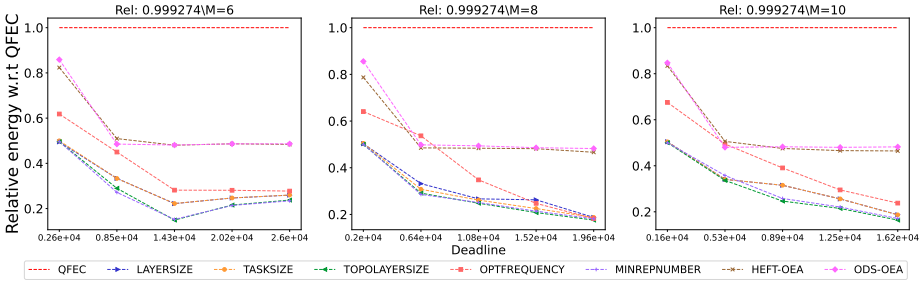

Figure 2608: Assessing the performance of different processor number on the Cycles workflow.

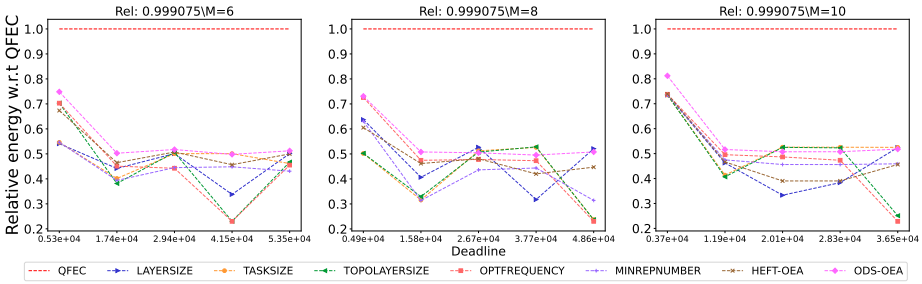

Figure 2609: Assessing the performance of different processor number on the Epigenomics workflow.

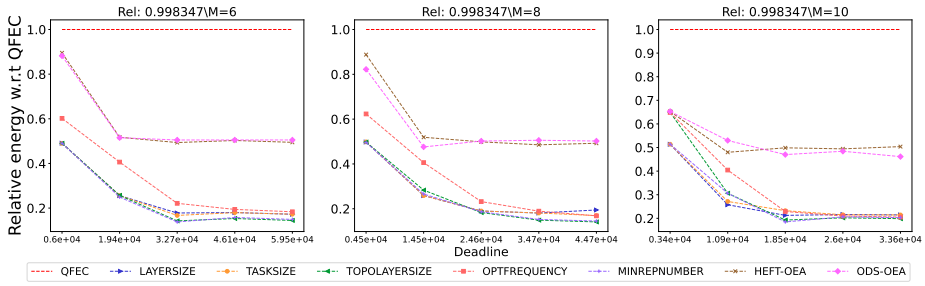

Figure 2610: Assessing the performance of different processor number on the Genome workflow.

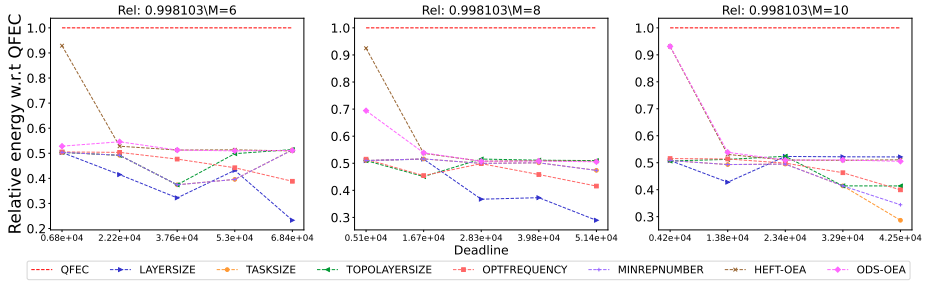

Figure 2611: Assessing the performance of different processor number on the LU.

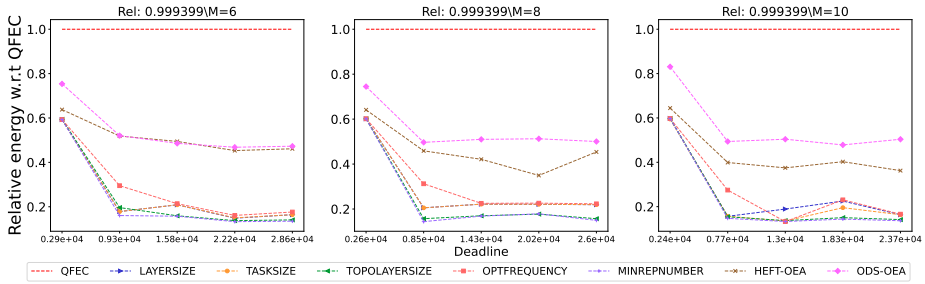

Figure 2612: Assessing the performance of different processor number on the Montage workflow.

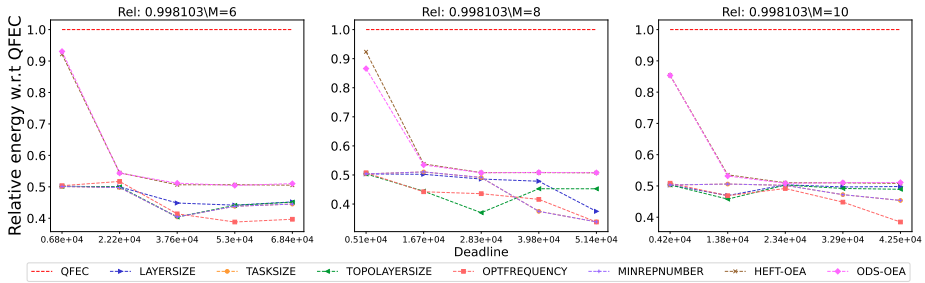

Figure 2613: Assessing the performance of different processor number on the QR.

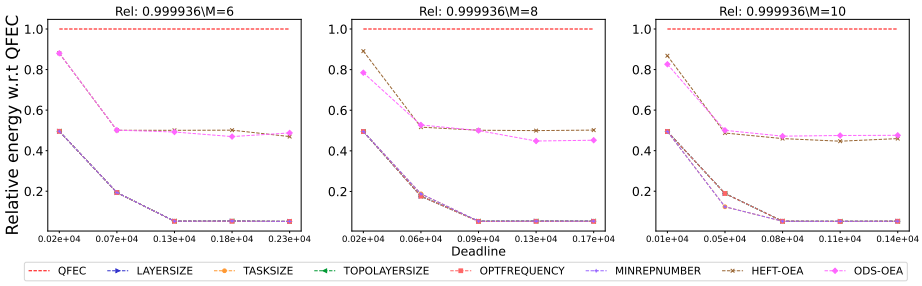

Figure 2614: Assessing the performance of different processor number on the Seismology workflow.

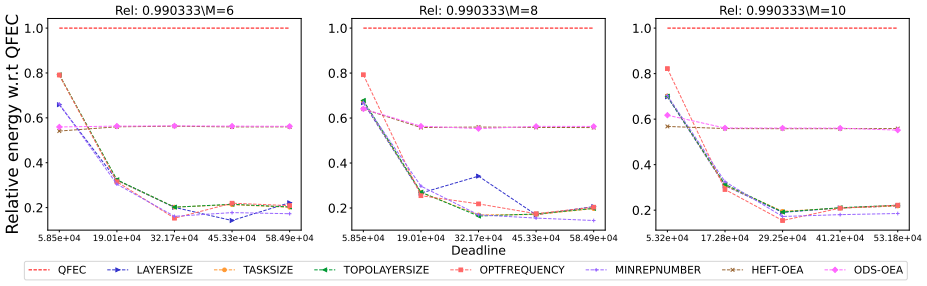

Figure 2615: Assessing the performance of different processor number on the SoyKB workflow.

H.9.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

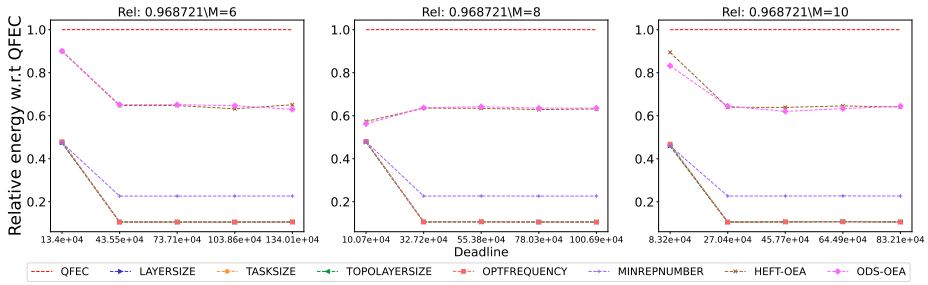

Figure 2616: Assessing the performance of different processor number on the BLAST workflow.

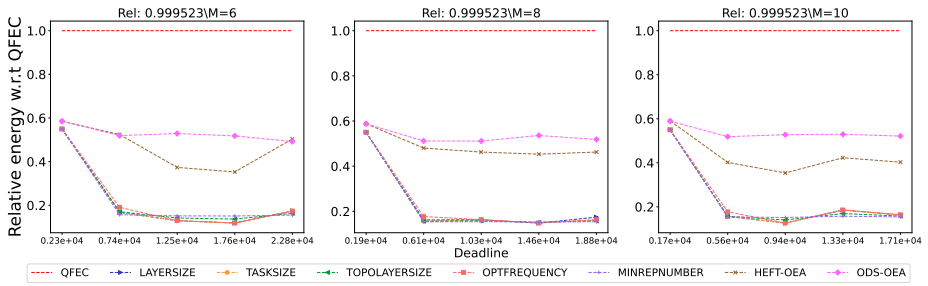

Figure 2617: Assessing the performance of different processor number on the BWA workflow.

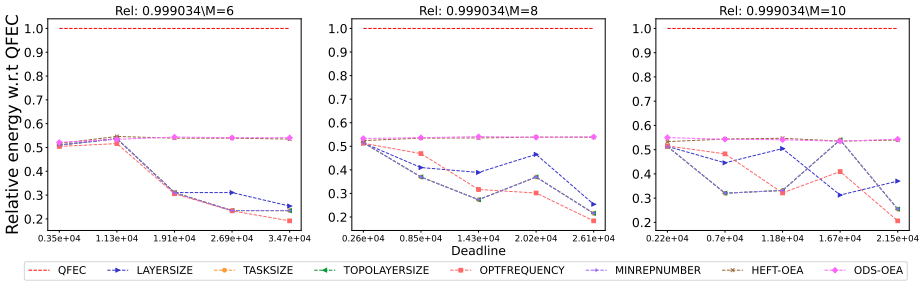

Figure 2618: Assessing the performance of different processor number on the Cholesky workflow.

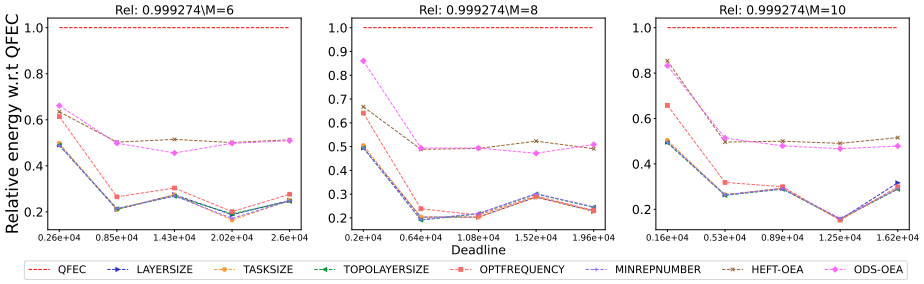

Figure 2619: Assessing the performance of different processor number on the Cycles workflow.

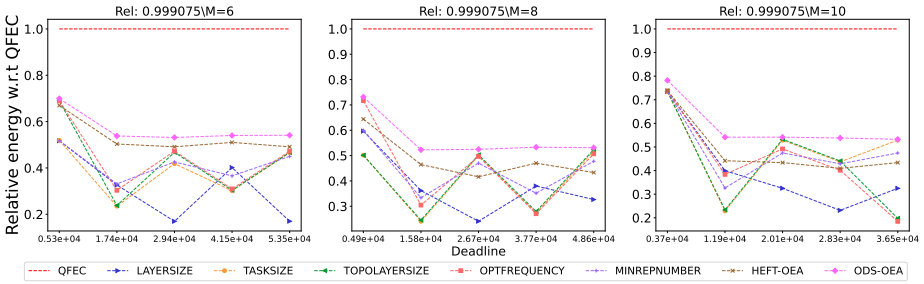

Figure 2620: Assessing the performance of different processor number on the Epigenomics workflow.

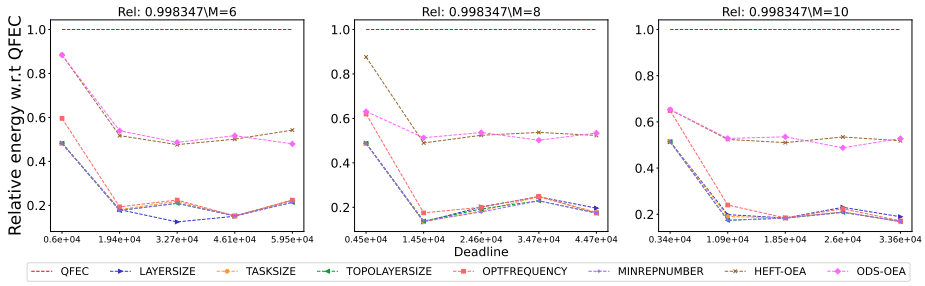

Figure 2621: Assessing the performance of different processor number on the Genome workflow.

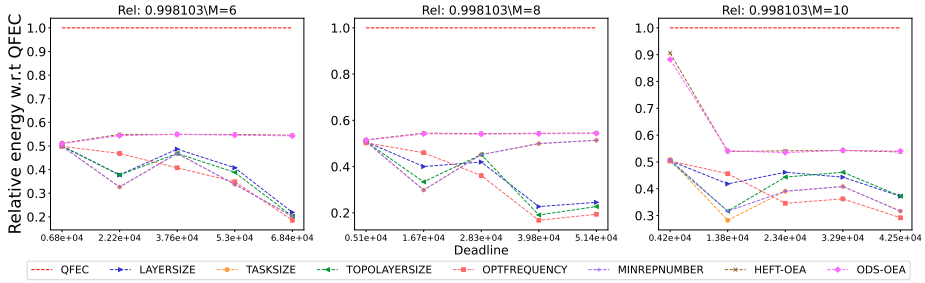

Figure 2622: Assessing the performance of different processor number on the LU.

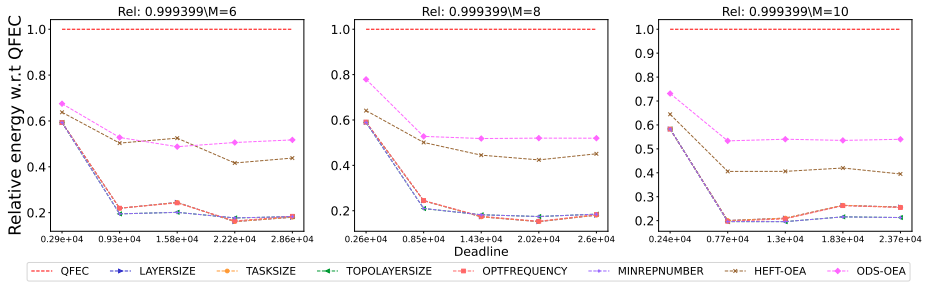

Figure 2623: Assessing the performance of different processor number on the Montage workflow.

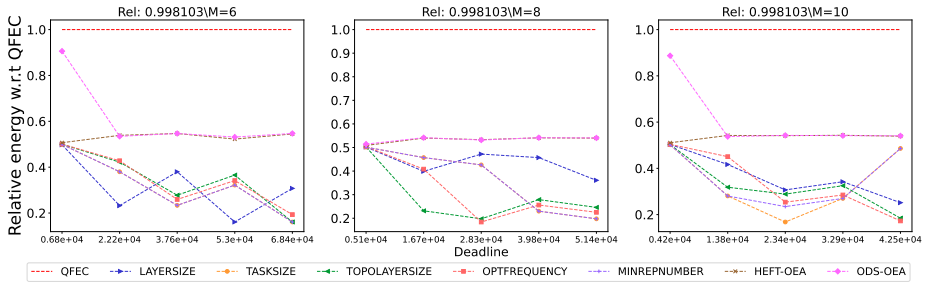

Figure 2624: Assessing the performance of different processor number on the QR.

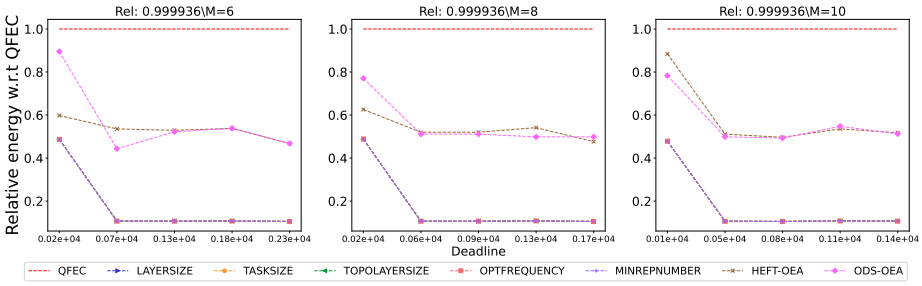

Figure 2625: Assessing the performance of different processor number on the Seismology workflow.

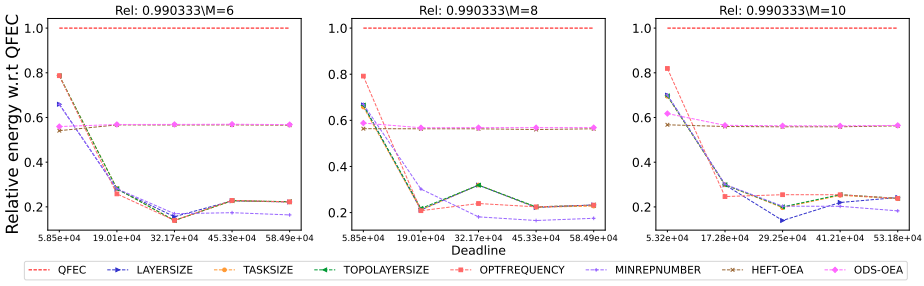

Figure 2626: Assessing the performance of different processor number on the SoyKB workflow.

H.9.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

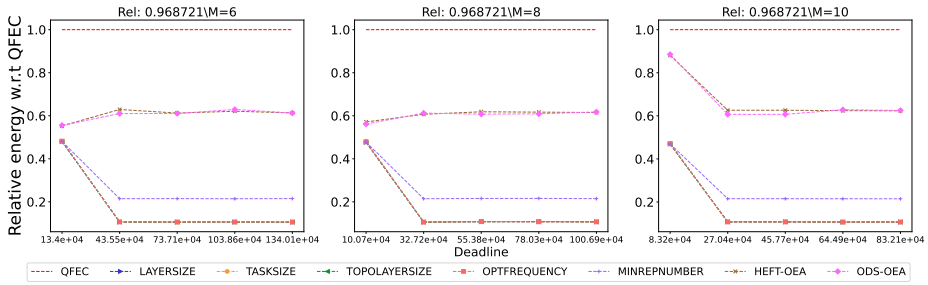

Figure 2627: Assessing the performance of different processor number on the BLAST workflow.

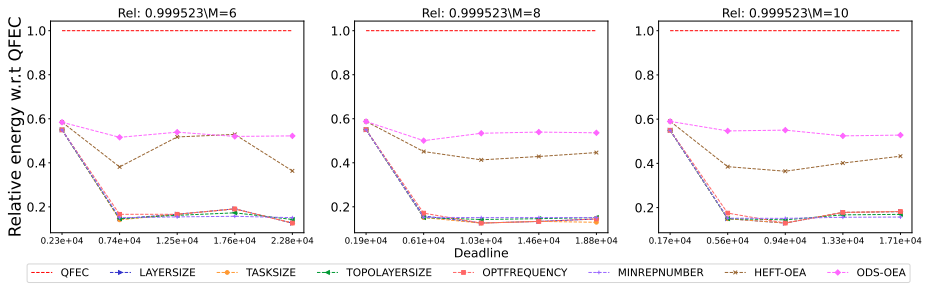

Figure 2628: Assessing the performance of different processor number on the BWA workflow.

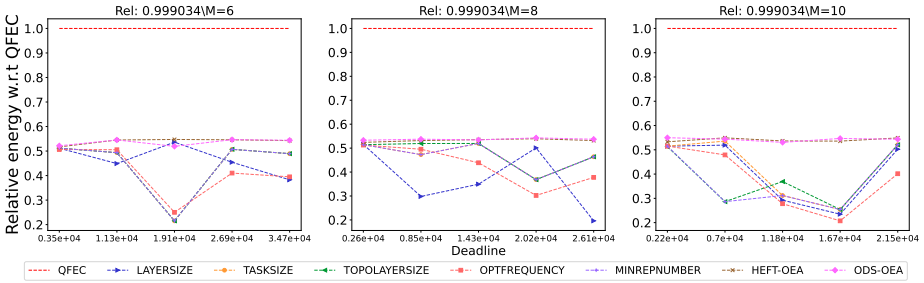

Figure 2629: Assessing the performance of different processor number on the Cholesky workflow.

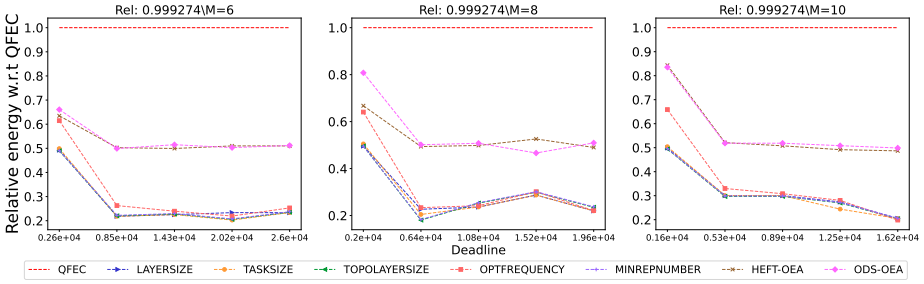

Figure 2630: Assessing the performance of different processor number on the Cycles workflow.

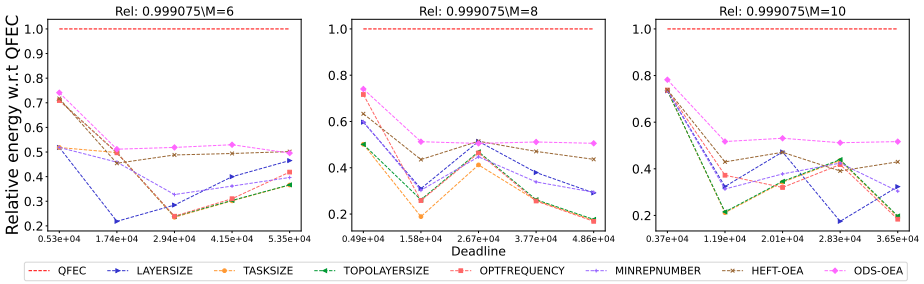

Figure 2631: Assessing the performance of different processor number on the Epigenomics workflow.

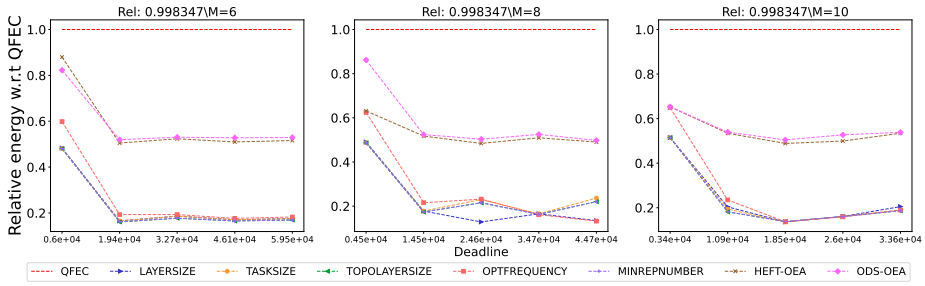

Figure 2632: Assessing the performance of different processor number on the Genome workflow.

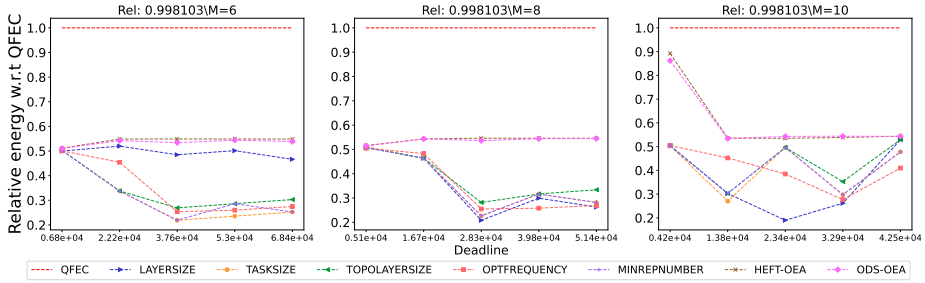

Figure 2633: Assessing the performance of different processor number on the LU.

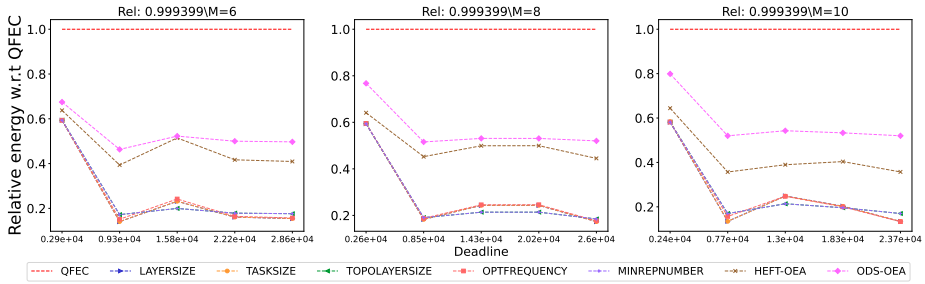

Figure 2634: Assessing the performance of different processor number on the Montage workflow.

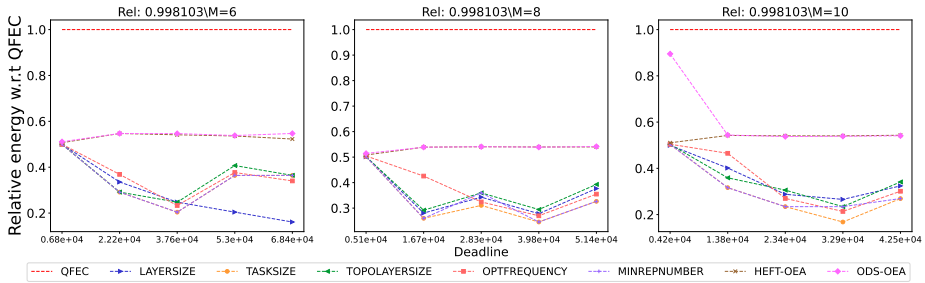

Figure 2635: Assessing the performance of different processor number on the QR.

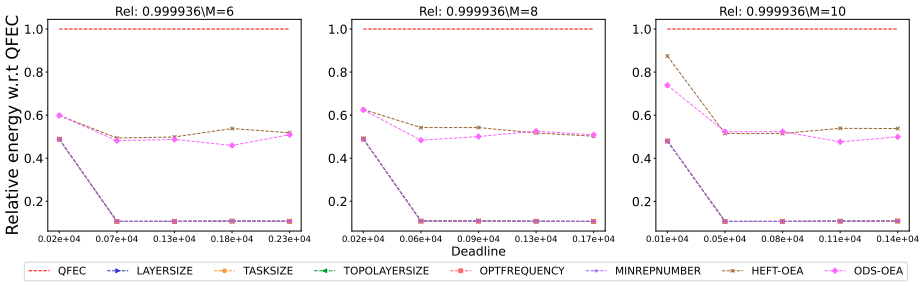

Figure 2636: Assessing the performance of different processor number on the Seismology workflow.

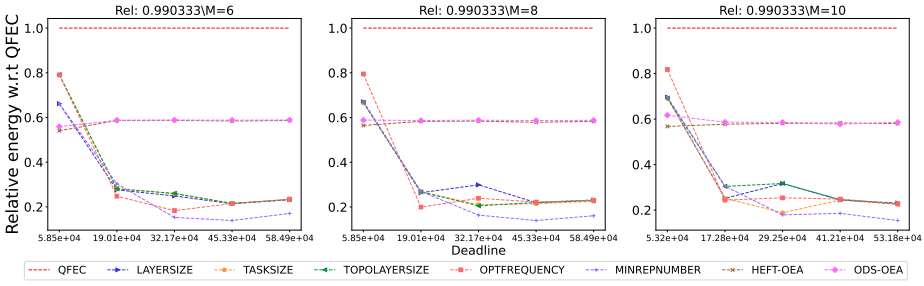

Figure 2637: Assessing the performance of different processor number on the SoyKB workflow.

I Study of different processor number influence when actual execution times are drawn from a truncated uniform distribution

I.1 $BC/WC = 0.1$

I.1.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

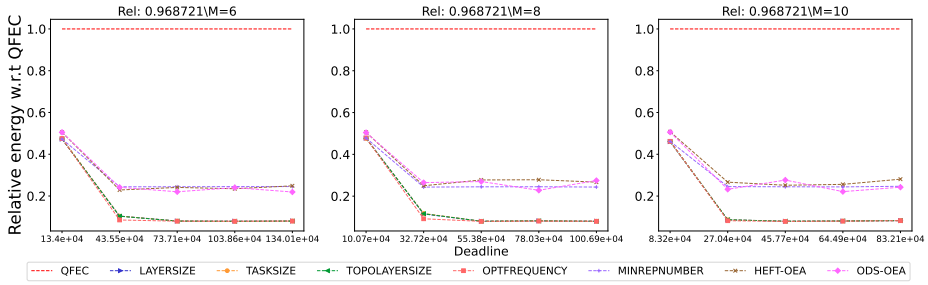

Figure 2638: Assessing the performance of different processor number on the BLAST workflow.

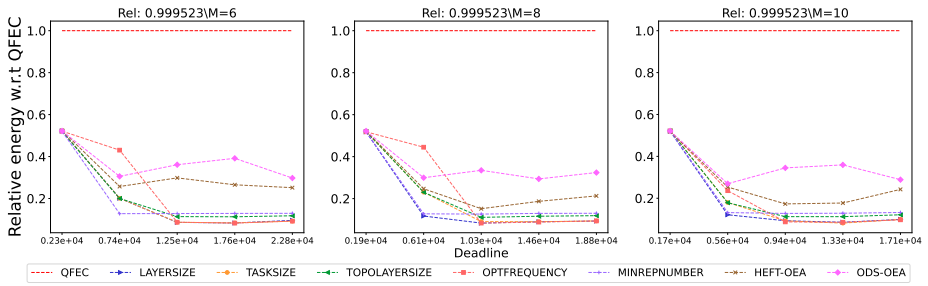

Figure 2639: Assessing the performance of different processor number on the BWA workflow.

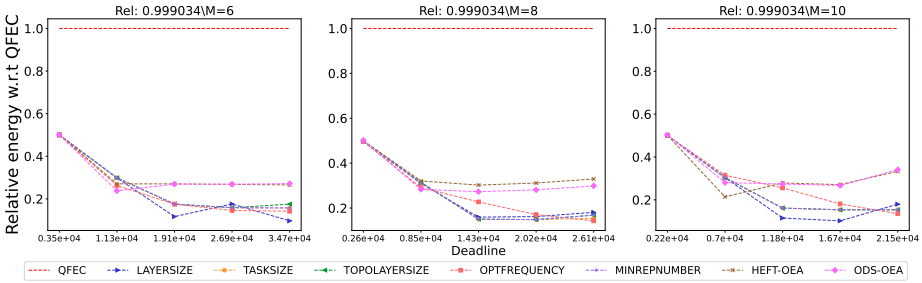

Figure 2640: Assessing the performance of different processor number on the Cholesky workflow.

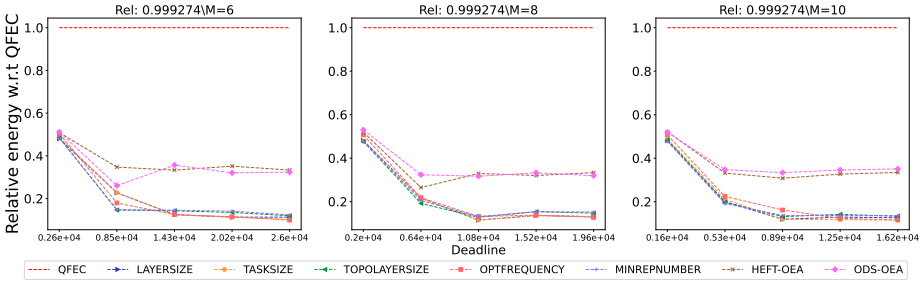

Figure 2641: Assessing the performance of different processor number on the Cycles workflow.

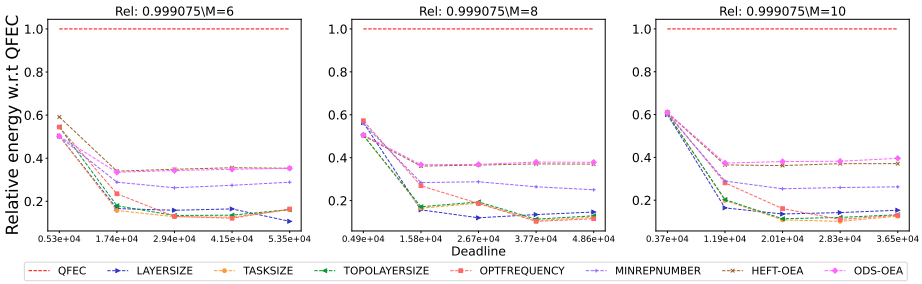

Figure 2642: Assessing the performance of different processor number on the Epigenomics workflow.

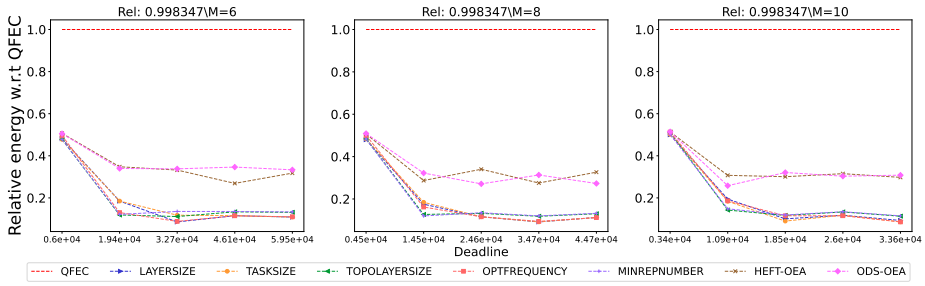

Figure 2643: Assessing the performance of different processor number on the Genome workflow.

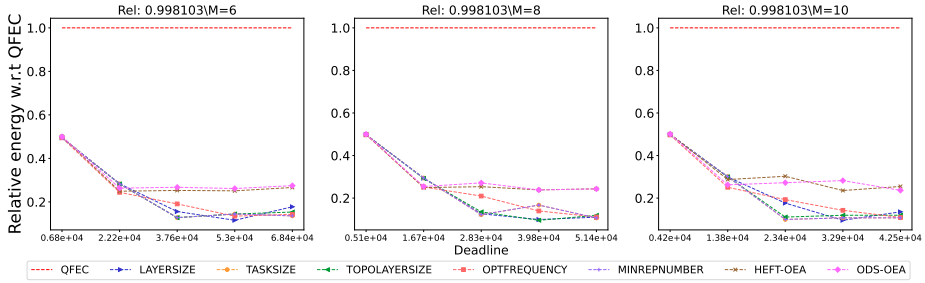

Figure 2644: Assessing the performance of different processor number on the LU.

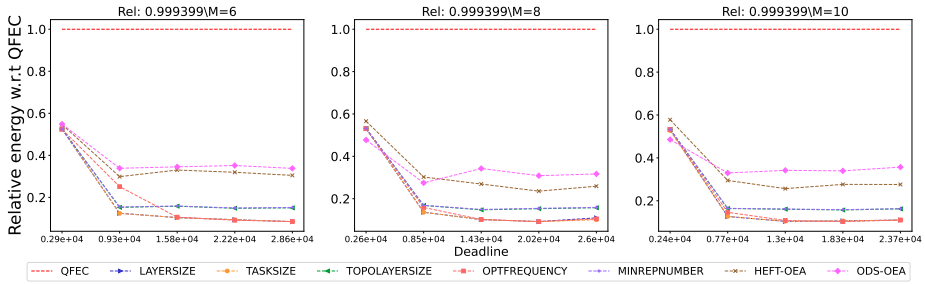

Figure 2645: Assessing the performance of different processor number on the Montage workflow.

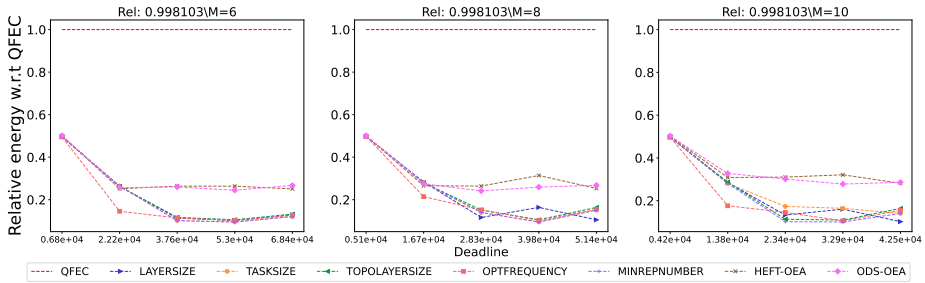

Figure 2646: Assessing the performance of different processor number on the QR.

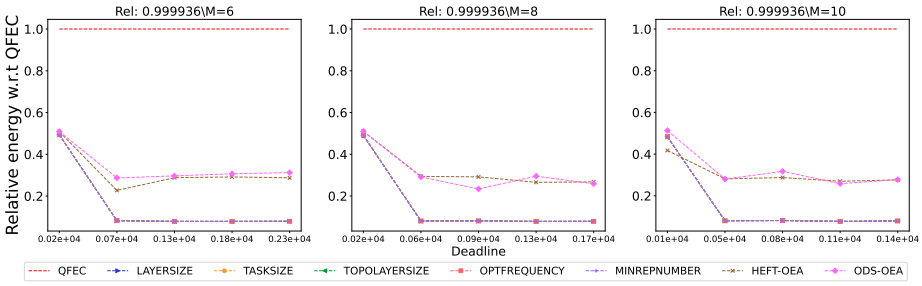

Figure 2647: Assessing the performance of different processor number on the Seismology workflow.

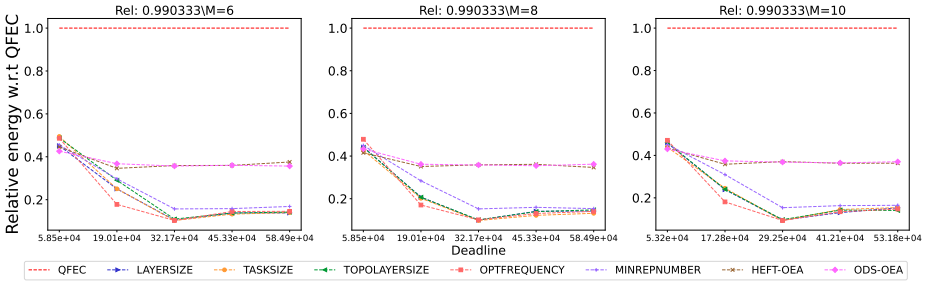

Figure 2648: Assessing the performance of different processor number on the SoyKB workflow.

I.1.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

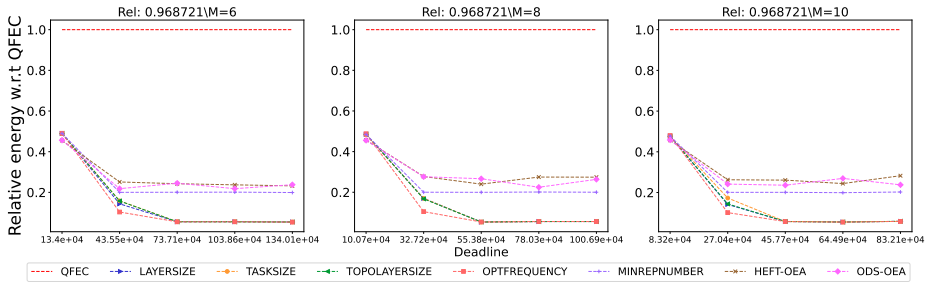

Figure 2649: Assessing the performance of different processor number on the BLAST workflow.

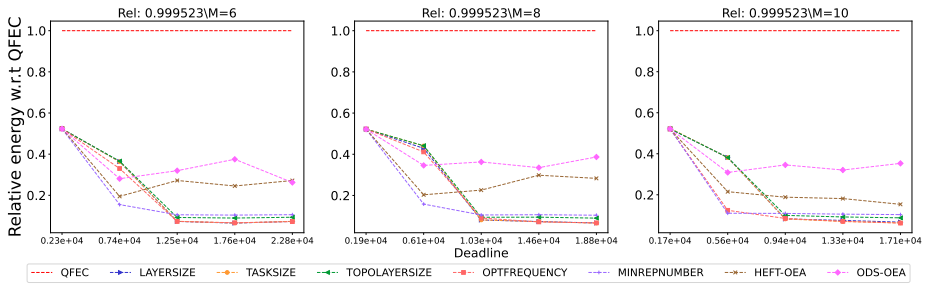

Figure 2650: Assessing the performance of different processor number on the BWA workflow.

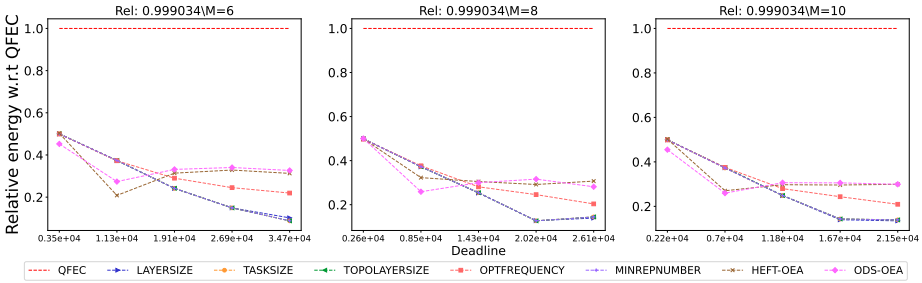

Figure 2651: Assessing the performance of different processor number on the Cholesky workflow.

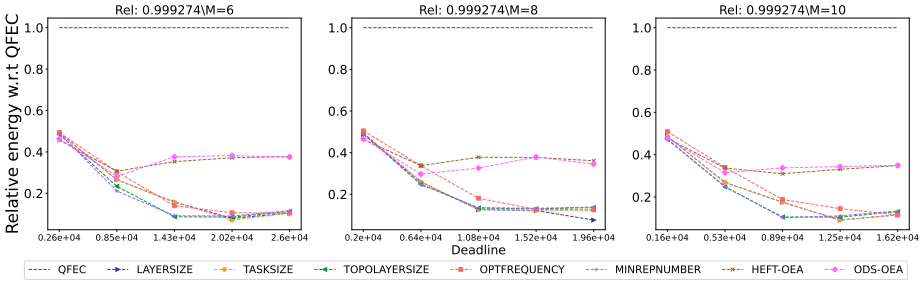

Figure 2652: Assessing the performance of different processor number on the Cycles workflow.

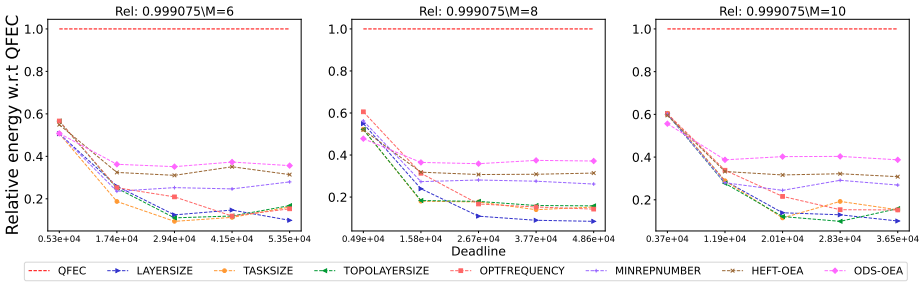

Figure 2653: Assessing the performance of different processor number on the Epigenomics workflow.

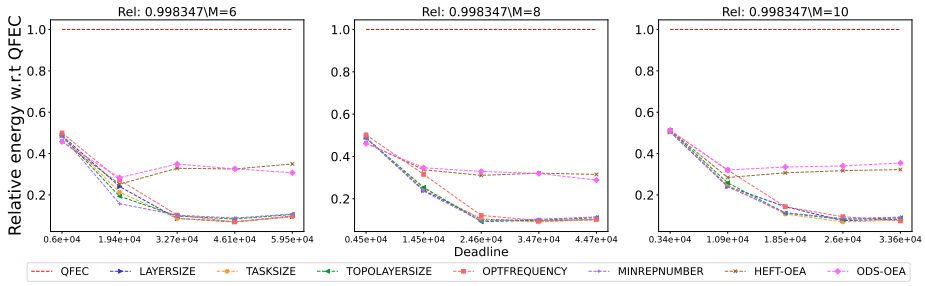

Figure 2654: Assessing the performance of different processor number on the Genome workflow.

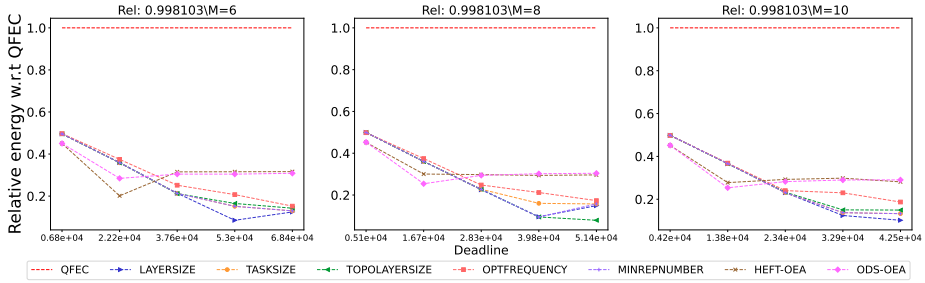

Figure 2655: Assessing the performance of different processor number on the LU.

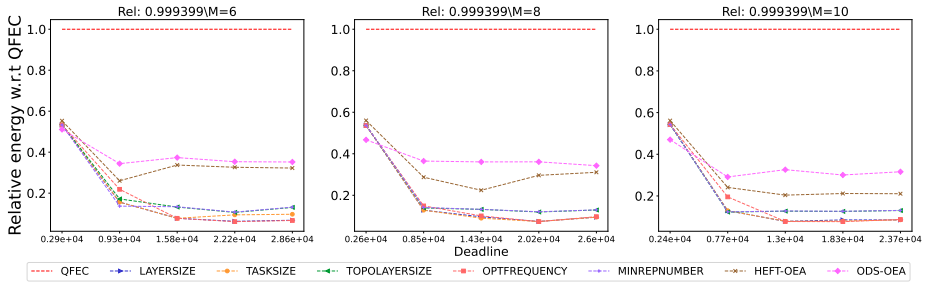

Figure 2656: Assessing the performance of different processor number on the Montage workflow.

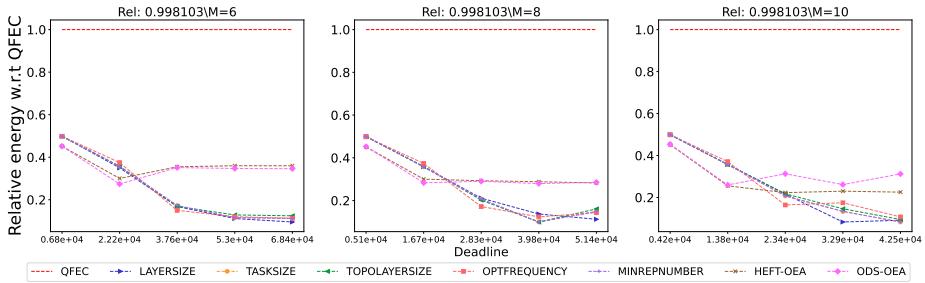

Figure 2657: Assessing the performance of different processor number on the QR.

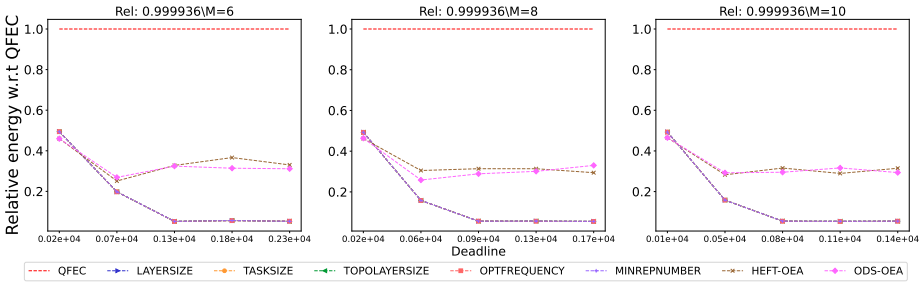

Figure 2658: Assessing the performance of different processor number on the Seismology workflow.

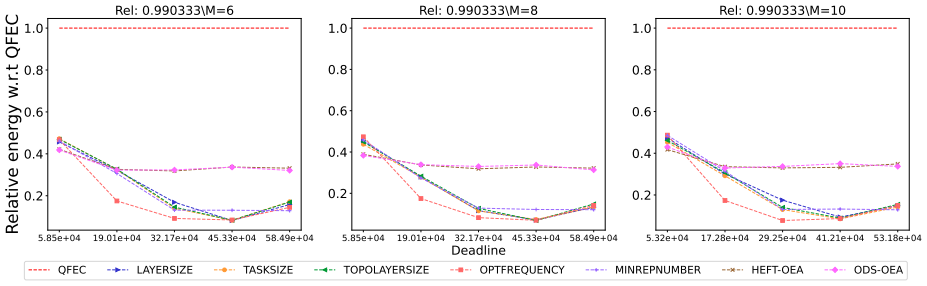

Figure 2659: Assessing the performance of different processor number on the SoyKB workflow.

I.1.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

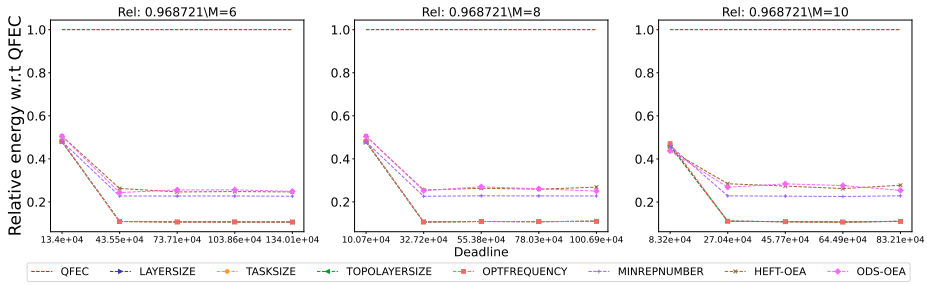

Figure 2660: Assessing the performance of different processor number on the BLAST workflow.

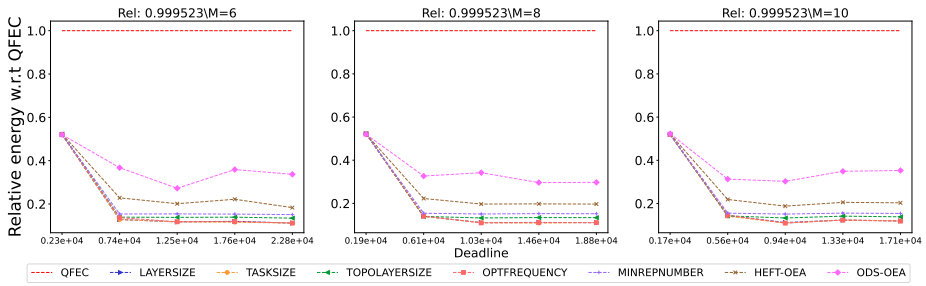

Figure 2661: Assessing the performance of different processor number on the BWA workflow.

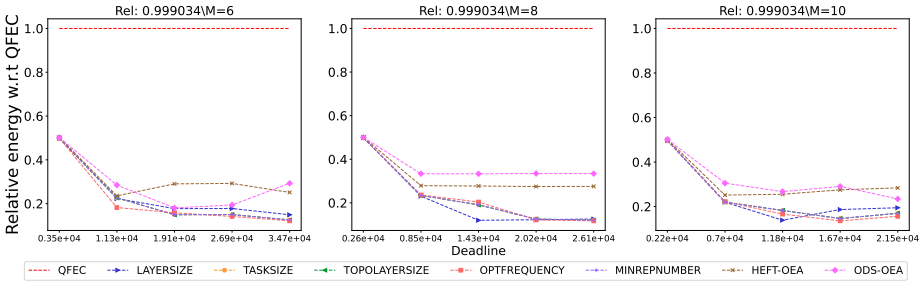

Figure 2662: Assessing the performance of different processor number on the Cholesky workflow.

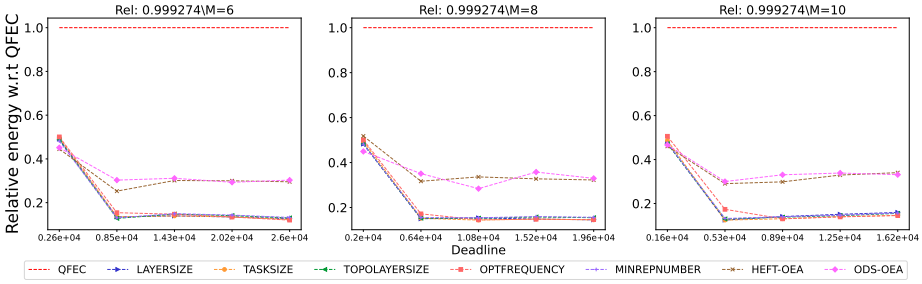

Figure 2663: Assessing the performance of different processor number on the Cycles workflow.

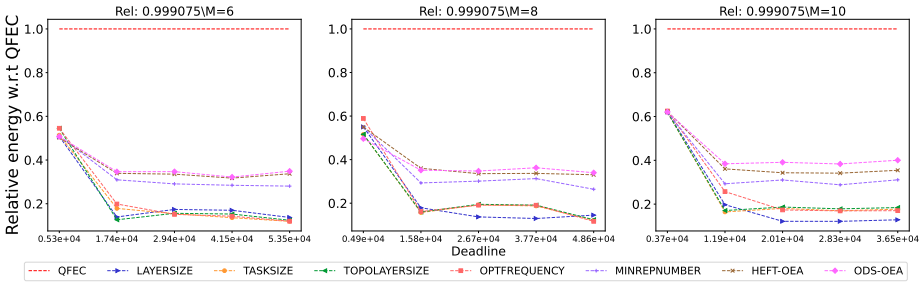

Figure 2664: Assessing the performance of different processor number on the Epigenomics workflow.

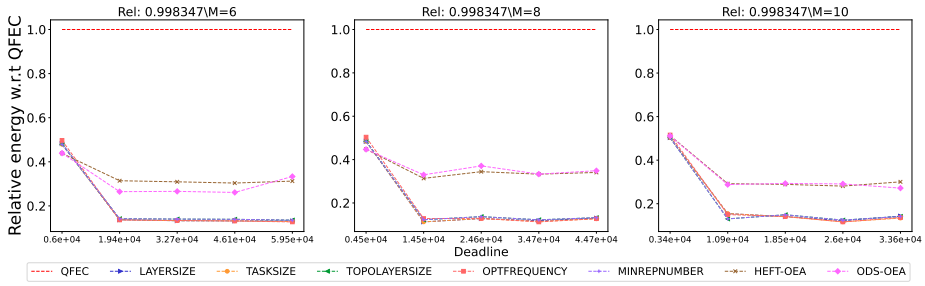

Figure 2665: Assessing the performance of different processor number on the Genome workflow.

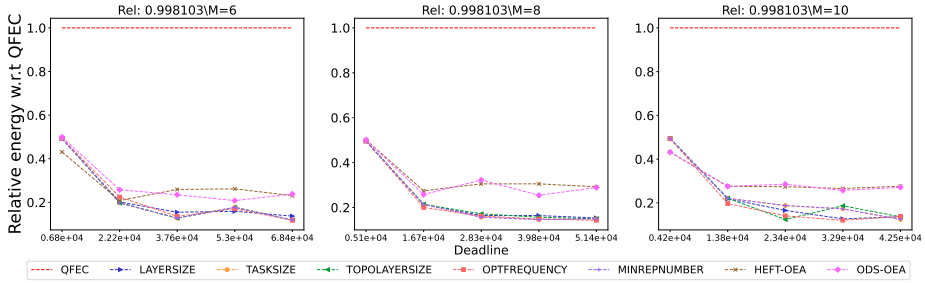

Figure 2666: Assessing the performance of different processor number on the LU.

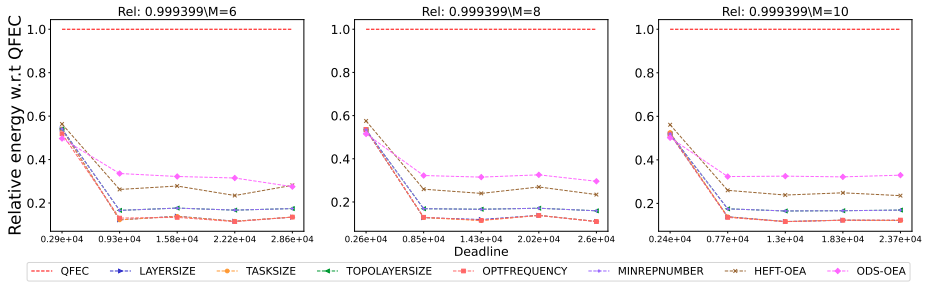

Figure 2667: Assessing the performance of different processor number on the Montage workflow.

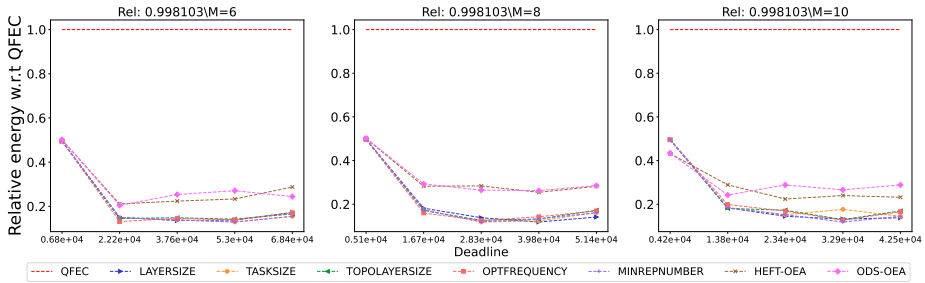

Figure 2668: Assessing the performance of different processor number on the QR.

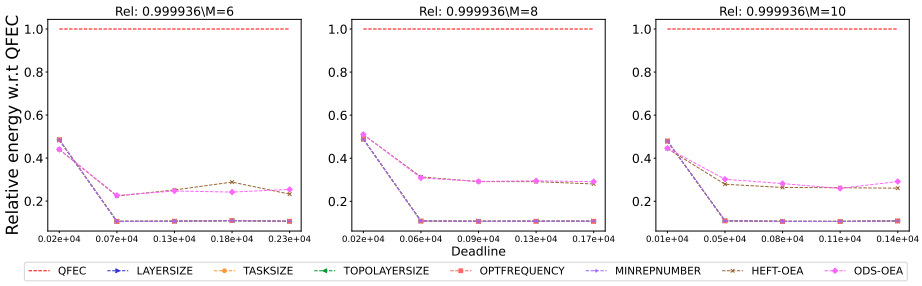

Figure 2669: Assessing the performance of different processor number on the Seismology workflow.

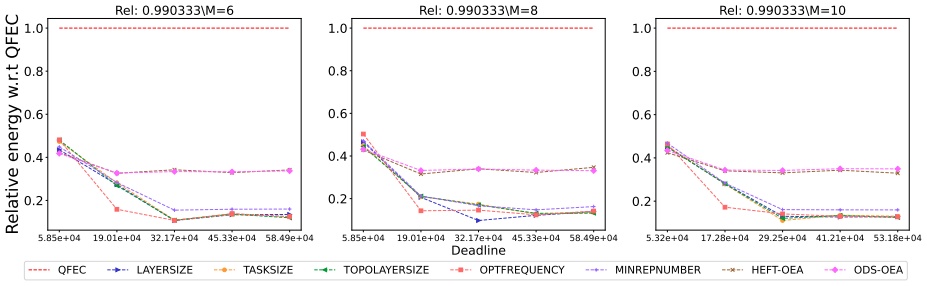

Figure 2670: Assessing the performance of different processor number on the SoyKB workflow.

I.1.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

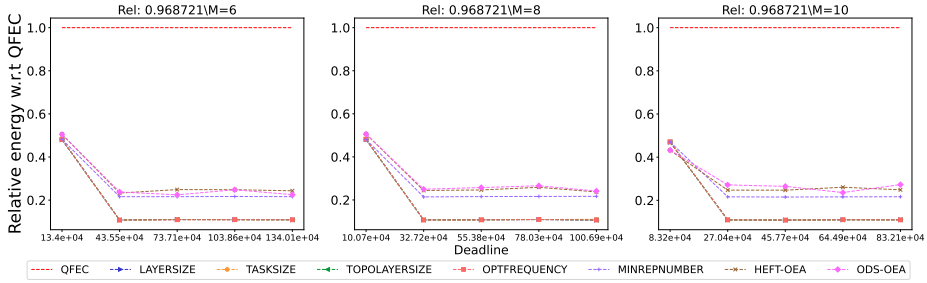

Figure 2671: Assessing the performance of different processor number on the BLAST workflow.

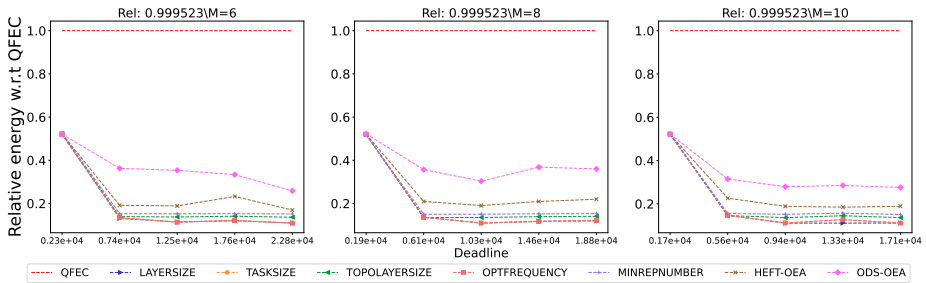

Figure 2672: Assessing the performance of different processor number on the BWA workflow.

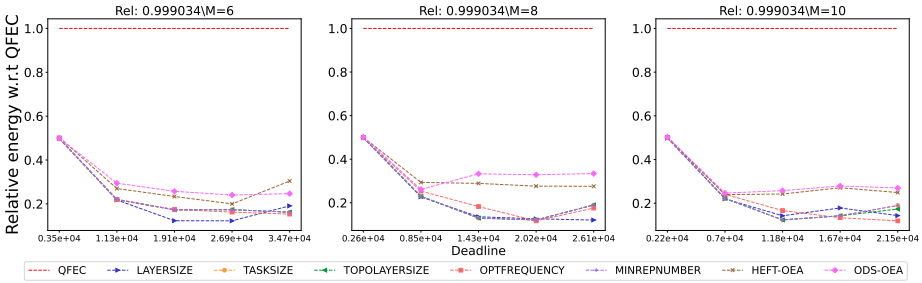

Figure 2673: Assessing the performance of different processor number on the Cholesky workflow.

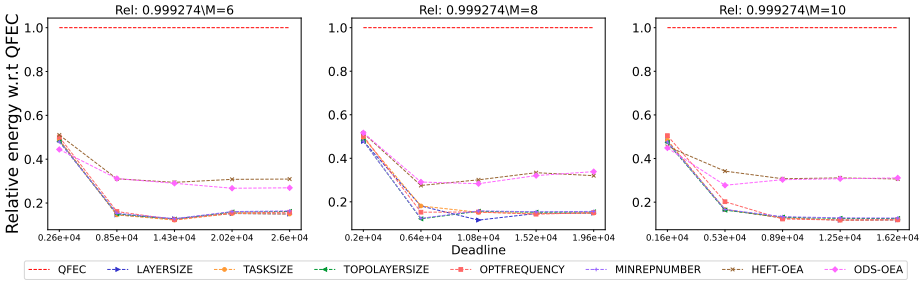

Figure 2674: Assessing the performance of different processor number on the Cycles workflow.

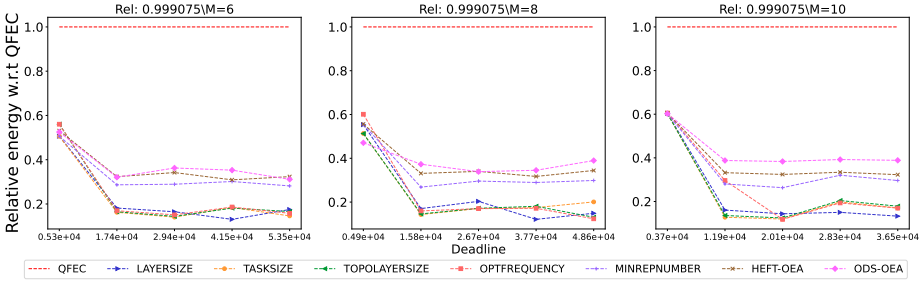

Figure 2675: Assessing the performance of different processor number on the Epigenomics workflow.

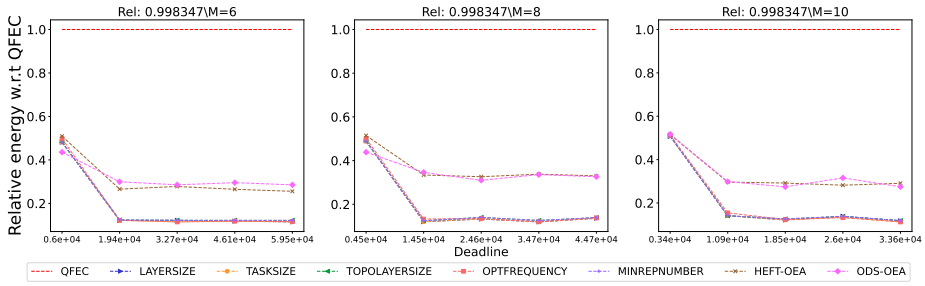

Figure 2676: Assessing the performance of different processor number on the Genome workflow.

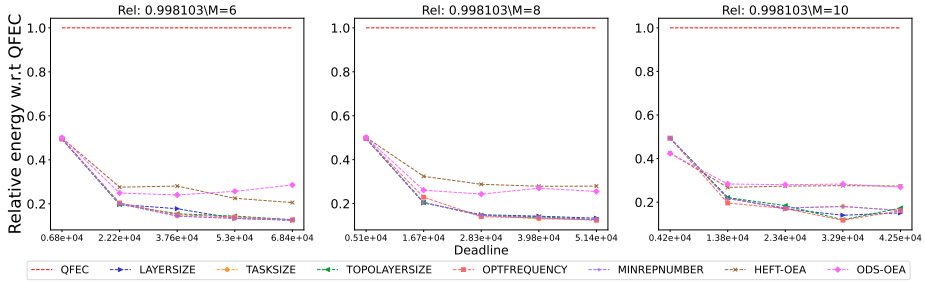

Figure 2677: Assessing the performance of different processor number on the LU.

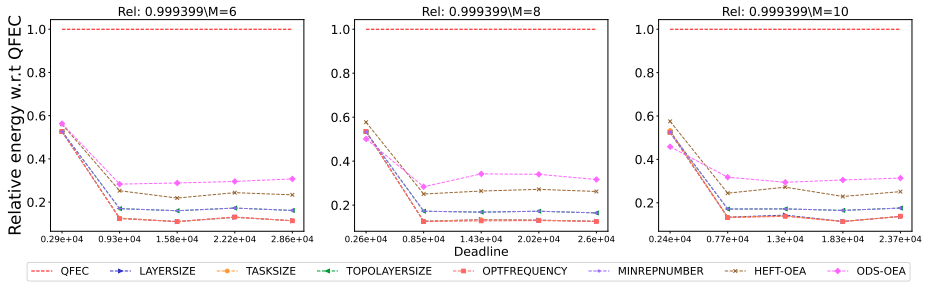

Figure 2678: Assessing the performance of different processor number on the Montage workflow.

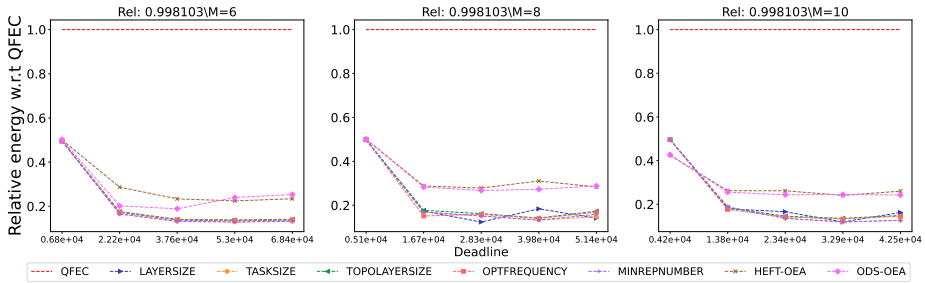

Figure 2679: Assessing the performance of different processor number on the QR.

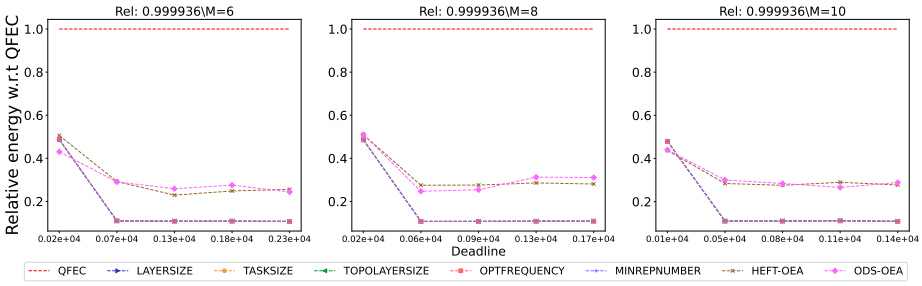

Figure 2680: Assessing the performance of different processor number on the Seismology workflow.

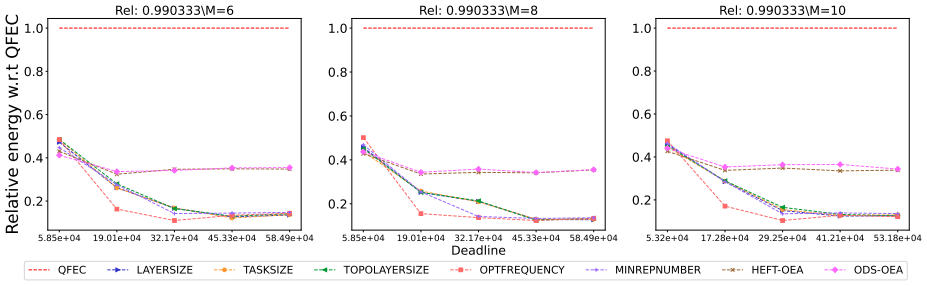

Figure 2681: Assessing the performance of different processor number on the SoyKB workflow.

I.2 $BC/WC = 0.2$

I.2.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

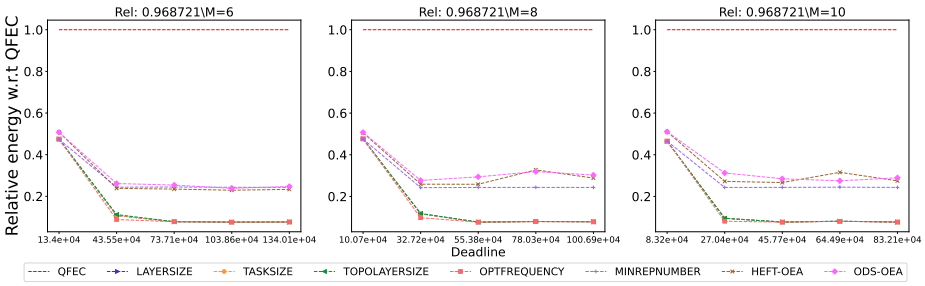

Figure 2682: Assessing the performance of different processor number on the BLAST workflow.

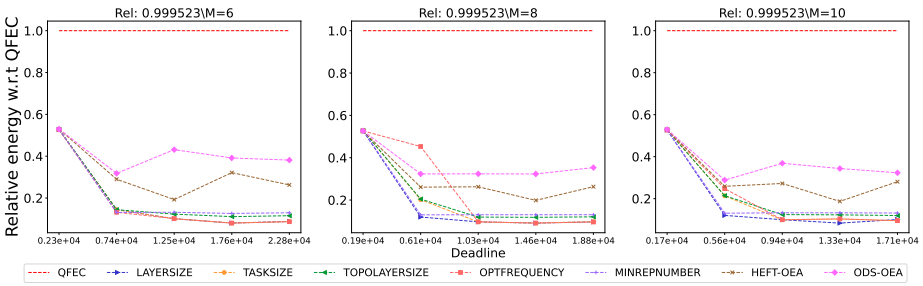

Figure 2683: Assessing the performance of different processor number on the BWA workflow.

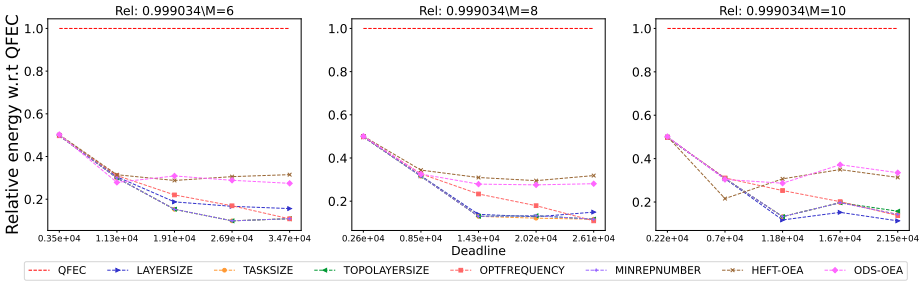

Figure 2684: Assessing the performance of different processor number on the Cholesky workflow.

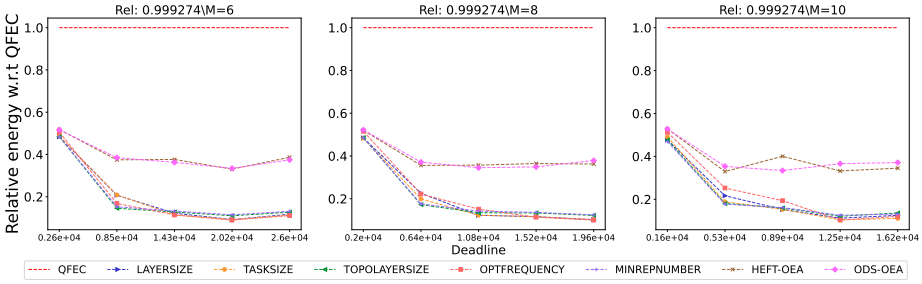

Figure 2685: Assessing the performance of different processor number on the Cycles workflow.

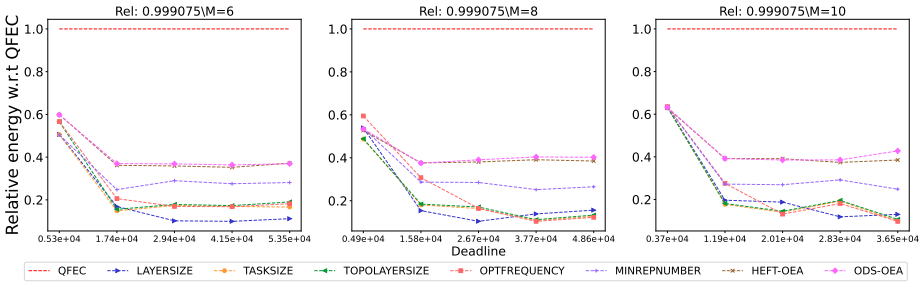

Figure 2686: Assessing the performance of different processor number on the Epigenomics workflow.

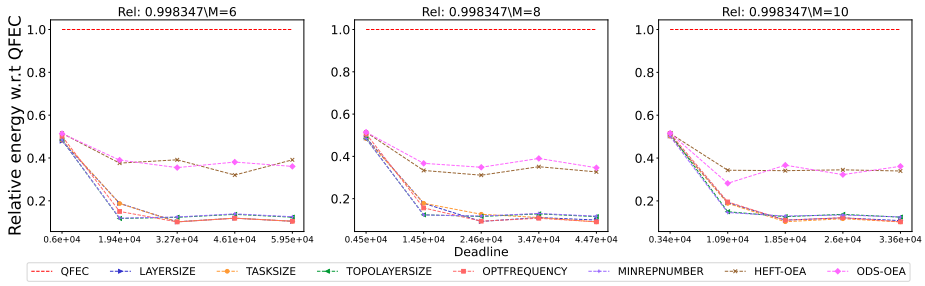

Figure 2687: Assessing the performance of different processor number on the Genome workflow.

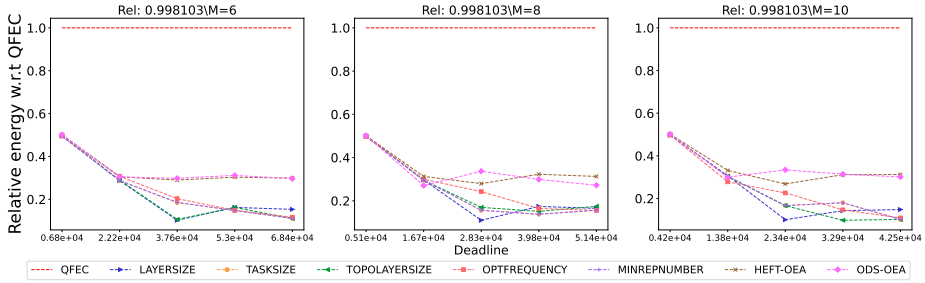

Figure 2688: Assessing the performance of different processor number on the LU.

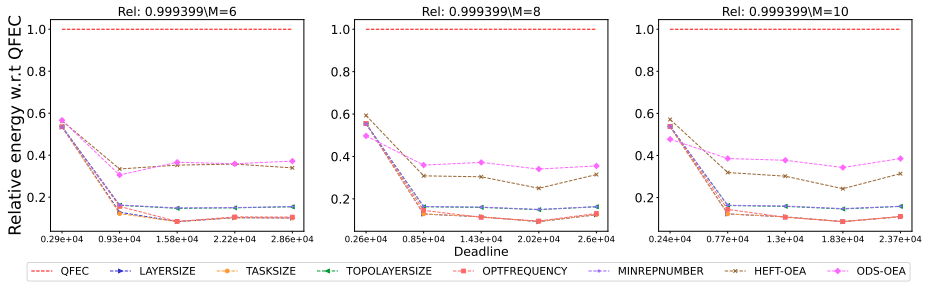

Figure 2689: Assessing the performance of different processor number on the Montage workflow.

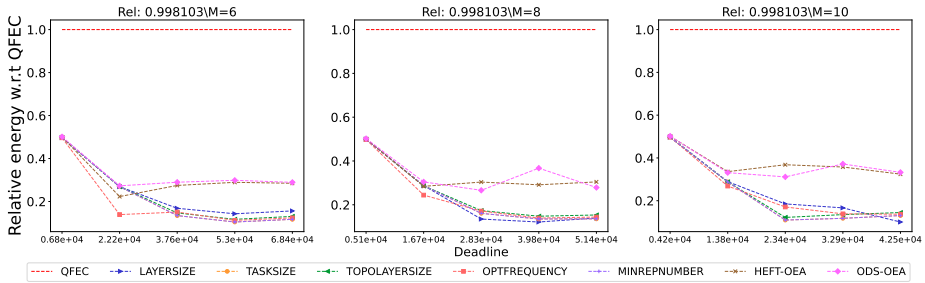

Figure 2690: Assessing the performance of different processor number on the QR.

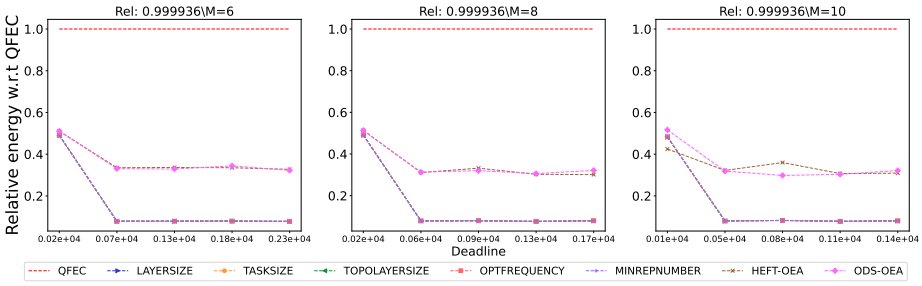

Figure 2691: Assessing the performance of different processor number on the Seismology workflow.

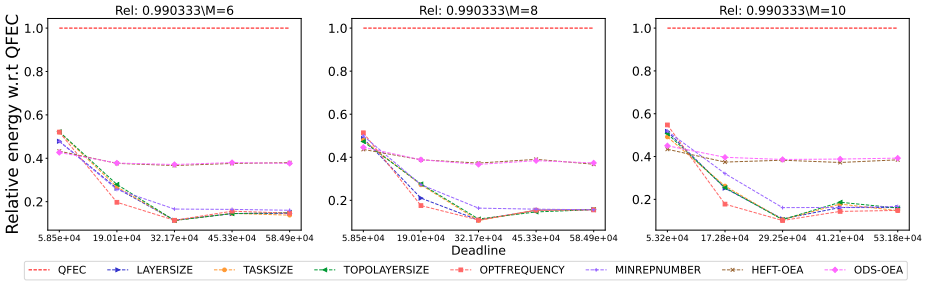

Figure 2692: Assessing the performance of different processor number on the SoyKB workflow.

I.2.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

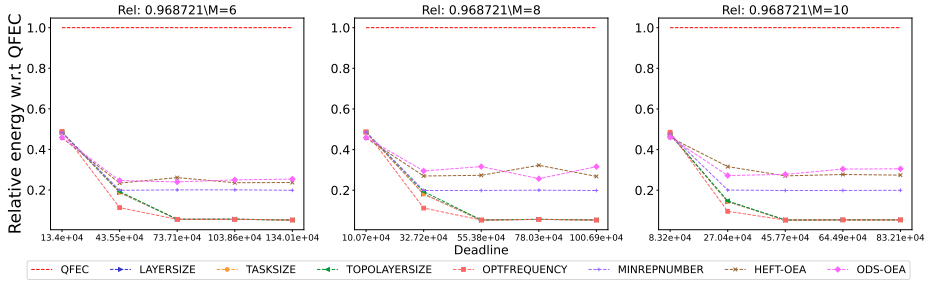

Figure 2693: Assessing the performance of different processor number on the BLAST workflow.

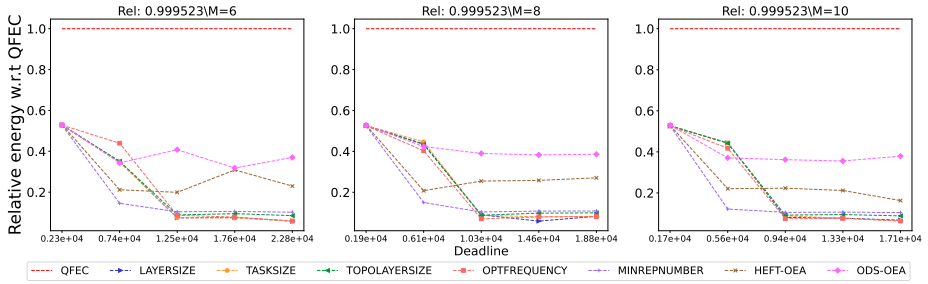

Figure 2694: Assessing the performance of different processor number on the BWA workflow.

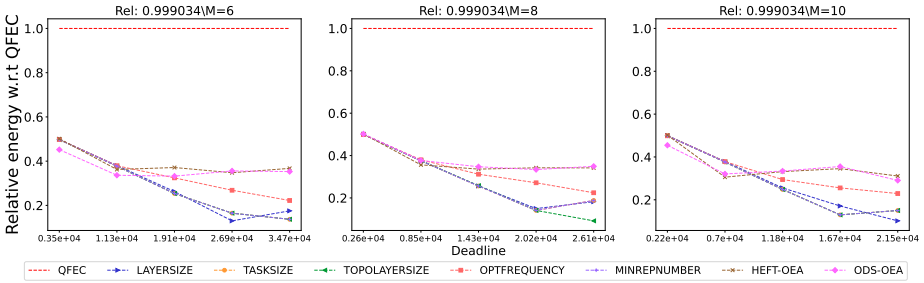

Figure 2695: Assessing the performance of different processor number on the Cholesky workflow.

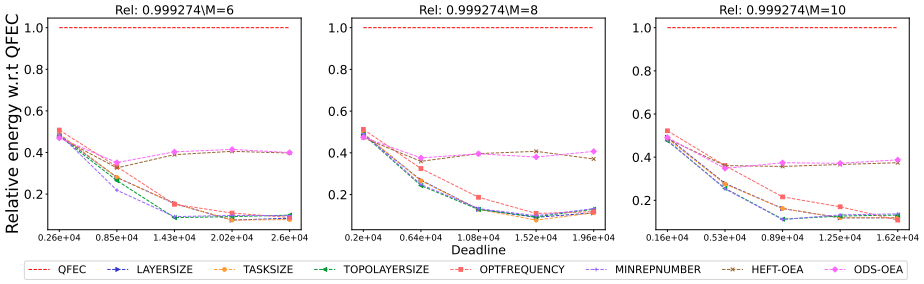

Figure 2696: Assessing the performance of different processor number on the Cycles workflow.

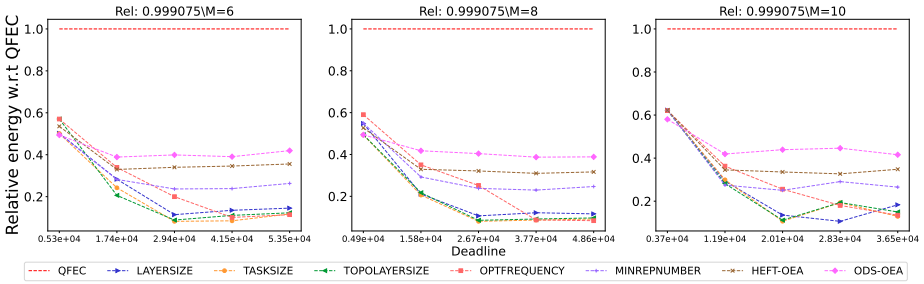

Figure 2697: Assessing the performance of different processor number on the Epigenomics workflow.

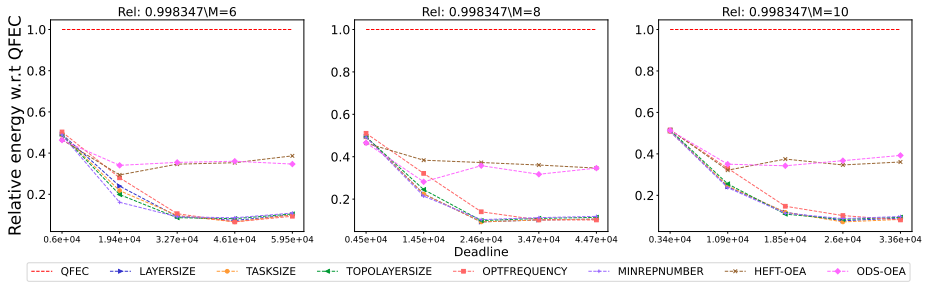

Figure 2698: Assessing the performance of different processor number on the Genome workflow.

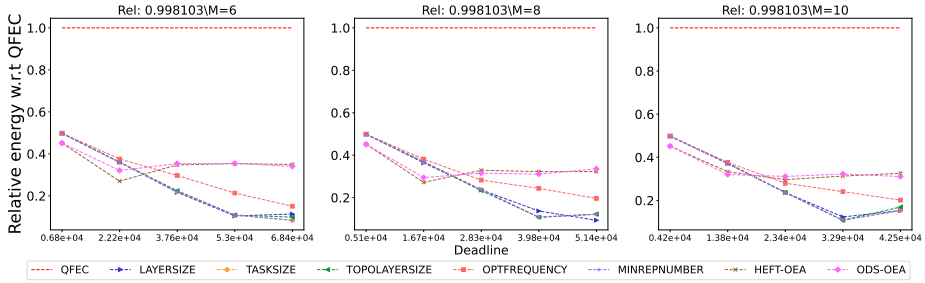

Figure 2699: Assessing the performance of different processor number on the LU.

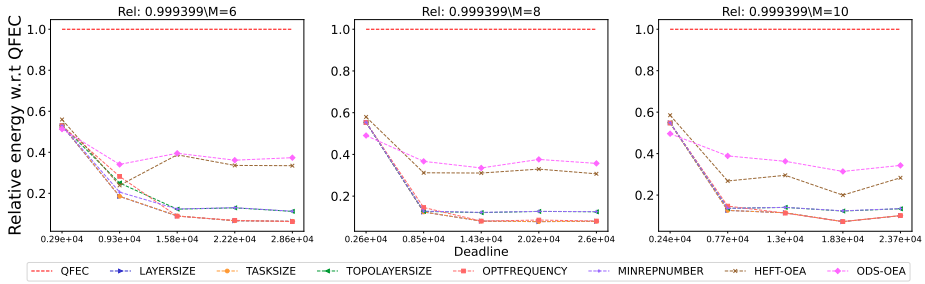

Figure 2700: Assessing the performance of different processor number on the Montage workflow.

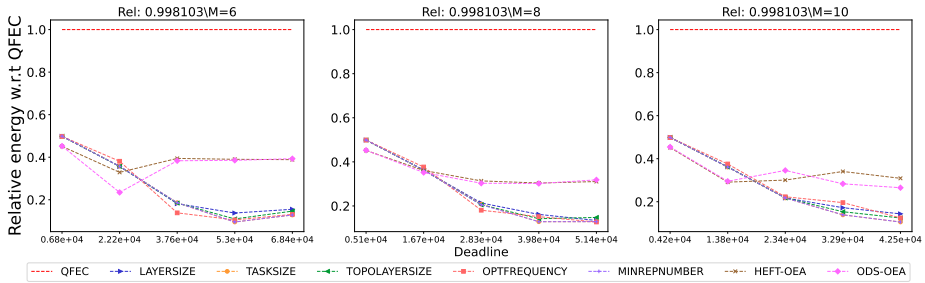

Figure 2701: Assessing the performance of different processor number on the QR.

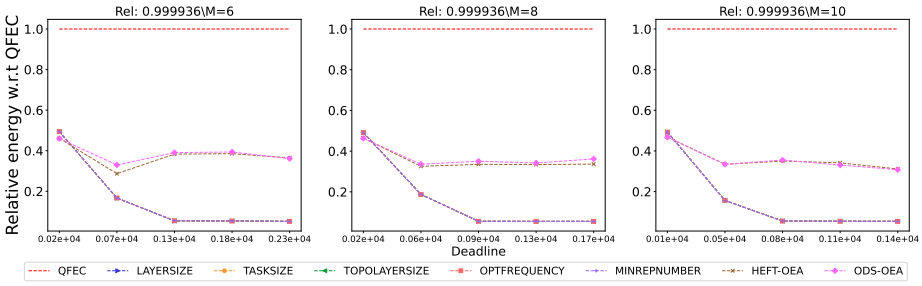

Figure 2702: Assessing the performance of different processor number on the Seismology workflow.

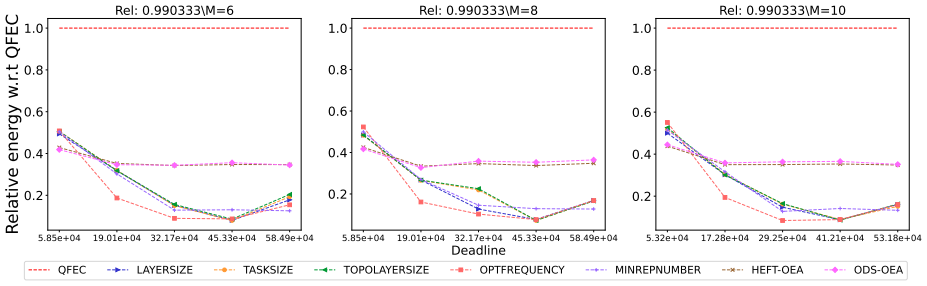

Figure 2703: Assessing the performance of different processor number on the SoyKB workflow.

I.2.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

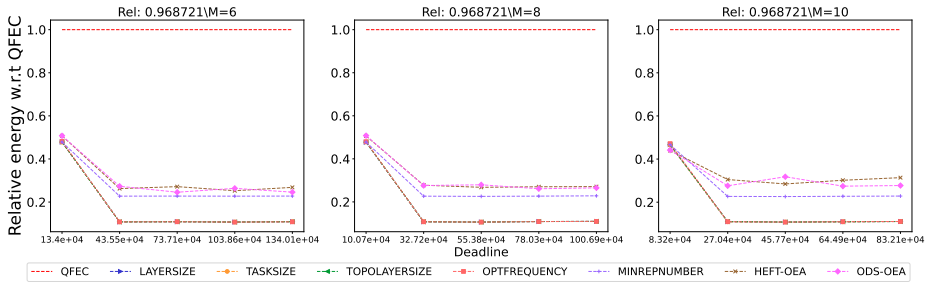

Figure 2704: Assessing the performance of different processor number on the BLAST workflow.

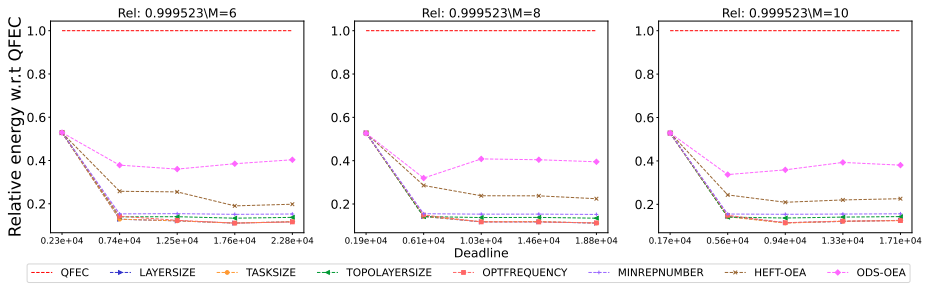

Figure 2705: Assessing the performance of different processor number on the BWA workflow.

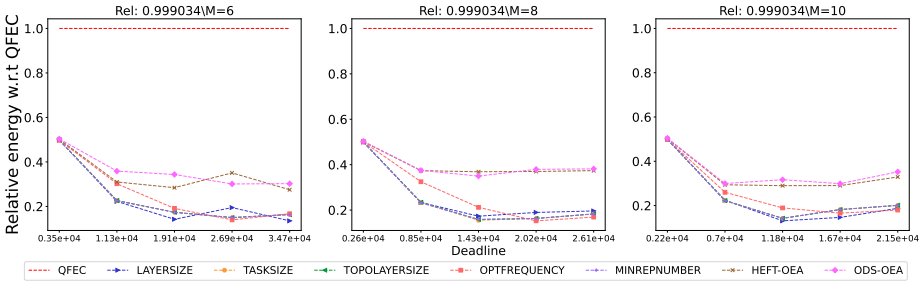

Figure 2706: Assessing the performance of different processor number on the Cholesky workflow.

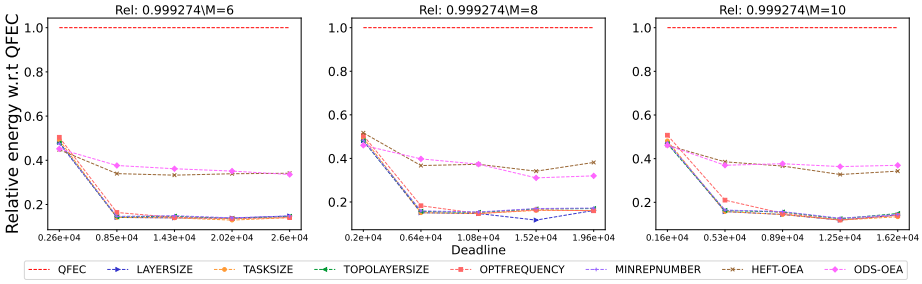

Figure 2707: Assessing the performance of different processor number on the Cycles workflow.

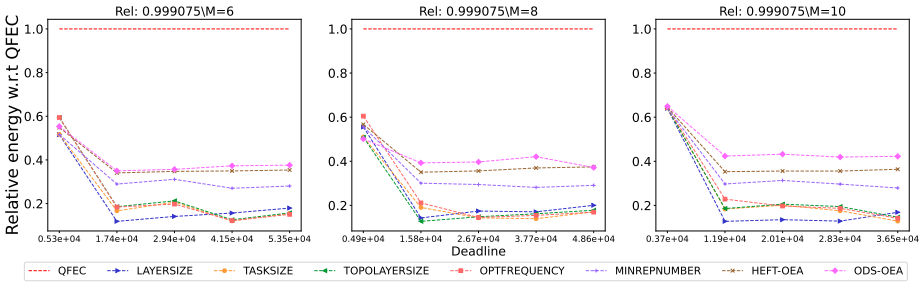

Figure 2708: Assessing the performance of different processor number on the Epigenomics workflow.

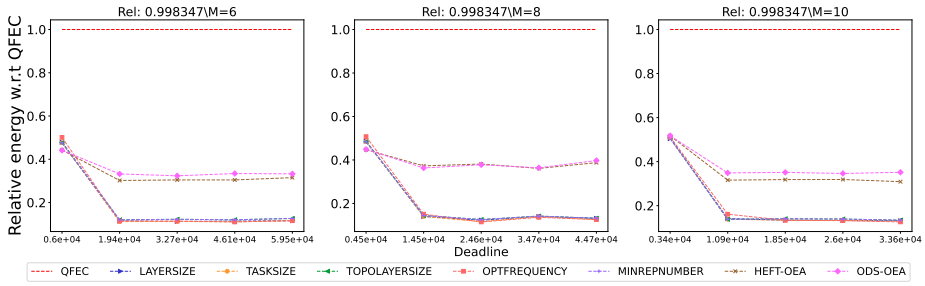

Figure 2709: Assessing the performance of different processor number on the Genome workflow.

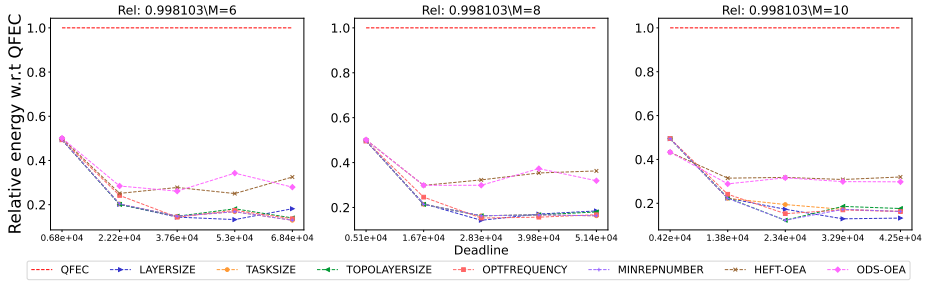

Figure 2710: Assessing the performance of different processor number on the LU.

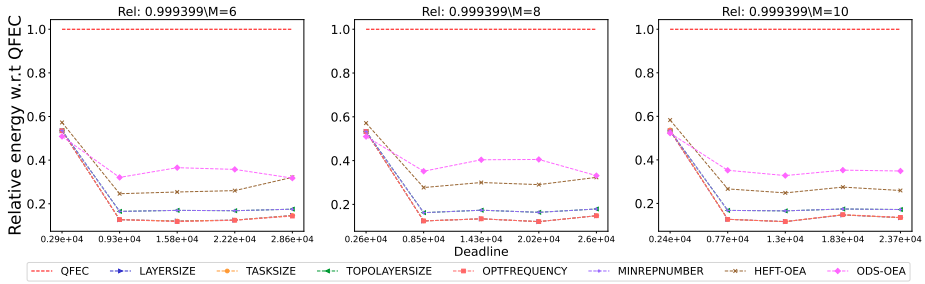

Figure 2711: Assessing the performance of different processor number on the Montage workflow.

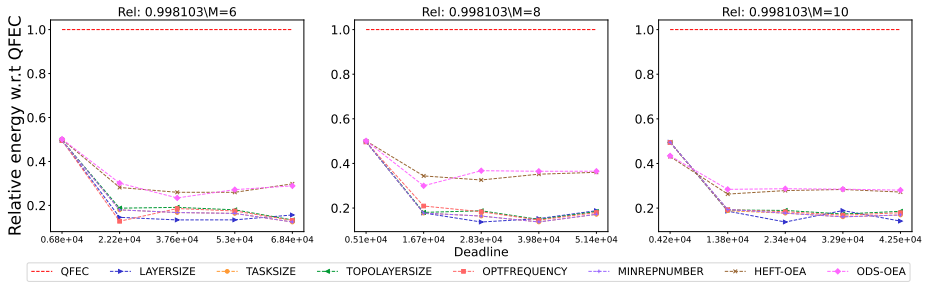

Figure 2712: Assessing the performance of different processor number on the QR.

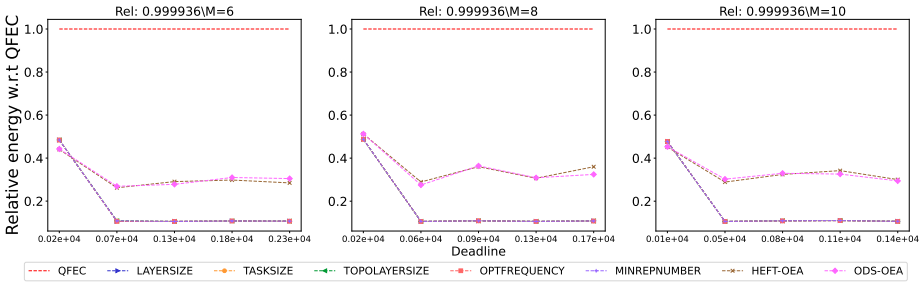

Figure 2713: Assessing the performance of different processor number on the Seismology workflow.

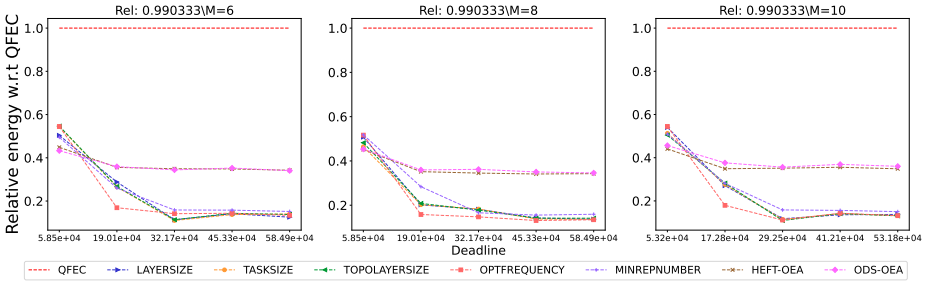

Figure 2714: Assessing the performance of different processor number on the SoyKB workflow.

I.2.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

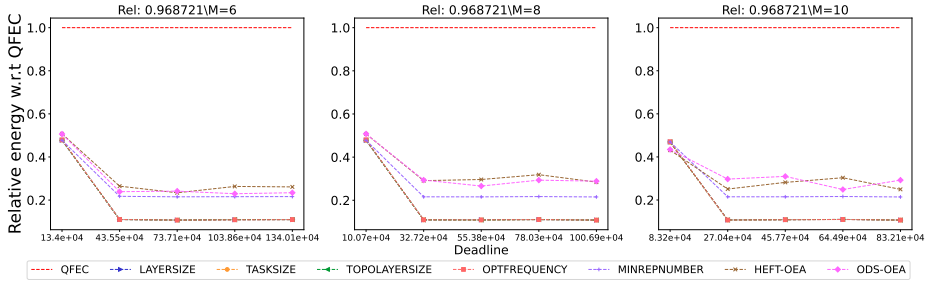

Figure 2715: Assessing the performance of different processor number on the BLAST workflow.

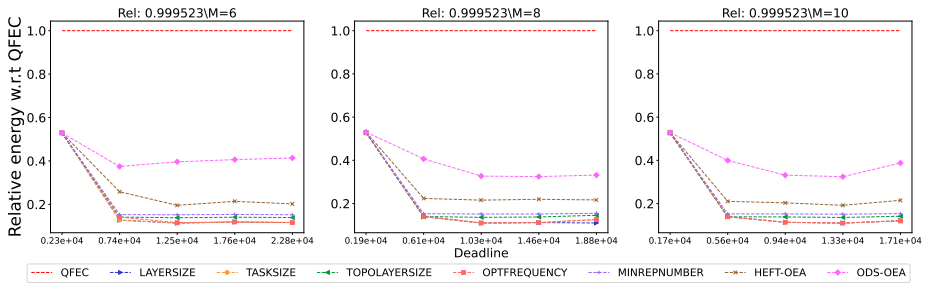

Figure 2716: Assessing the performance of different processor number on the BWA workflow.

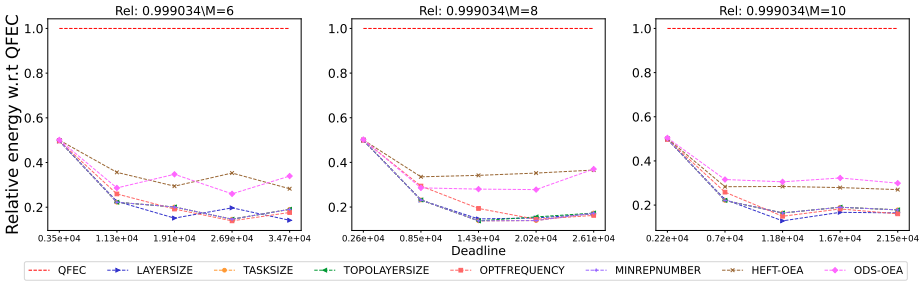

Figure 2717: Assessing the performance of different processor number on the Cholesky workflow.

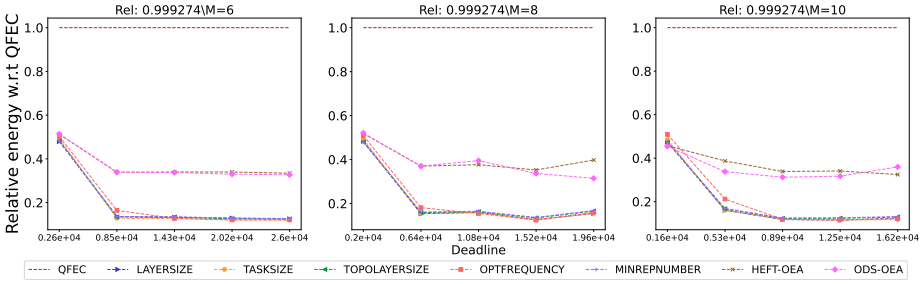

Figure 2718: Assessing the performance of different processor number on the Cycles workflow.

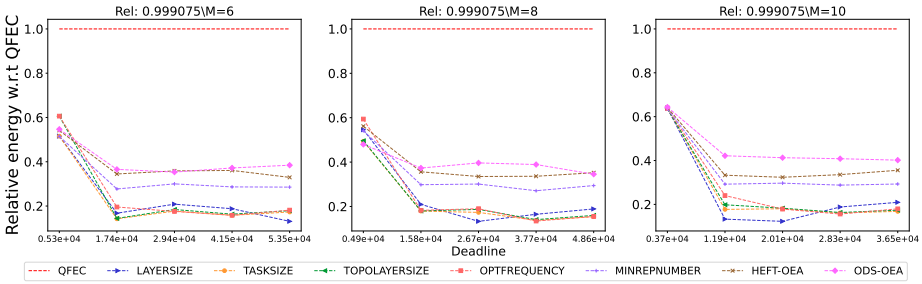

Figure 2719: Assessing the performance of different processor number on the Epigenomics workflow.

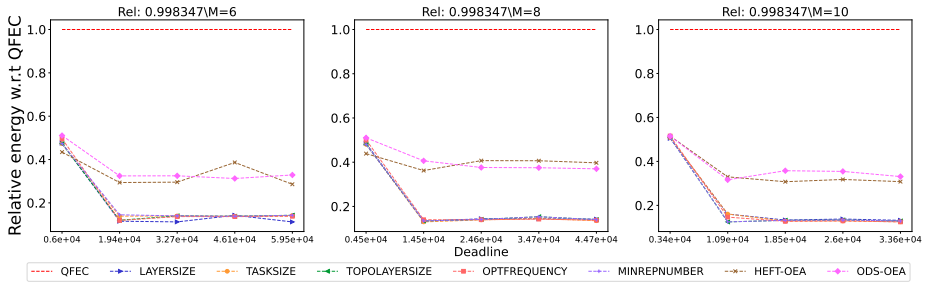

Figure 2720: Assessing the performance of different processor number on the Genome workflow.

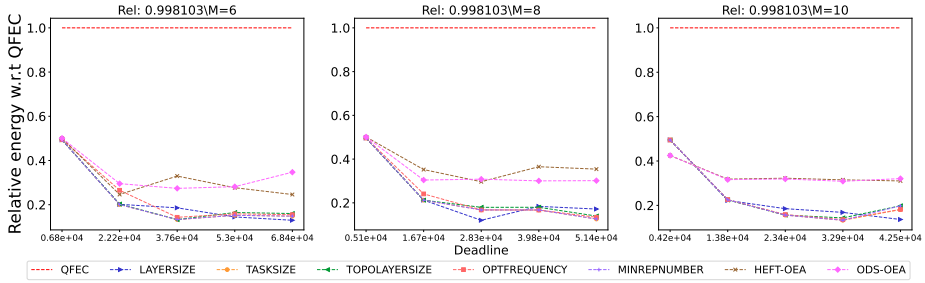

Figure 2721: Assessing the performance of different processor number on the LU.

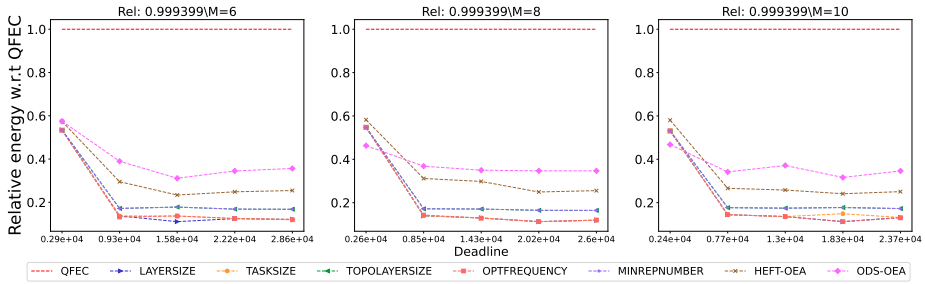

Figure 2722: Assessing the performance of different processor number on the Montage workflow.

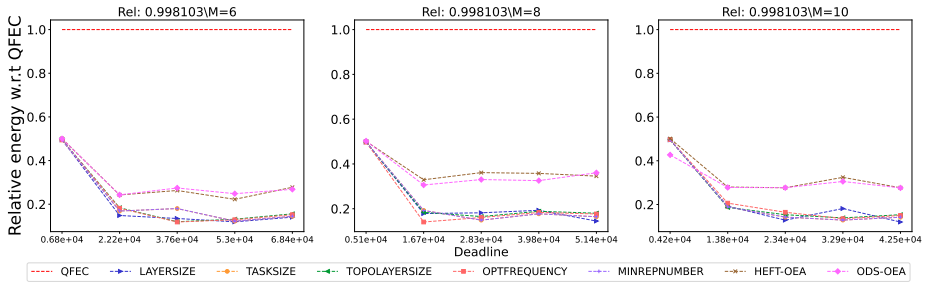

Figure 2723: Assessing the performance of different processor number on the QR.

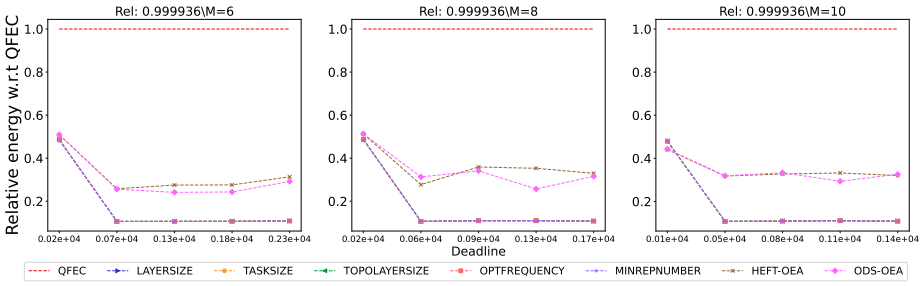

Figure 2724: Assessing the performance of different processor number on the Seismology workflow.

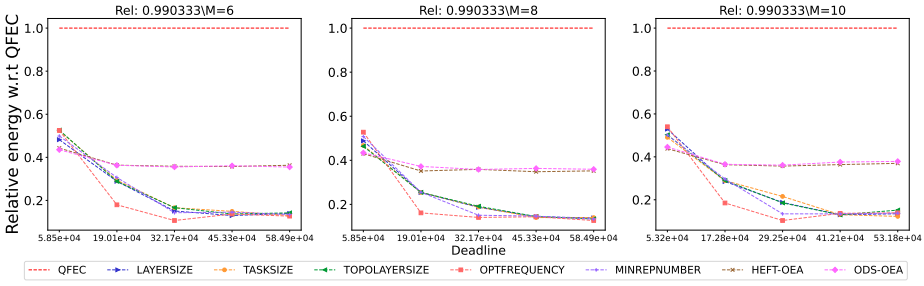

Figure 2725: Assessing the performance of different processor number on the SoyKB workflow.

I.3 $BC/WC = 0.3$

I.3.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

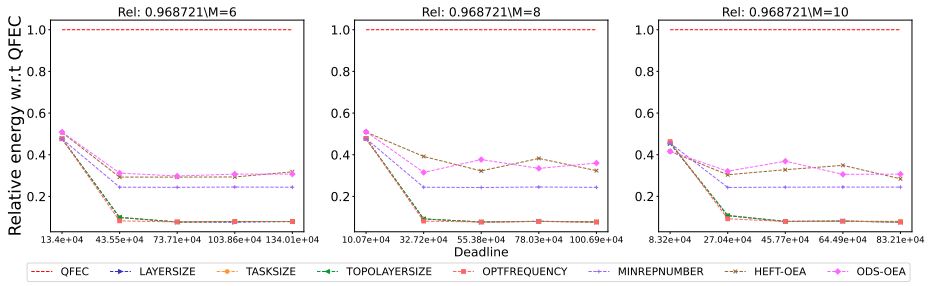

Figure 2726: Assessing the performance of different processor number on the BLAST workflow.

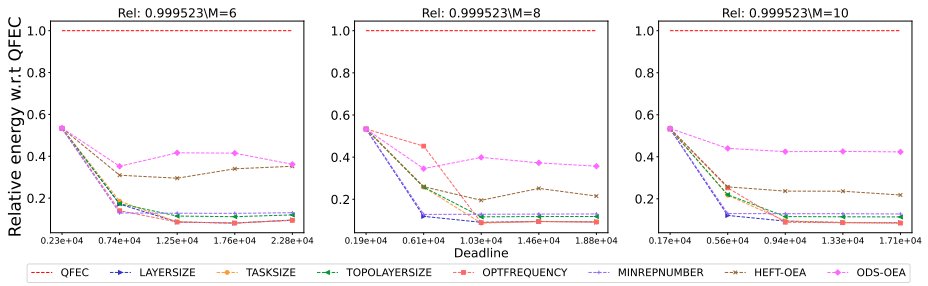

Figure 2727: Assessing the performance of different processor number on the BWA workflow.

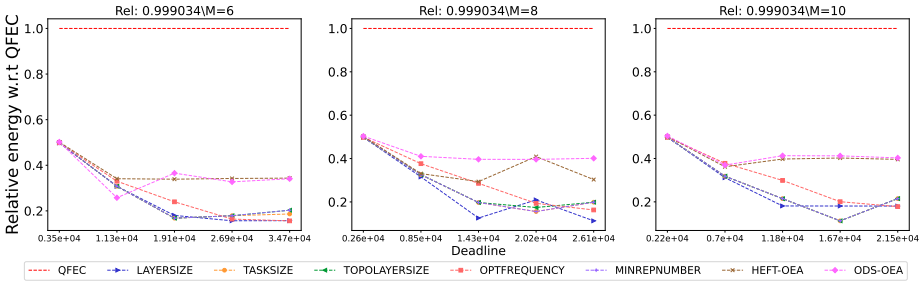

Figure 2728: Assessing the performance of different processor number on the Cholesky workflow.

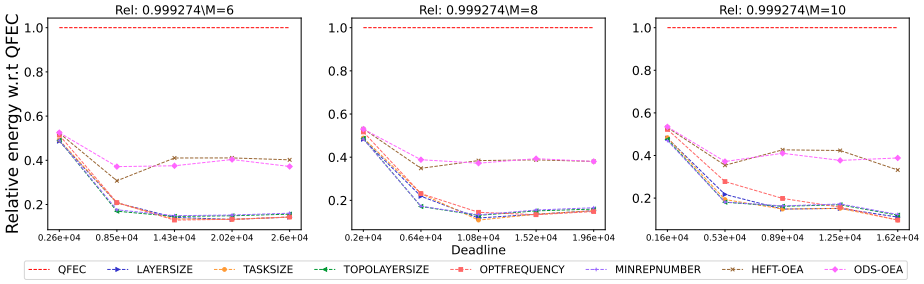

Figure 2729: Assessing the performance of different processor number on the Cycles workflow.

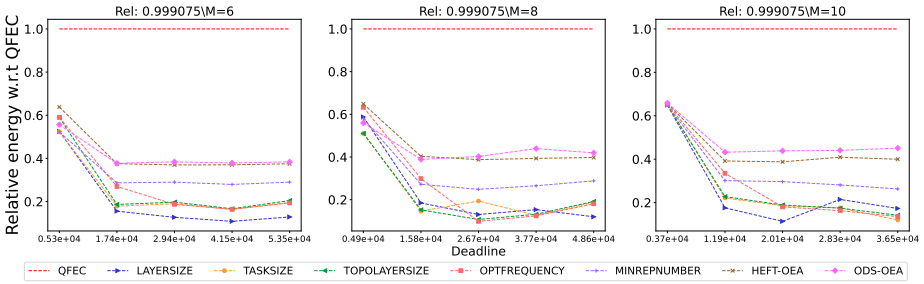

Figure 2730: Assessing the performance of different processor number on the Epigenomics workflow.

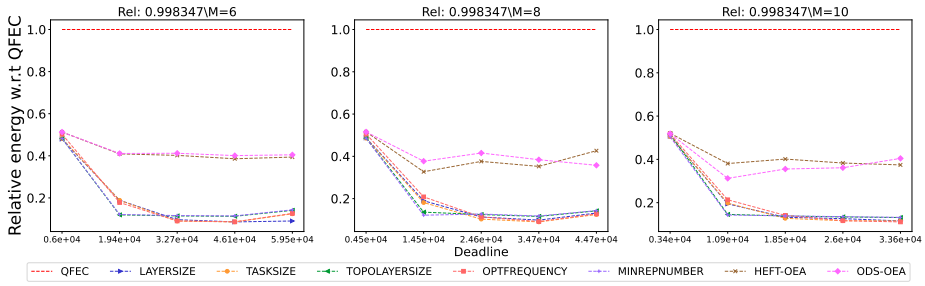

Figure 2731: Assessing the performance of different processor number on the Genome workflow.

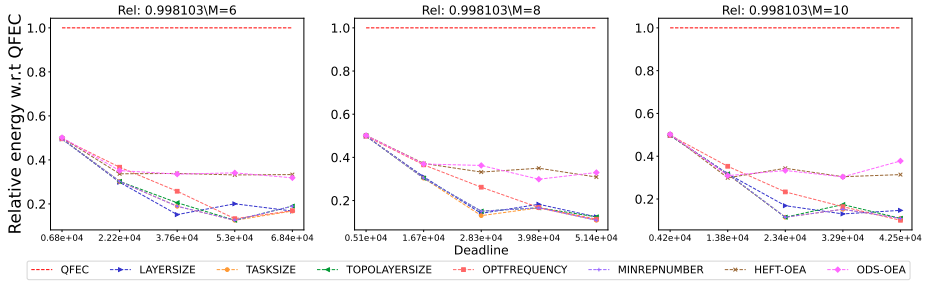

Figure 2732: Assessing the performance of different processor number on the LU.

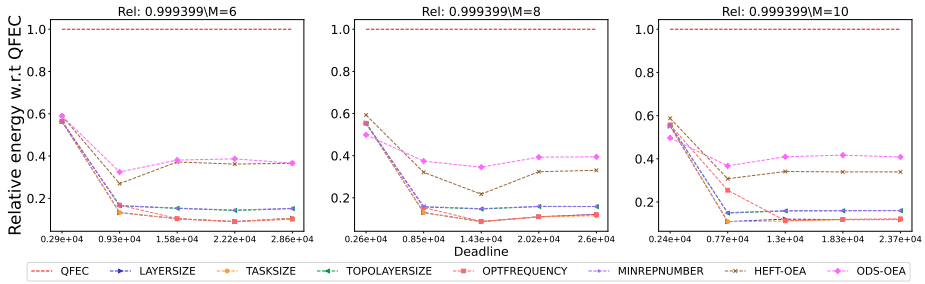

Figure 2733: Assessing the performance of different processor number on the Montage workflow.

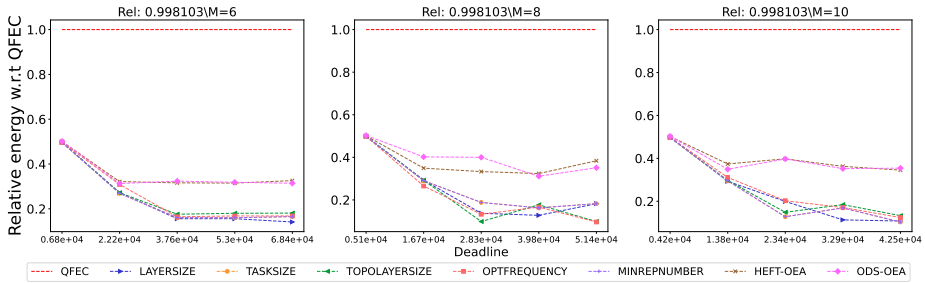

Figure 2734: Assessing the performance of different processor number on the QR.

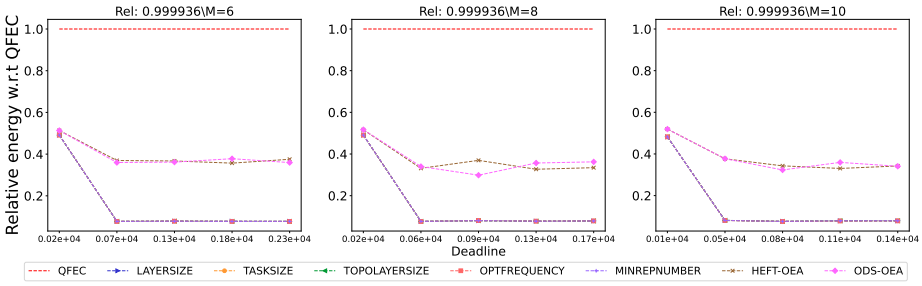

Figure 2735: Assessing the performance of different processor number on the Seismology workflow.

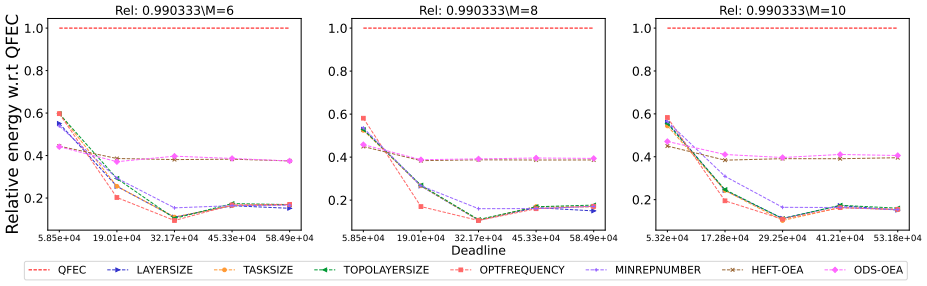

Figure 2736: Assessing the performance of different processor number on the SoyKB workflow.

I.3.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

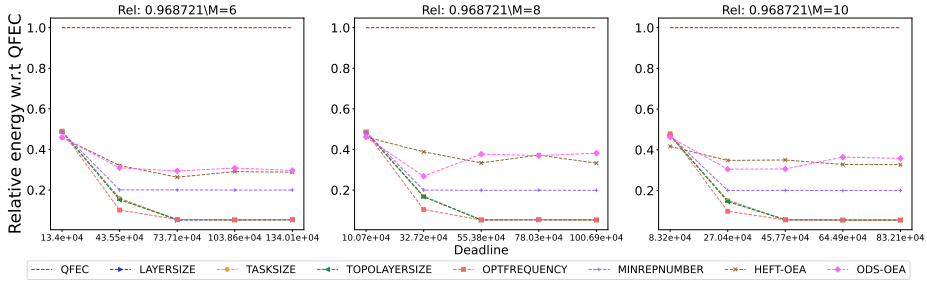

Figure 2737: Assessing the performance of different processor number on the BLAST workflow.

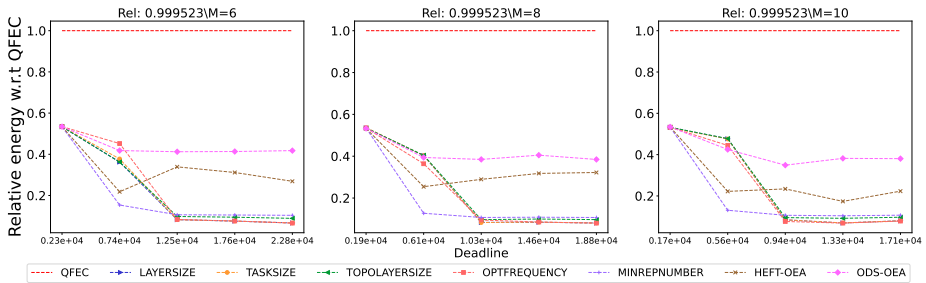

Figure 2738: Assessing the performance of different processor number on the BWA workflow.

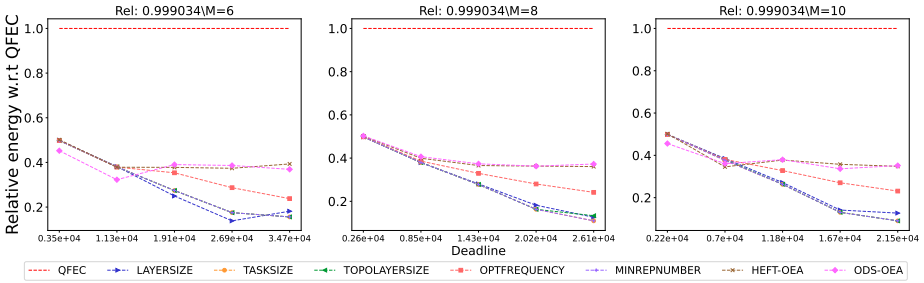

Figure 2739: Assessing the performance of different processor number on the Cholesky workflow.

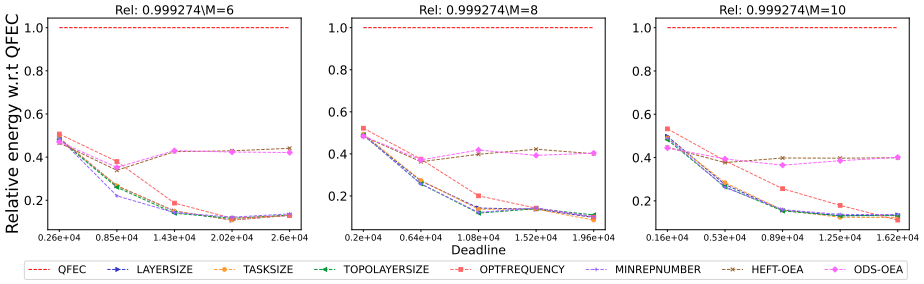

Figure 2740: Assessing the performance of different processor number on the Cycles workflow.

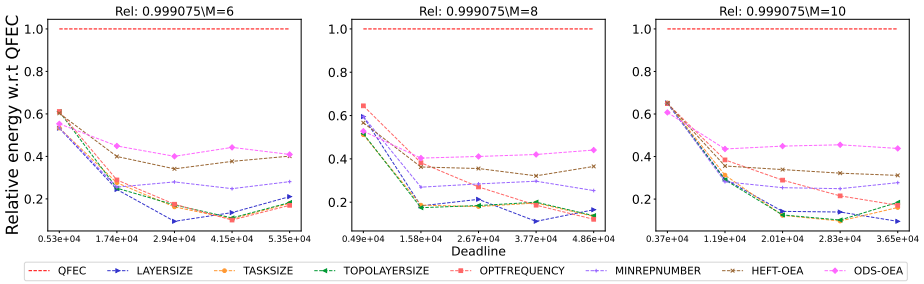

Figure 2741: Assessing the performance of different processor number on the Epigenomics workflow.

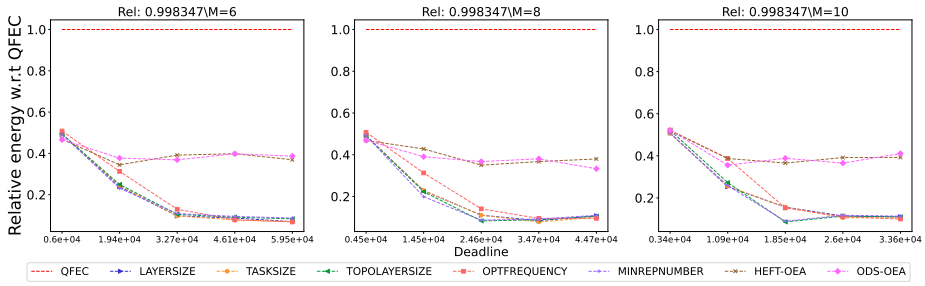

Figure 2742: Assessing the performance of different processor number on the Genome workflow.

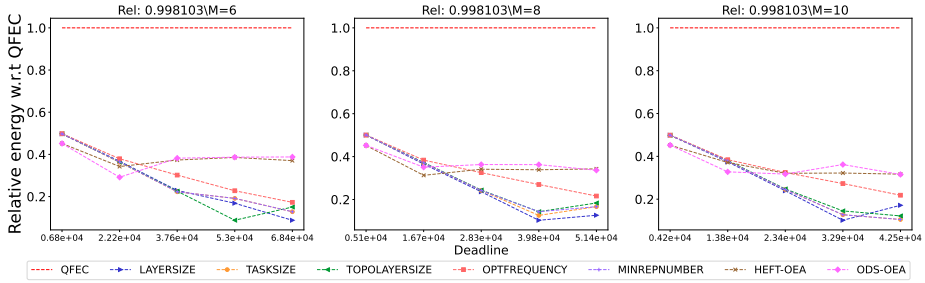

Figure 2743: Assessing the performance of different processor number on the LU.

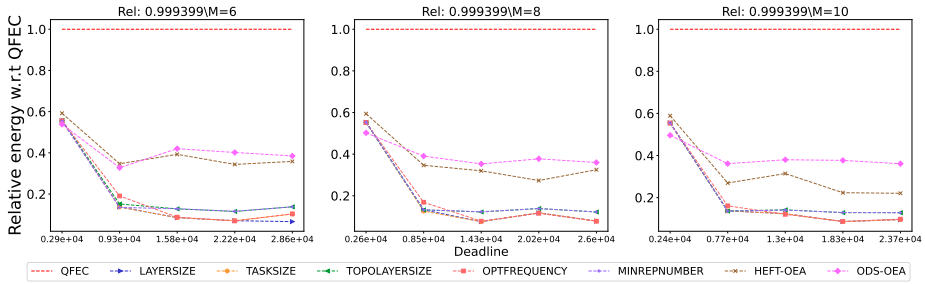

Figure 2744: Assessing the performance of different processor number on the Montage workflow.

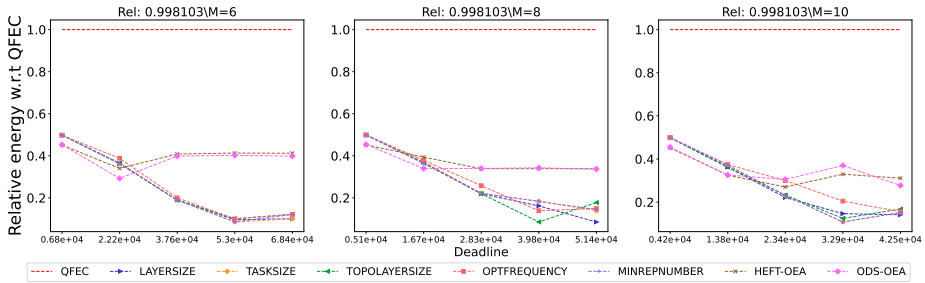

Figure 2745: Assessing the performance of different processor number on the QR.

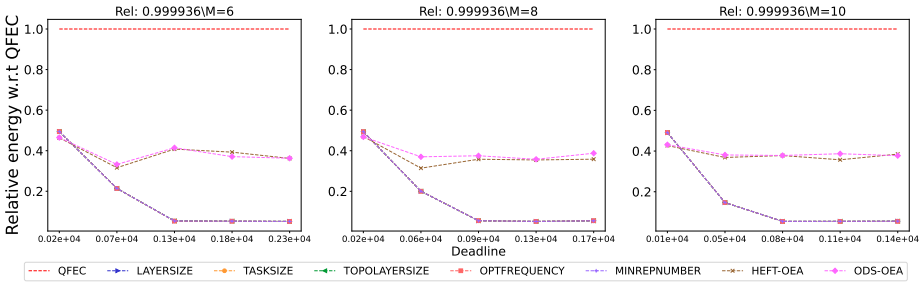

Figure 2746: Assessing the performance of different processor number on the Seismology workflow.

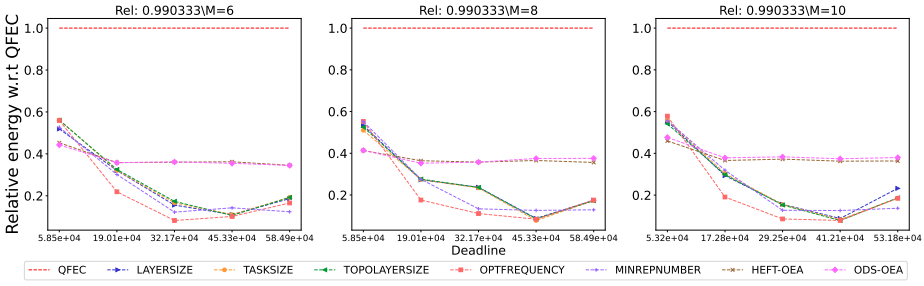

Figure 2747: Assessing the performance of different processor number on the SoyKB workflow.

I.3.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

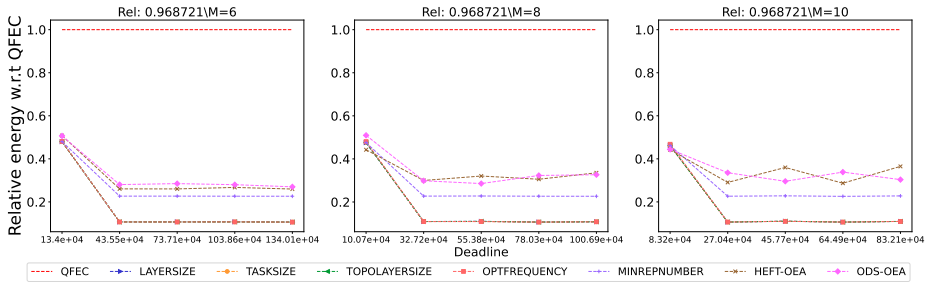

Figure 2748: Assessing the performance of different processor number on the BLAST workflow.

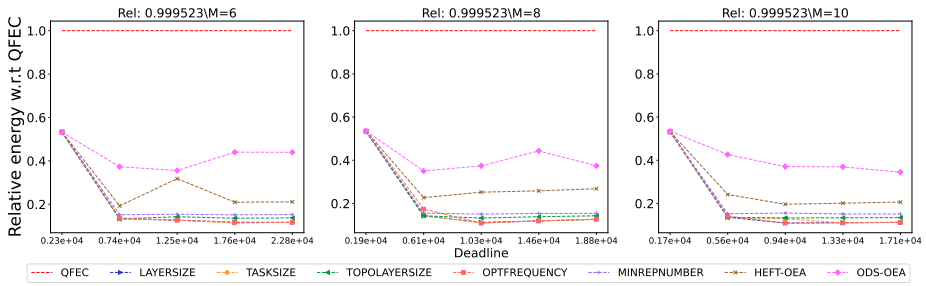

Figure 2749: Assessing the performance of different processor number on the BWA workflow.

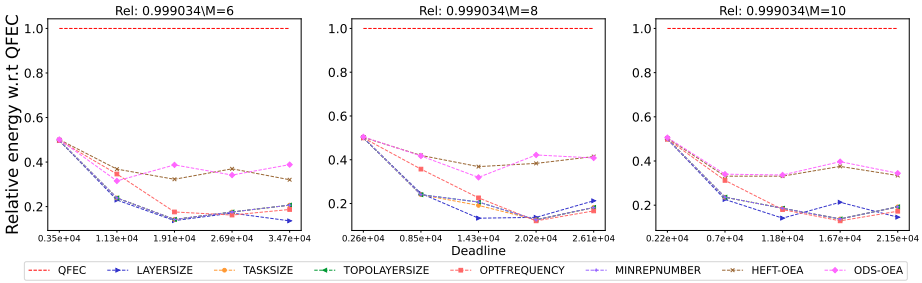

Figure 2750: Assessing the performance of different processor number on the Cholesky workflow.

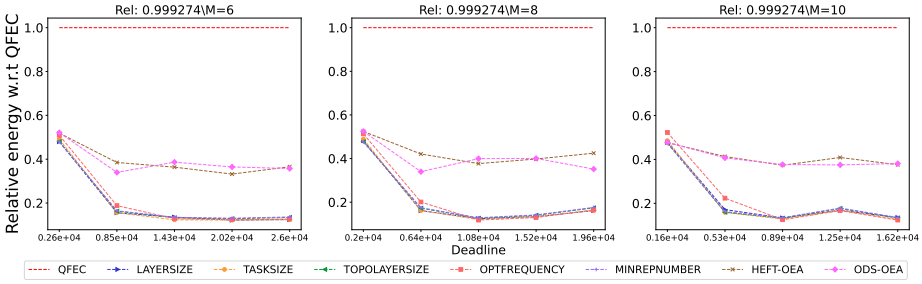

Figure 2751: Assessing the performance of different processor number on the Cycles workflow.

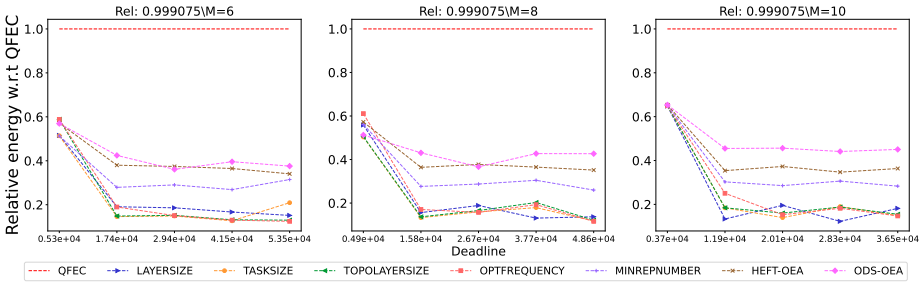

Figure 2752: Assessing the performance of different processor number on the Epigenomics workflow.

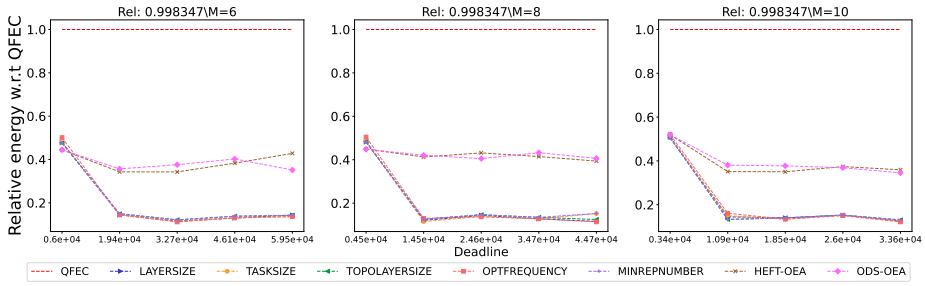

Figure 2753: Assessing the performance of different processor number on the Genome workflow.

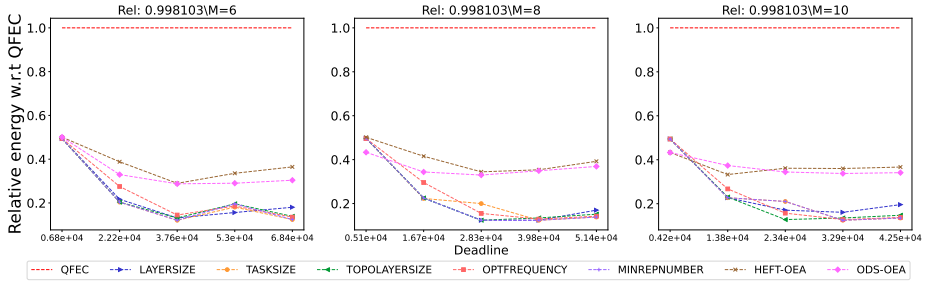

Figure 2754: Assessing the performance of different processor number on the LU.

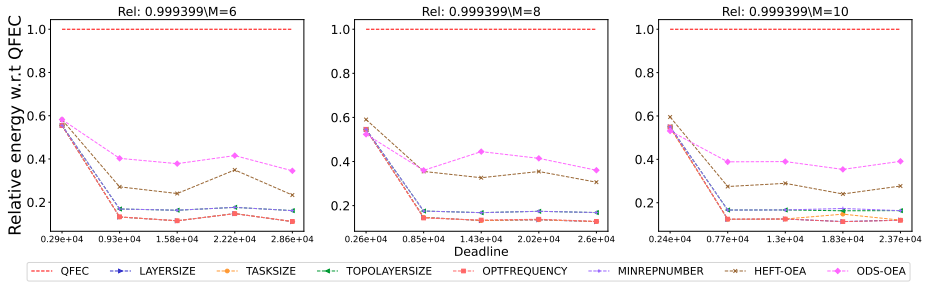

Figure 2755: Assessing the performance of different processor number on the Montage workflow.

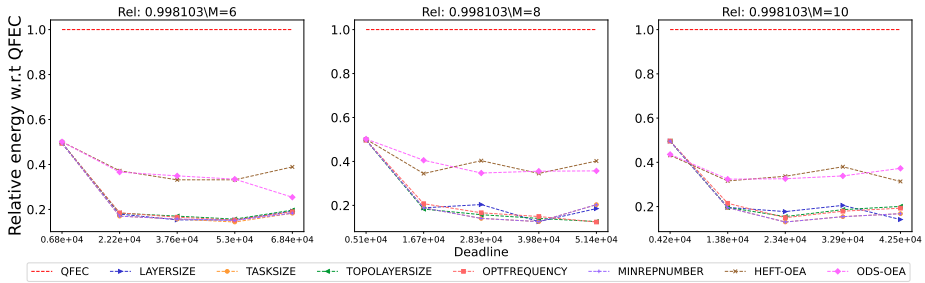

Figure 2756: Assessing the performance of different processor number on the QR.

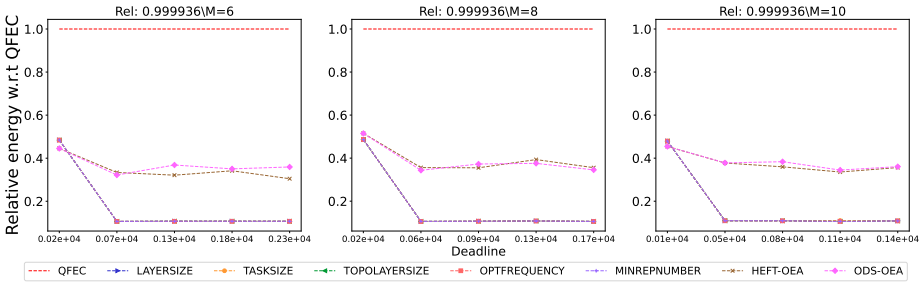

Figure 2757: Assessing the performance of different processor number on the Seismology workflow.

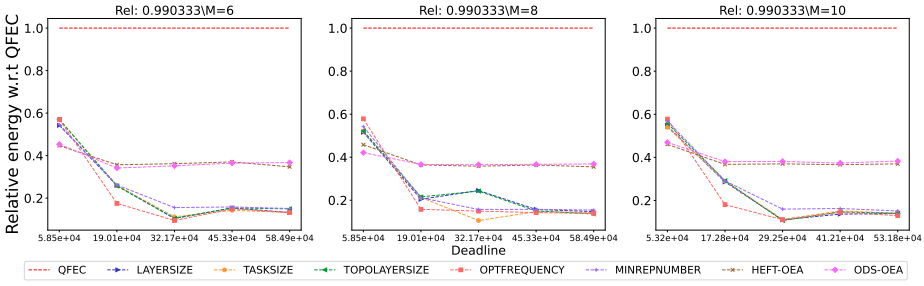

Figure 2758: Assessing the performance of different processor number on the SoyKB workflow.

I.3.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

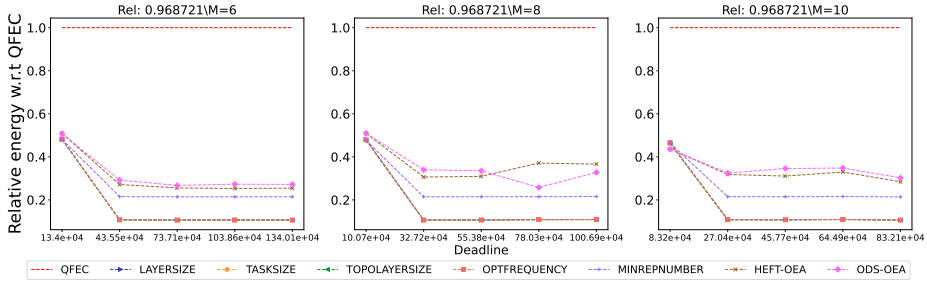

Figure 2759: Assessing the performance of different processor number on the BLAST workflow.

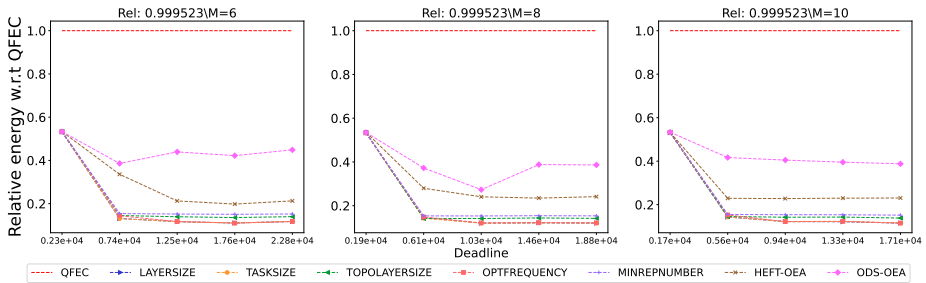

Figure 2760: Assessing the performance of different processor number on the BWA workflow.

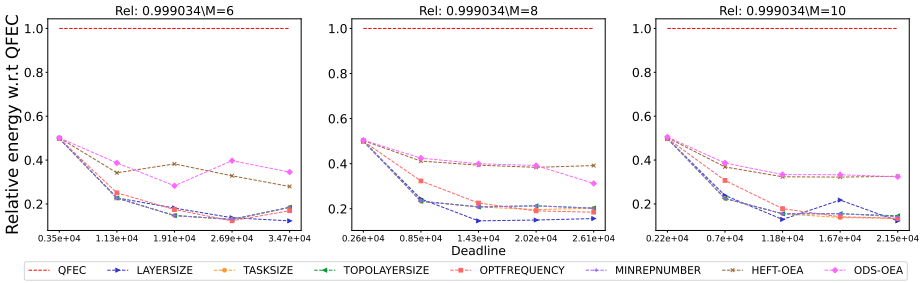

Figure 2761: Assessing the performance of different processor number on the Cholesky workflow.

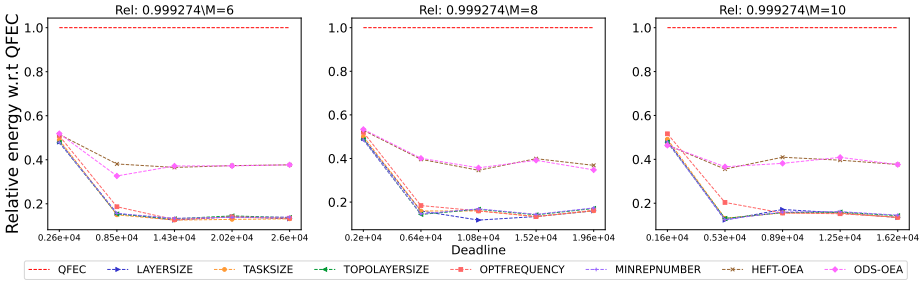

Figure 2762: Assessing the performance of different processor number on the Cycles workflow.

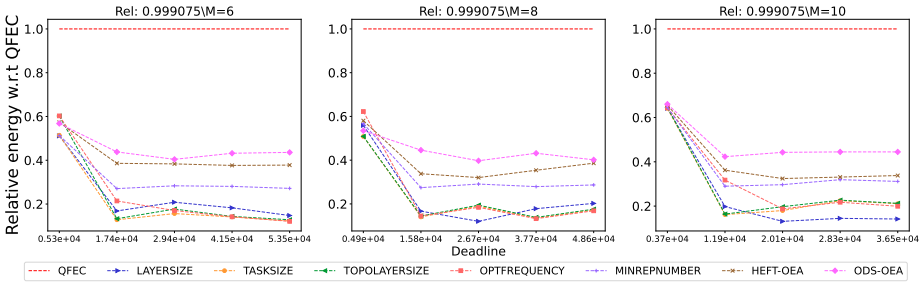

Figure 2763: Assessing the performance of different processor number on the Epigenomics workflow.

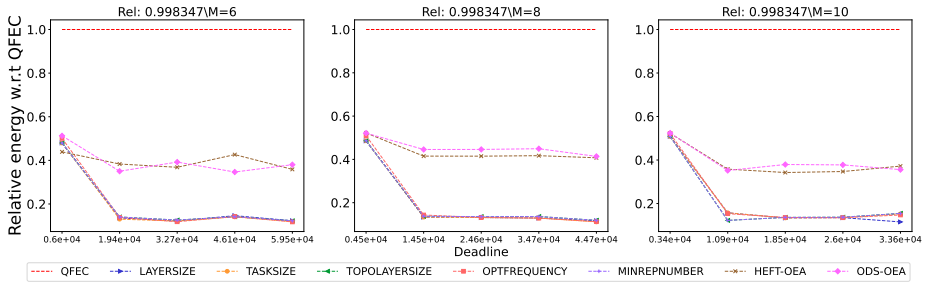

Figure 2764: Assessing the performance of different processor number on the Genome workflow.

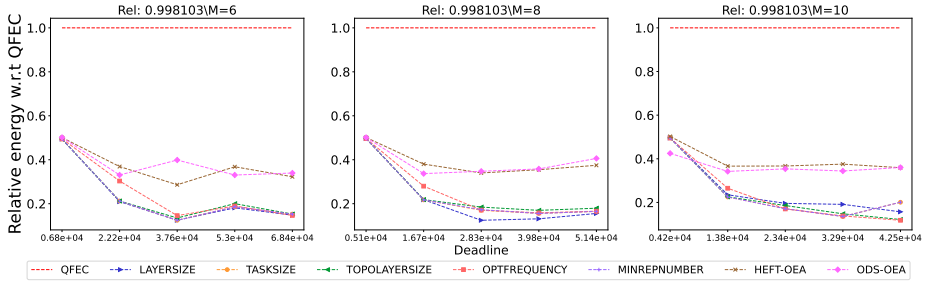

Figure 2765: Assessing the performance of different processor number on the LU.

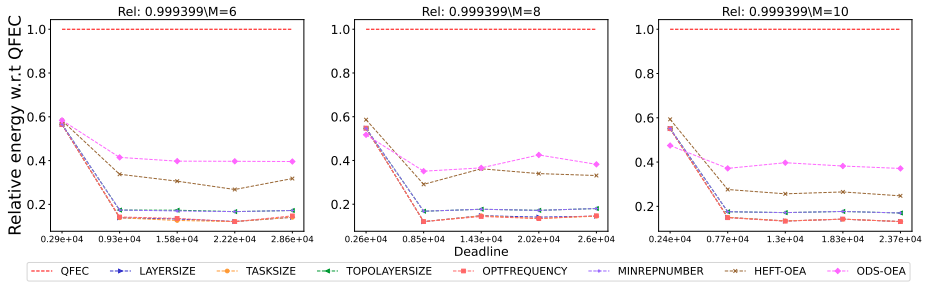

Figure 2766: Assessing the performance of different processor number on the Montage workflow.

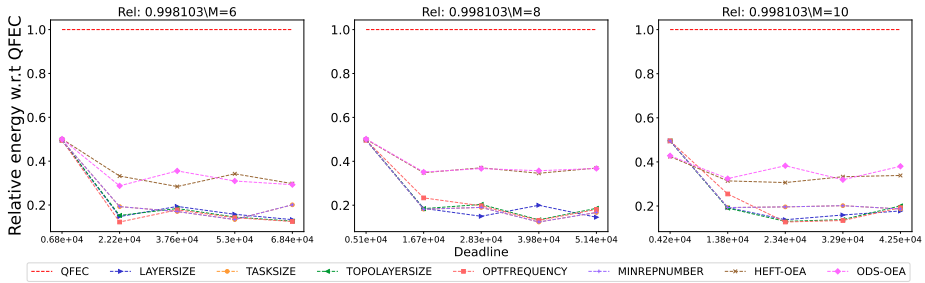

Figure 2767: Assessing the performance of different processor number on the QR.

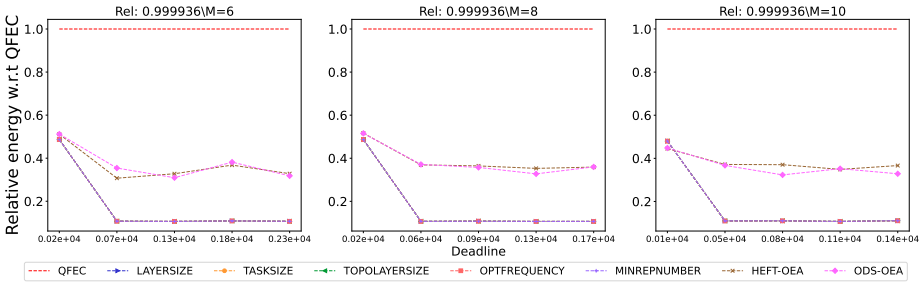

Figure 2768: Assessing the performance of different processor number on the Seismology workflow.

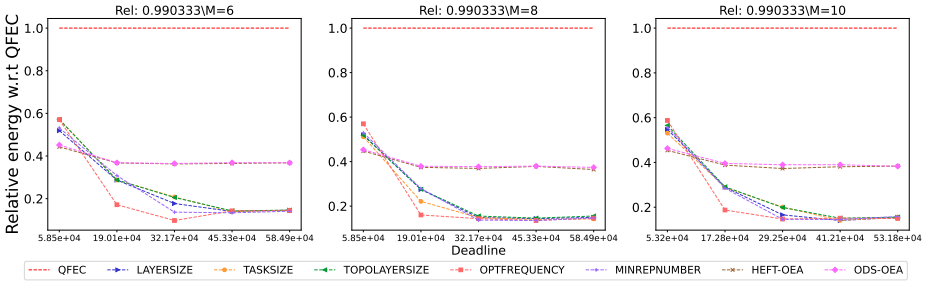

Figure 2769: Assessing the performance of different processor number on the SoyKB workflow.

I.4 $BC/WC = 0.4$

I.4.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

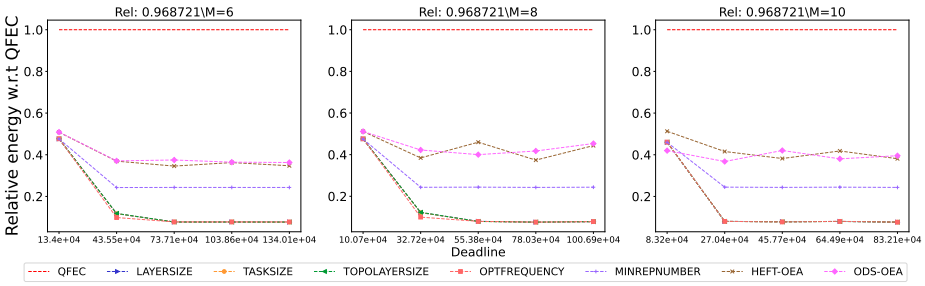

Figure 2770: Assessing the performance of different processor number on the BLAST workflow.

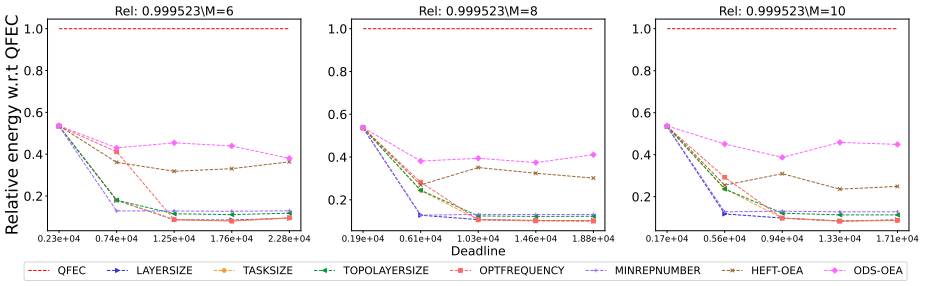

Figure 2771: Assessing the performance of different processor number on the BWA workflow.

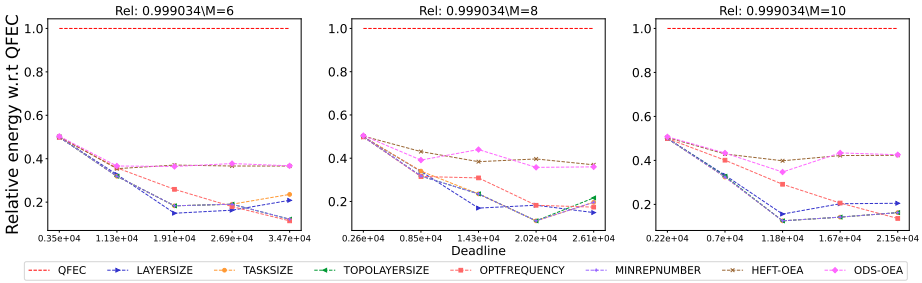

Figure 2772: Assessing the performance of different processor number on the Cholesky workflow.

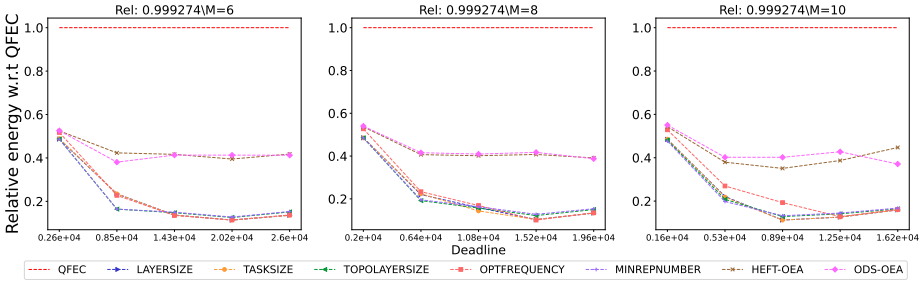

Figure 2773: Assessing the performance of different processor number on the Cycles workflow.

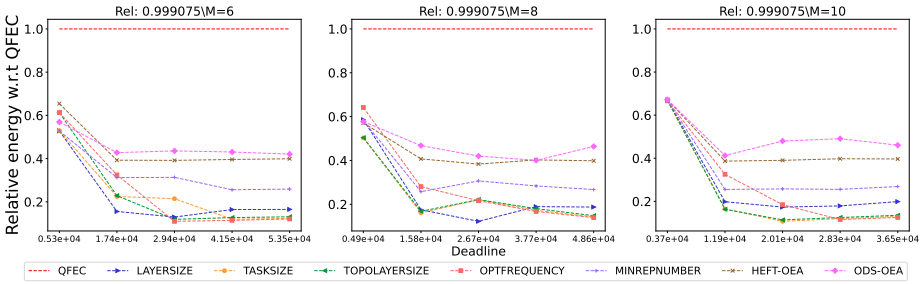

Figure 2774: Assessing the performance of different processor number on the Epigenomics workflow.

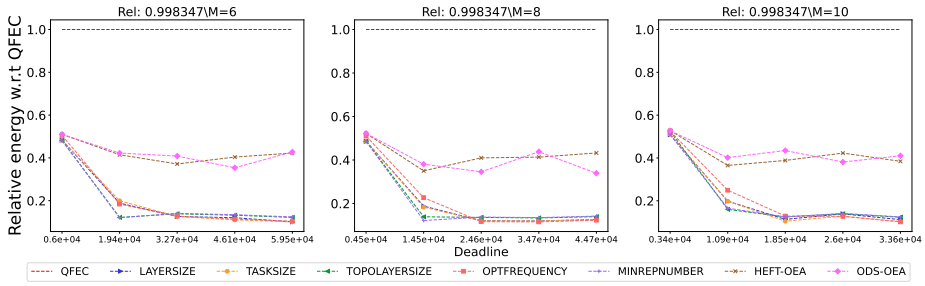

Figure 2775: Assessing the performance of different processor number on the Genome workflow.

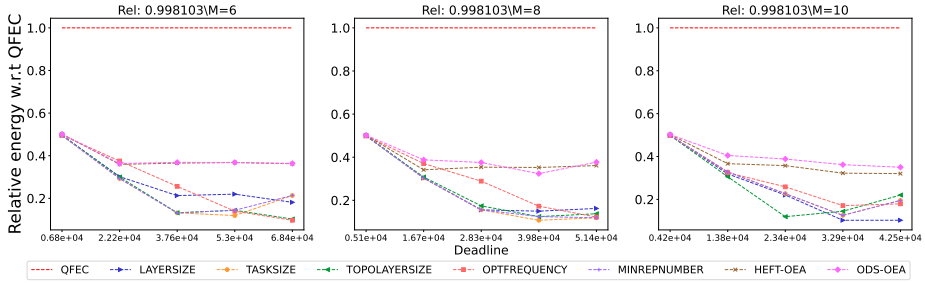

Figure 2776: Assessing the performance of different processor number on the LU.

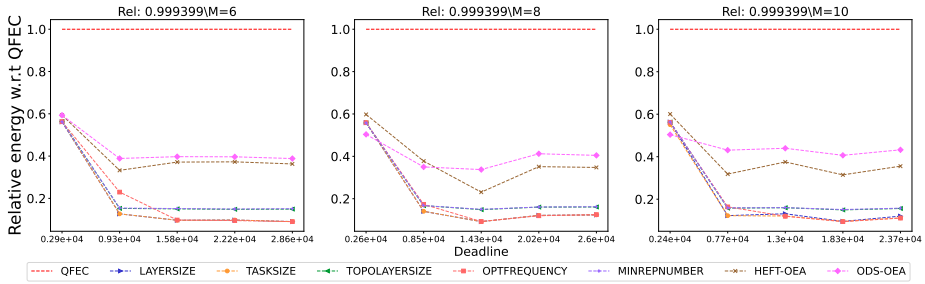

Figure 2777: Assessing the performance of different processor number on the Montage workflow.

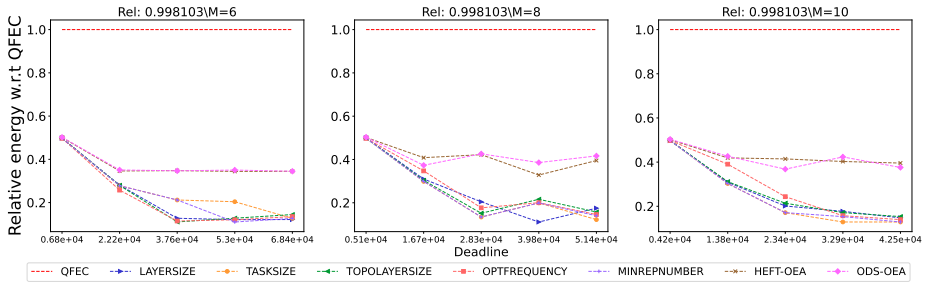

Figure 2778: Assessing the performance of different processor number on the QR.

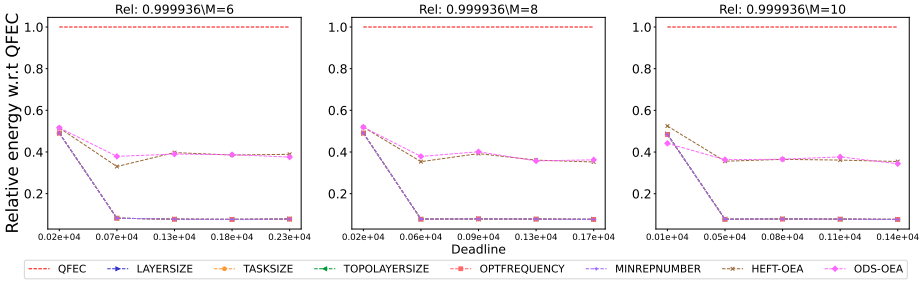

Figure 2779: Assessing the performance of different processor number on the Seismology workflow.

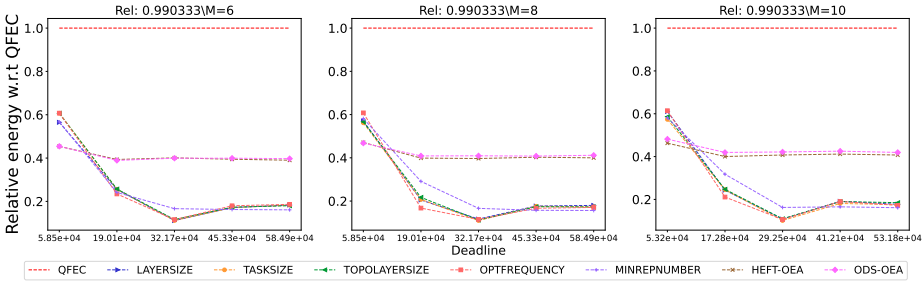

Figure 2780: Assessing the performance of different processor number on the SoyKB workflow.

I.4.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

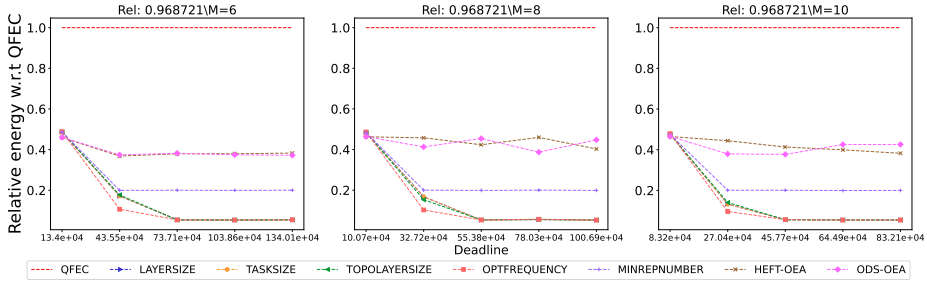

Figure 2781: Assessing the performance of different processor number on the BLAST workflow.

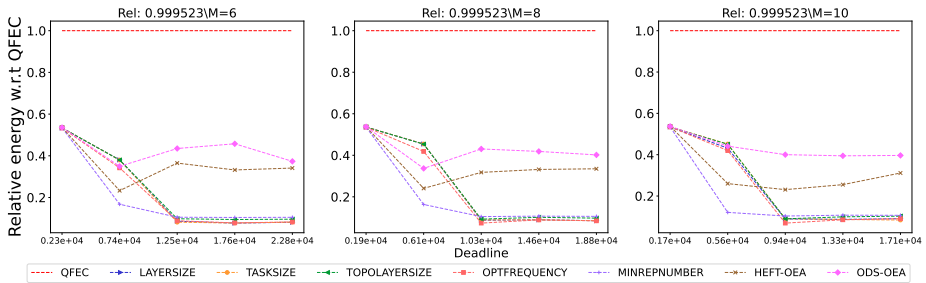

Figure 2782: Assessing the performance of different processor number on the BWA workflow.

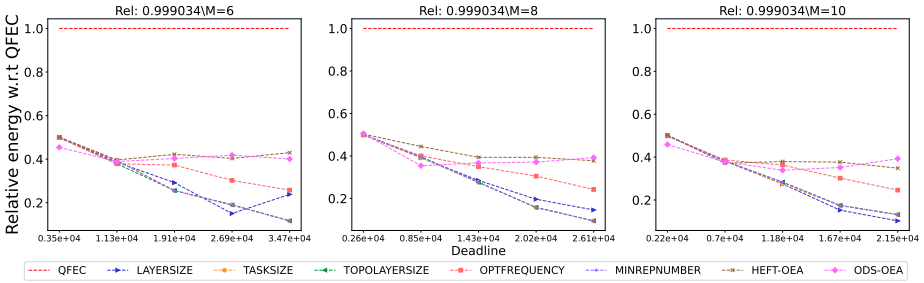

Figure 2783: Assessing the performance of different processor number on the Cholesky workflow.

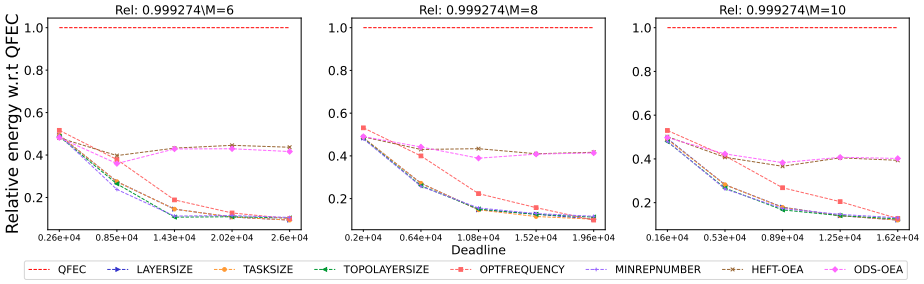

Figure 2784: Assessing the performance of different processor number on the Cycles workflow.

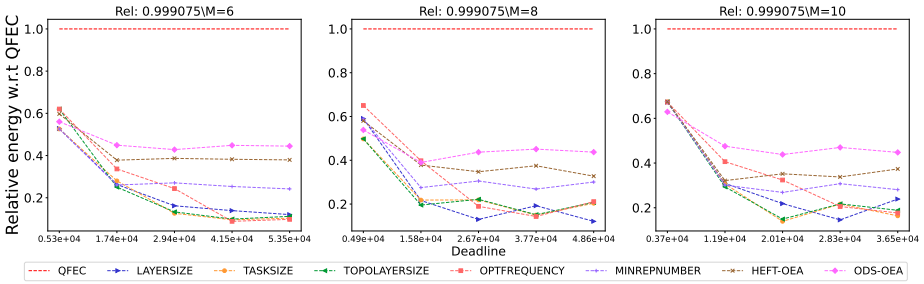

Figure 2785: Assessing the performance of different processor number on the Epigenomics workflow.

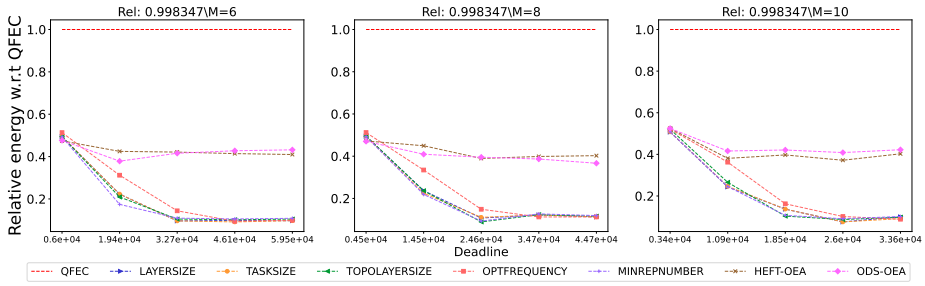

Figure 2786: Assessing the performance of different processor number on the Genome workflow.

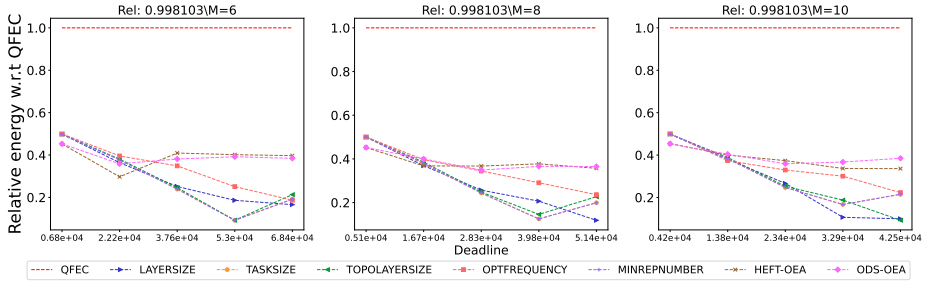

Figure 2787: Assessing the performance of different processor number on the LU.

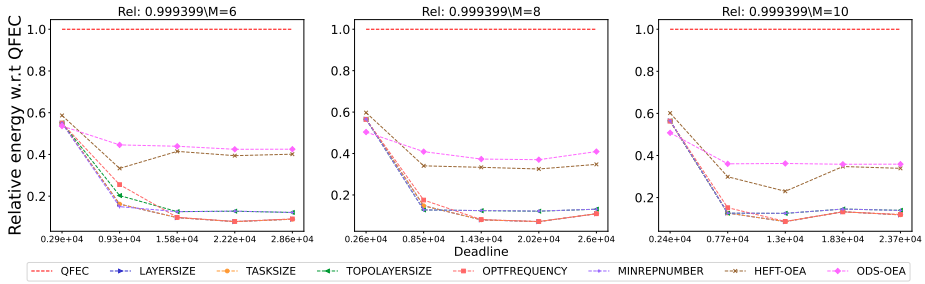

Figure 2788: Assessing the performance of different processor number on the Montage workflow.

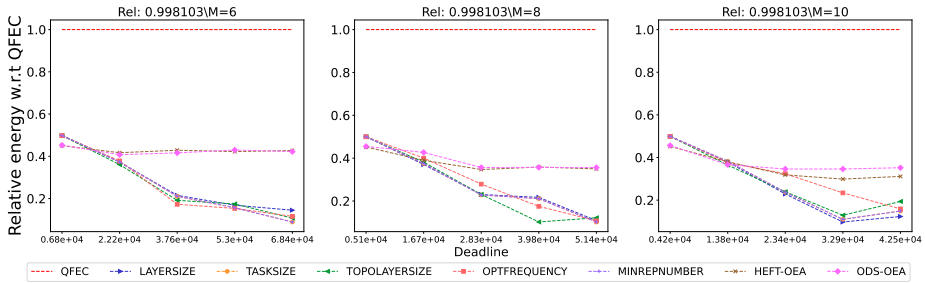

Figure 2789: Assessing the performance of different processor number on the QR.

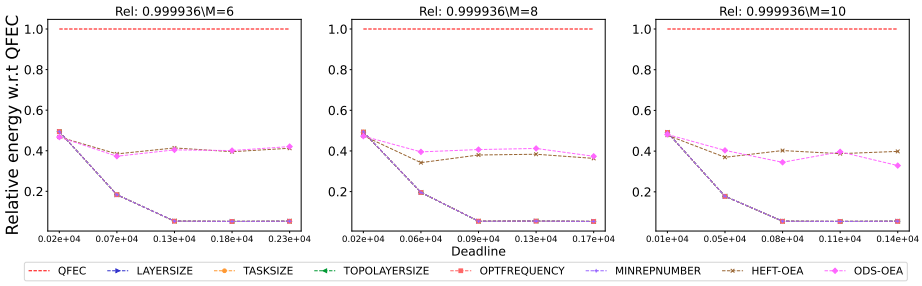

Figure 2790: Assessing the performance of different processor number on the Seismology workflow.

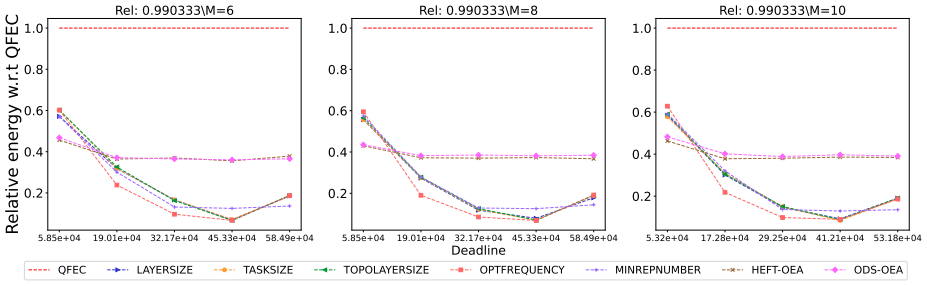

Figure 2791: Assessing the performance of different processor number on the SoyKB workflow.

I.4.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

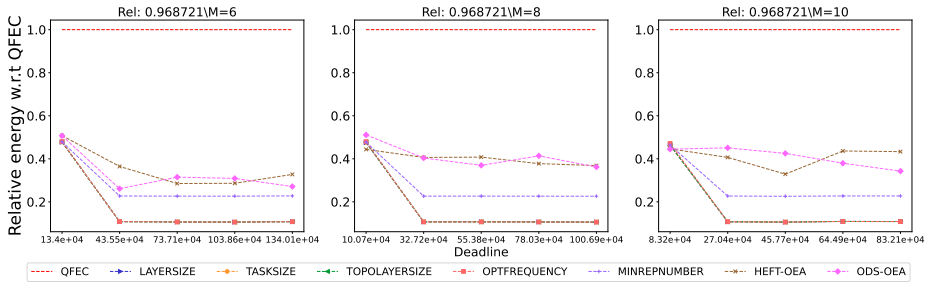

Figure 2792: Assessing the performance of different processor number on the BLAST workflow.

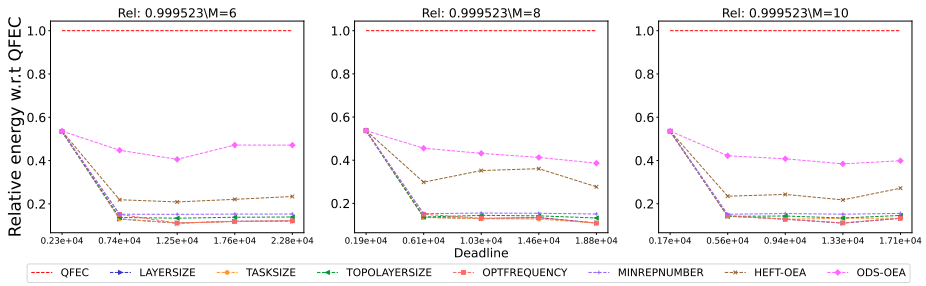

Figure 2793: Assessing the performance of different processor number on the BWA workflow.

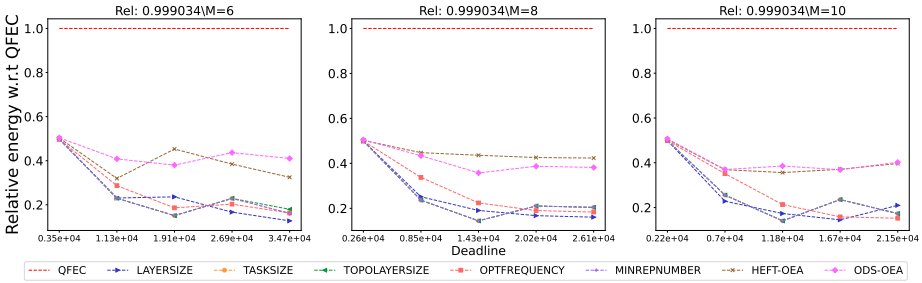

Figure 2794: Assessing the performance of different processor number on the Cholesky workflow.

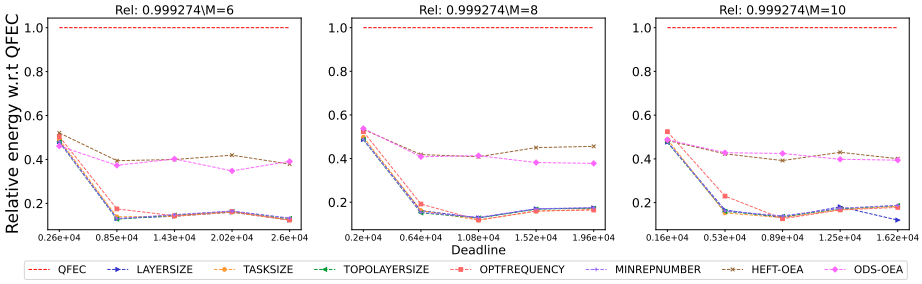

Figure 2795: Assessing the performance of different processor number on the Cycles workflow.

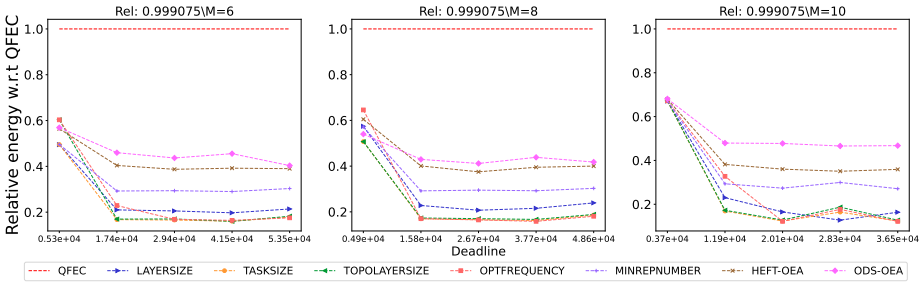

Figure 2796: Assessing the performance of different processor number on the Epigenomics workflow.

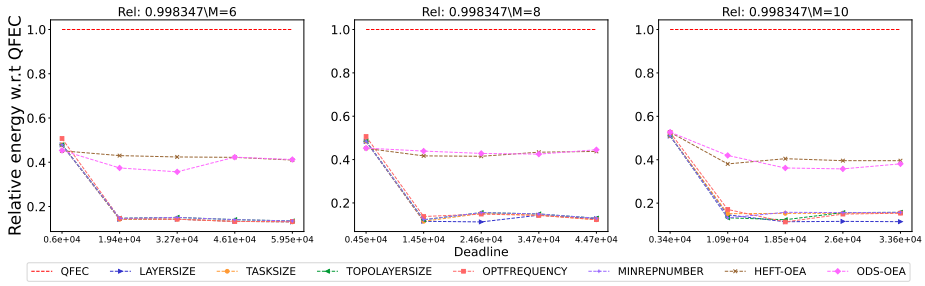

Figure 2797: Assessing the performance of different processor number on the Genome workflow.

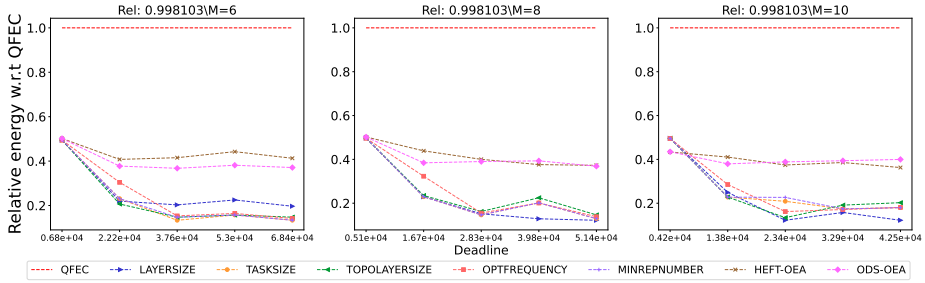

Figure 2798: Assessing the performance of different processor number on the LU.

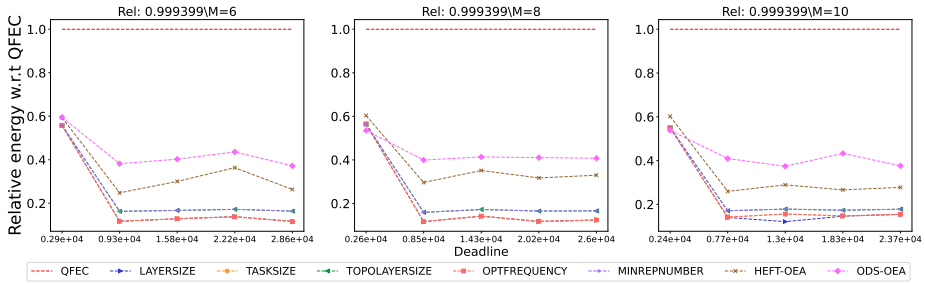

Figure 2799: Assessing the performance of different processor number on the Montage workflow.

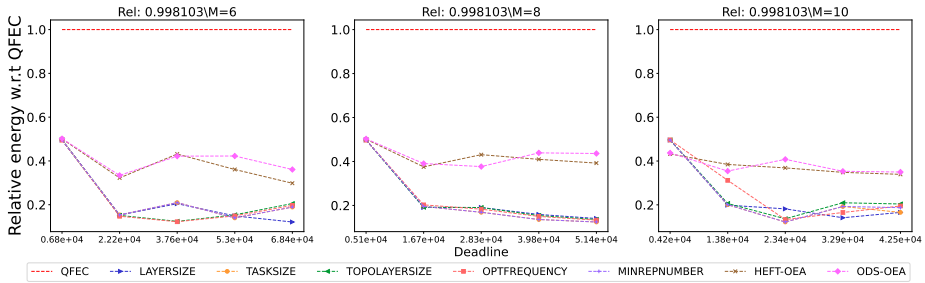

Figure 2800: Assessing the performance of different processor number on the QR.

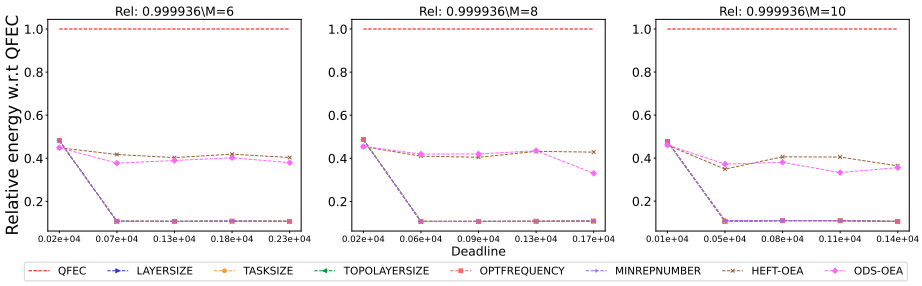

Figure 2801: Assessing the performance of different processor number on the Seismology workflow.

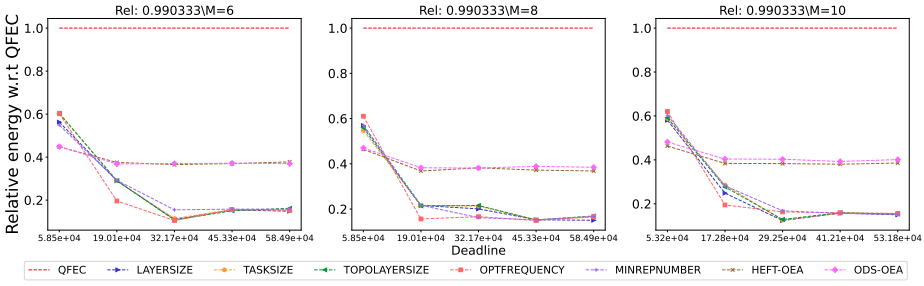

Figure 2802: Assessing the performance of different processor number on the SoyKB workflow.

I.4.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

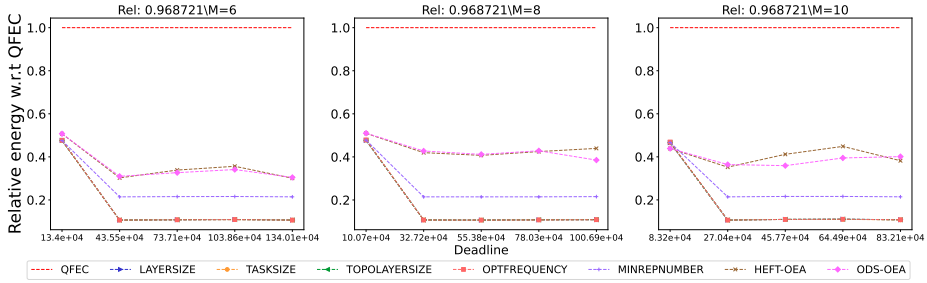

Figure 2803: Assessing the performance of different processor number on the BLAST workflow.

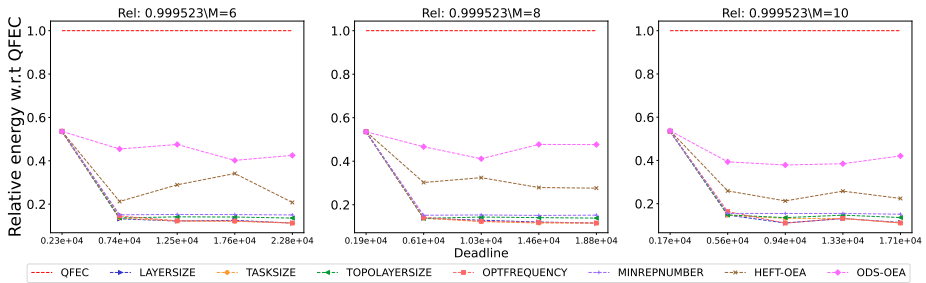

Figure 2804: Assessing the performance of different processor number on the BWA workflow.

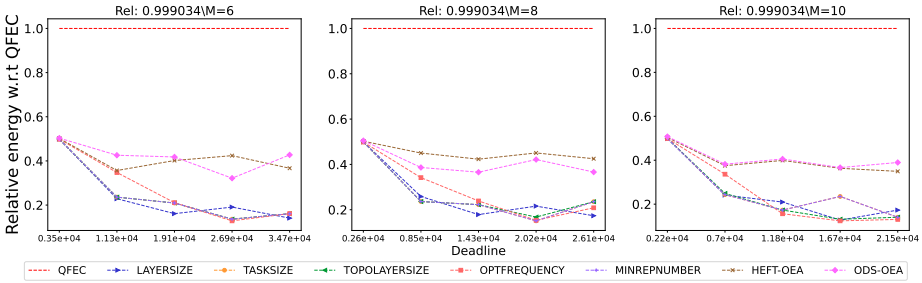

Figure 2805: Assessing the performance of different processor number on the Cholesky workflow.

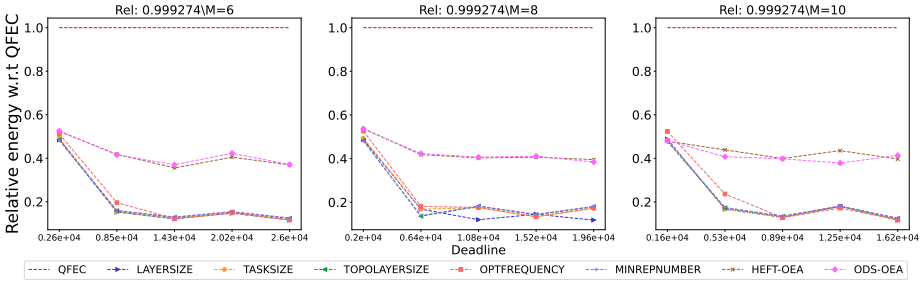

Figure 2806: Assessing the performance of different processor number on the Cycles workflow.

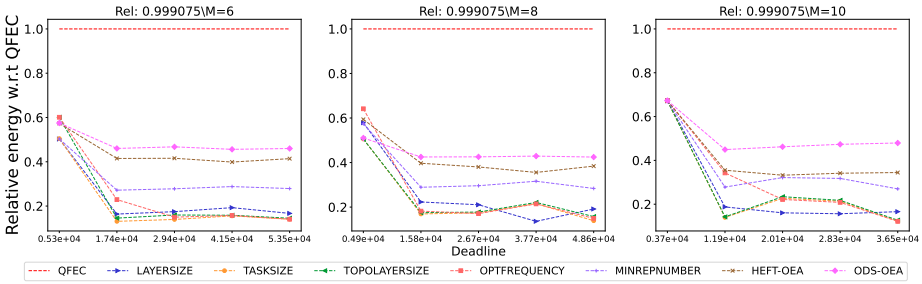

Figure 2807: Assessing the performance of different processor number on the Epigenomics workflow.

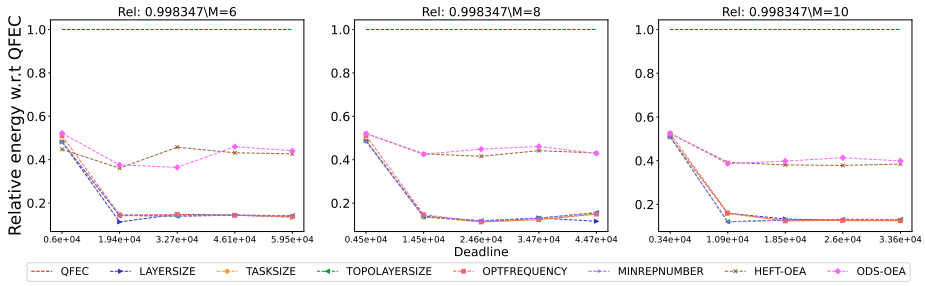

Figure 2808: Assessing the performance of different processor number on the Genome workflow.

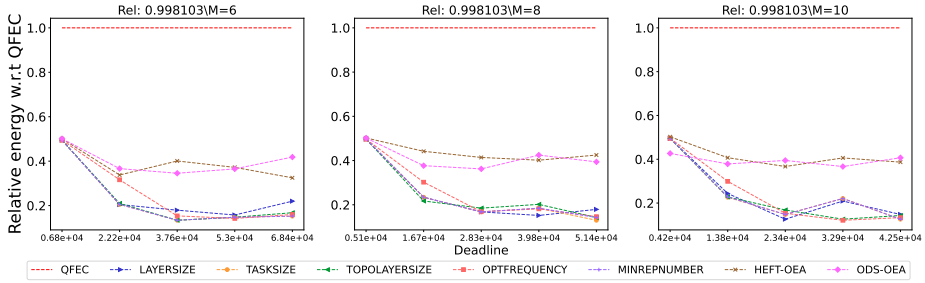

Figure 2809: Assessing the performance of different processor number on the LU.

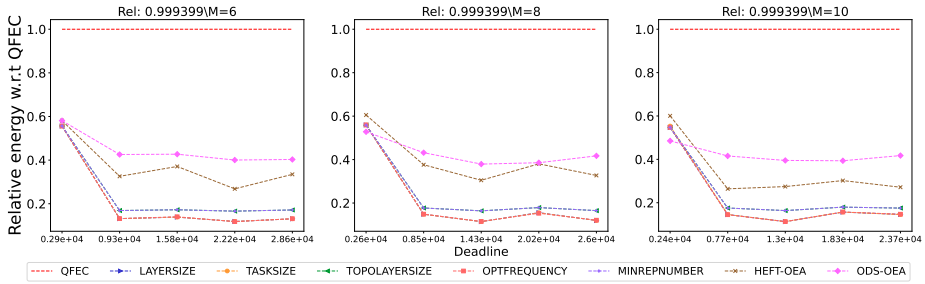

Figure 2810: Assessing the performance of different processor number on the Montage workflow.

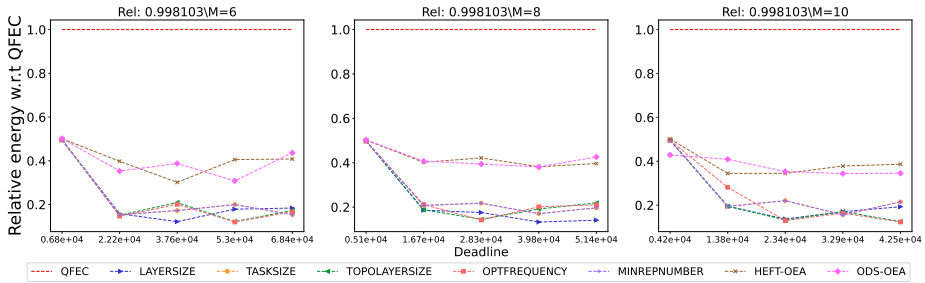

Figure 2811: Assessing the performance of different processor number on the QR.

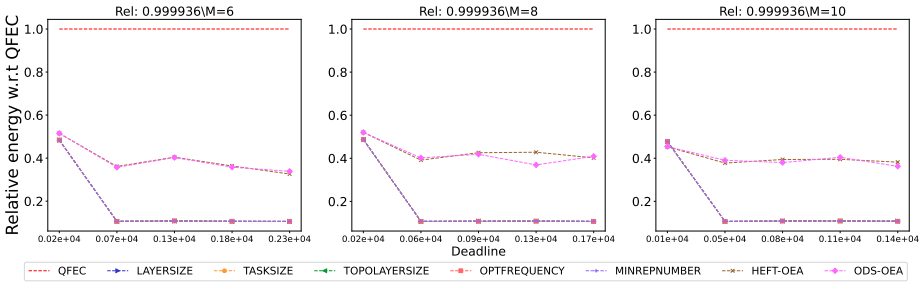

Figure 2812: Assessing the performance of different processor number on the Seismology workflow.

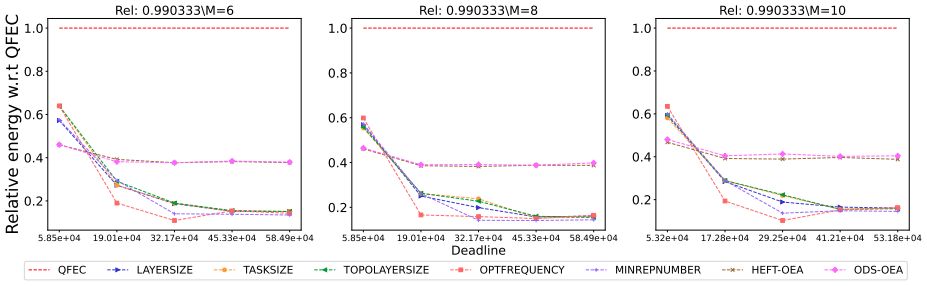

Figure 2813: Assessing the performance of different processor number on the SoyKB workflow.

I.5 $BC/WC = 0.5$

I.5.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

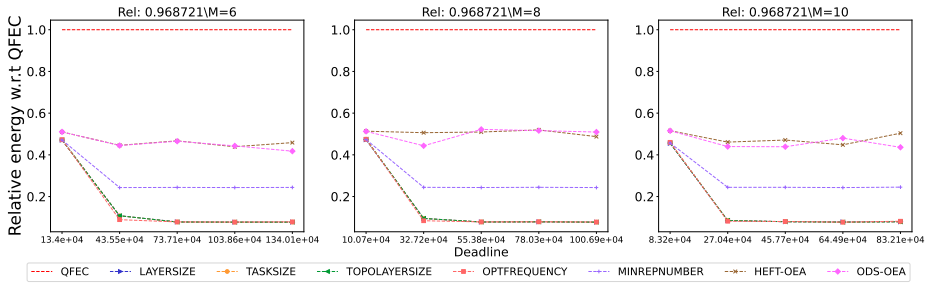

Figure 2814: Assessing the performance of different processor number on the BLAST workflow.

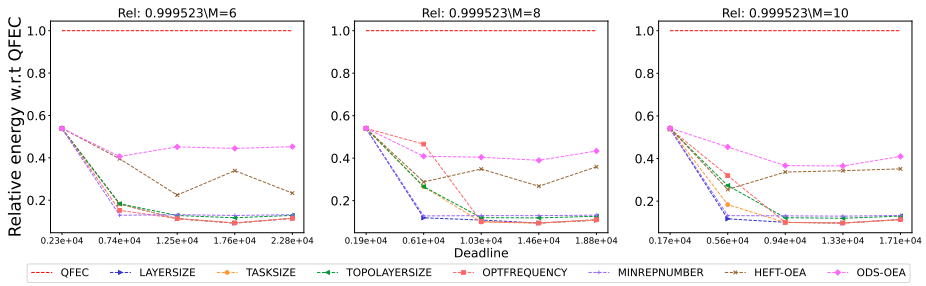

Figure 2815: Assessing the performance of different processor number on the BWA workflow.

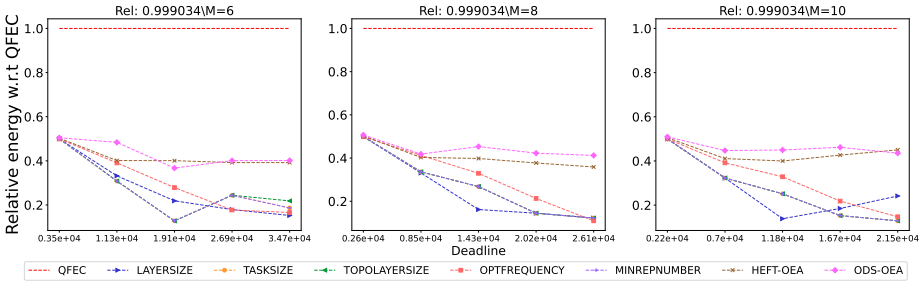

Figure 2816: Assessing the performance of different processor number on the Cholesky workflow.

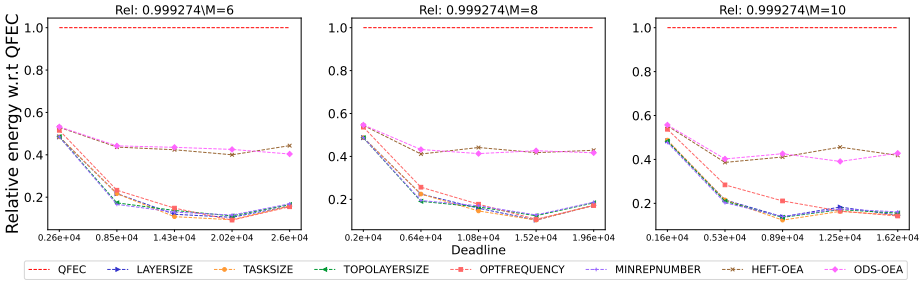

Figure 2817: Assessing the performance of different processor number on the Cycles workflow.

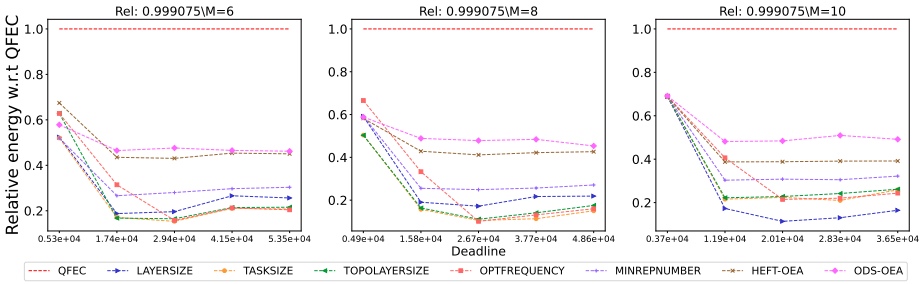

Figure 2818: Assessing the performance of different processor number on the Epigenomics workflow.

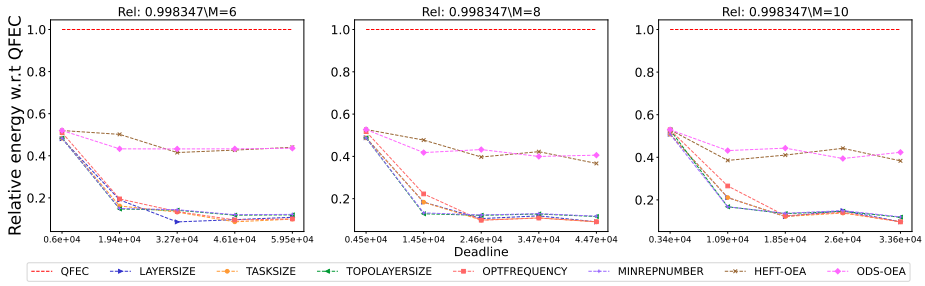

Figure 2819: Assessing the performance of different processor number on the Genome workflow.

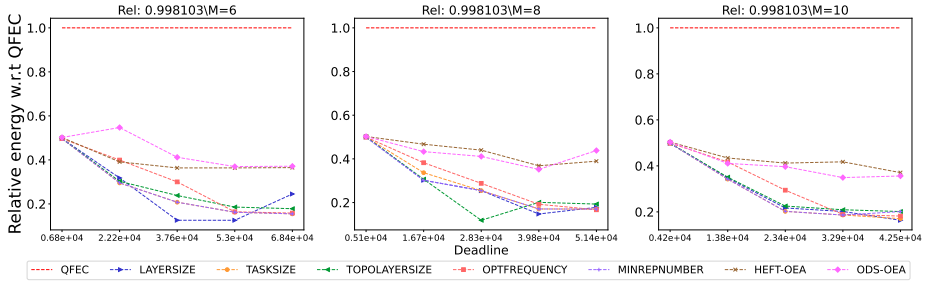

Figure 2820: Assessing the performance of different processor number on the LU.

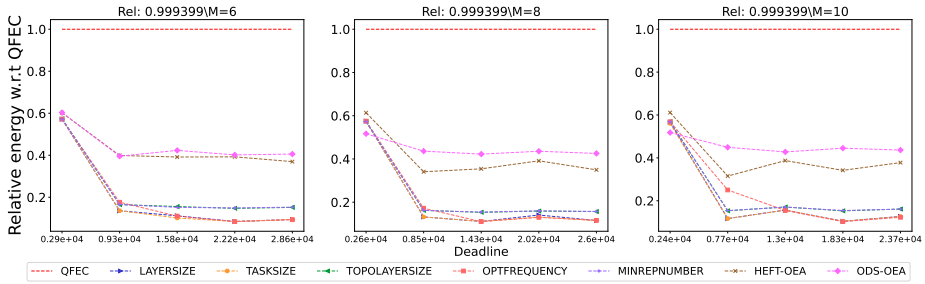

Figure 2821: Assessing the performance of different processor number on the Montage workflow.

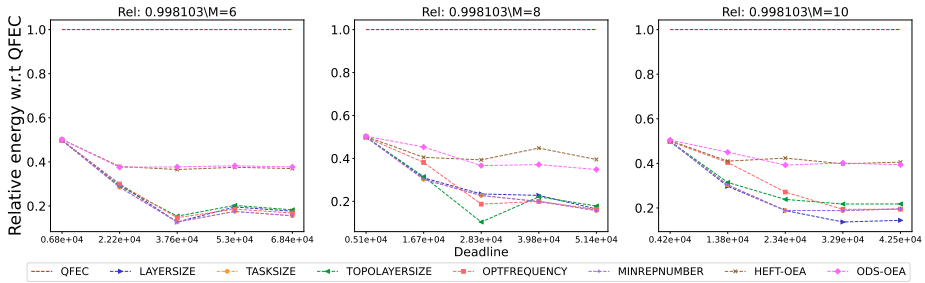

Figure 2822: Assessing the performance of different processor number on the QR.

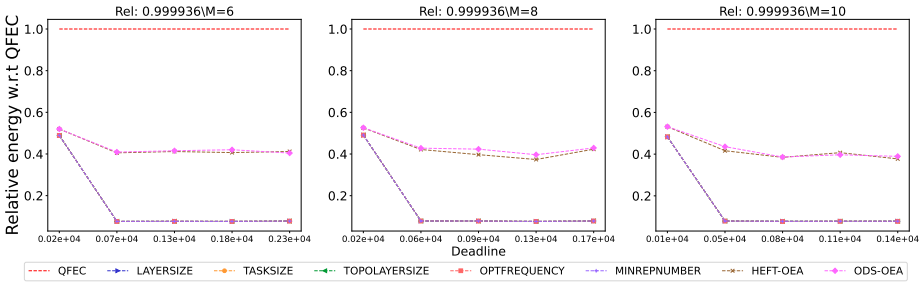

Figure 2823: Assessing the performance of different processor number on the Seismology workflow.

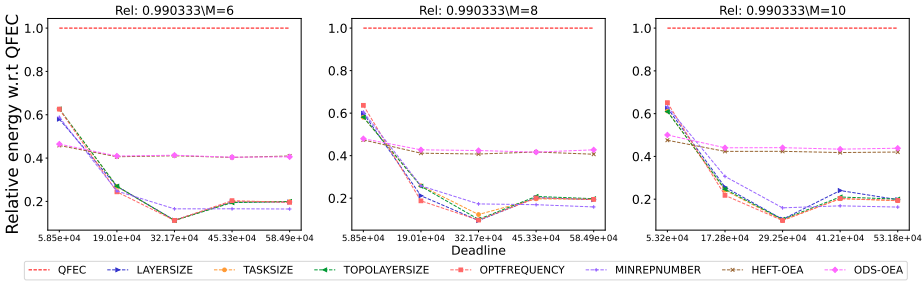

Figure 2824: Assessing the performance of different processor number on the SoyKB workflow.

I.5.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

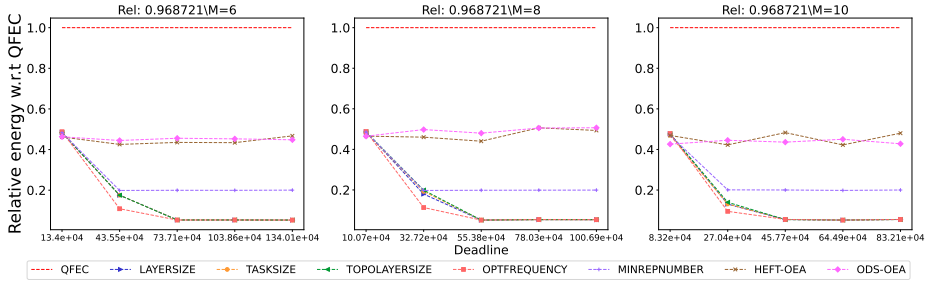

Figure 2825: Assessing the performance of different processor number on the BLAST workflow.

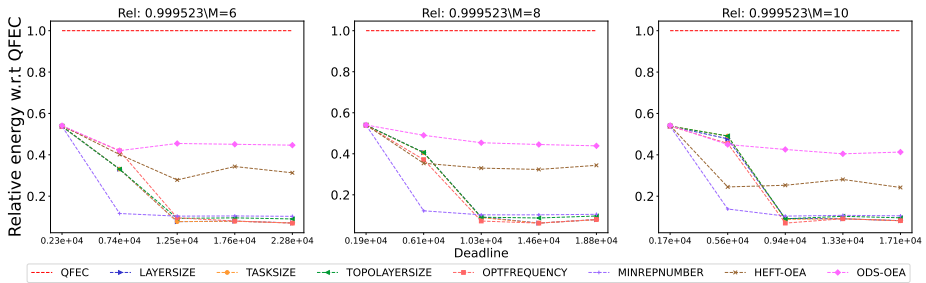

Figure 2826: Assessing the performance of different processor number on the BWA workflow.

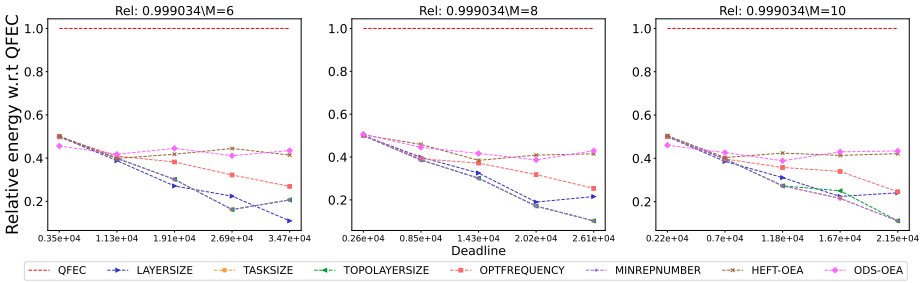

Figure 2827: Assessing the performance of different processor number on the Cholesky workflow.

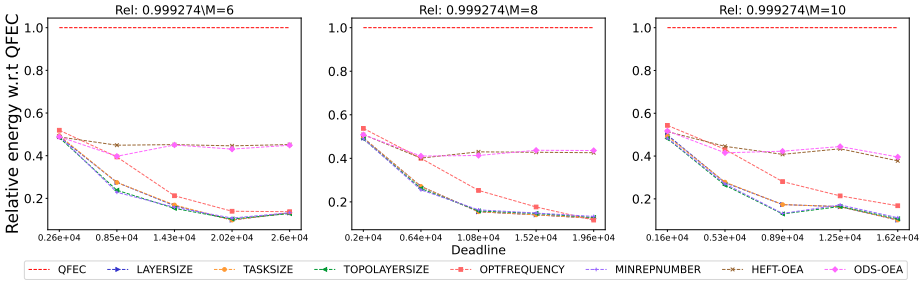

Figure 2828: Assessing the performance of different processor number on the Cycles workflow.

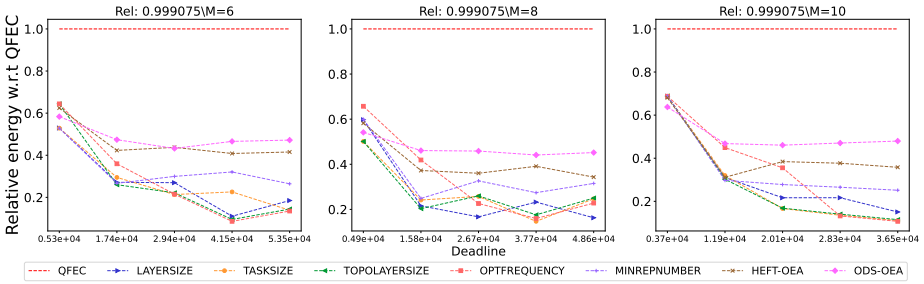

Figure 2829: Assessing the performance of different processor number on the Epigenomics workflow.

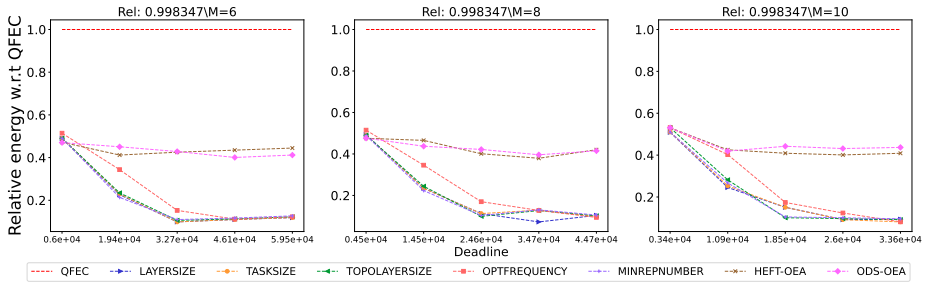

Figure 2830: Assessing the performance of different processor number on the Genome workflow.

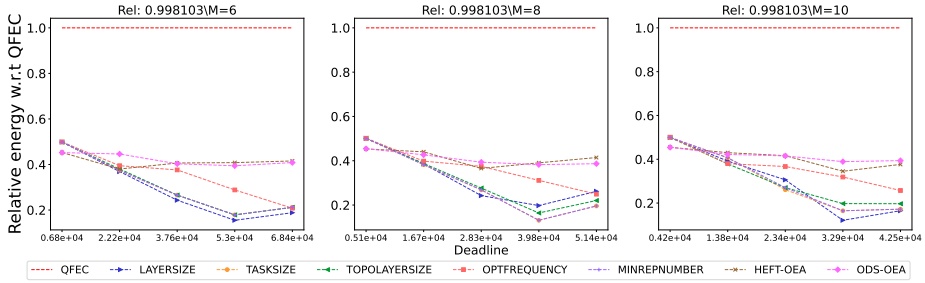

Figure 2831: Assessing the performance of different processor number on the LU.

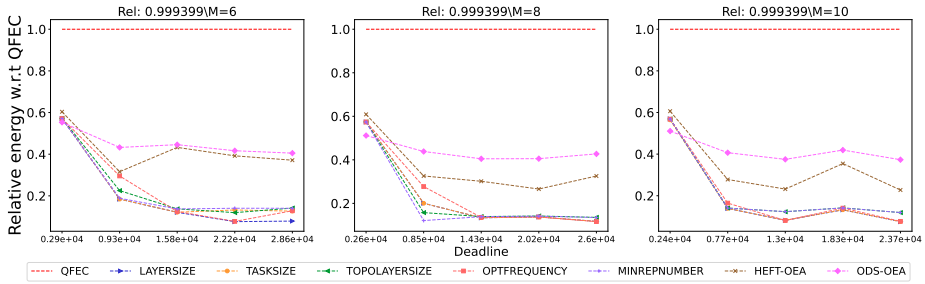

Figure 2832: Assessing the performance of different processor number on the Montage workflow.

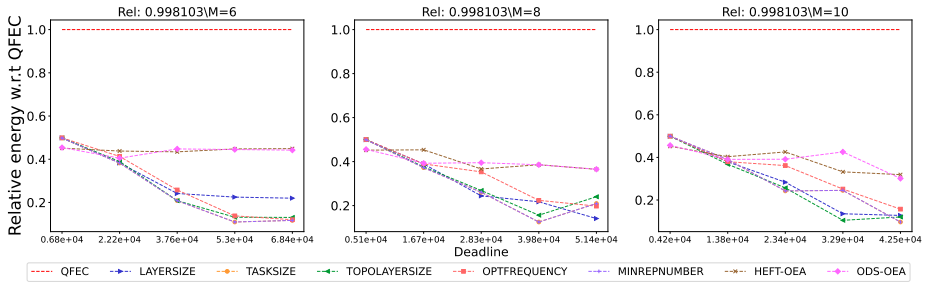

Figure 2833: Assessing the performance of different processor number on the QR.

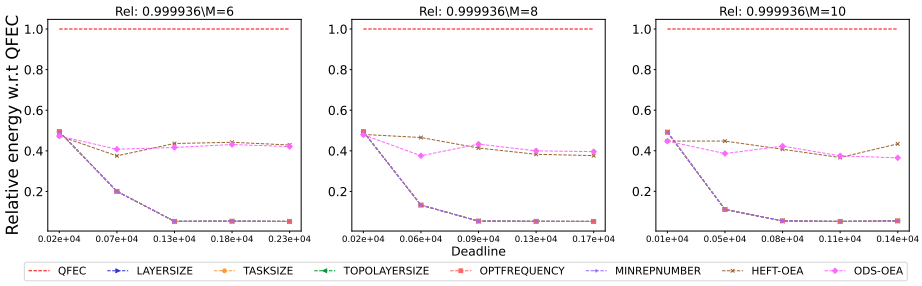

Figure 2834: Assessing the performance of different processor number on the Seismology workflow.

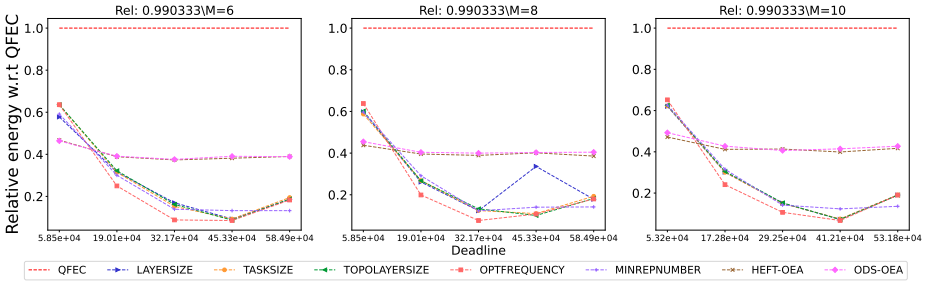

Figure 2835: Assessing the performance of different processor number on the SoyKB workflow.

I.5.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

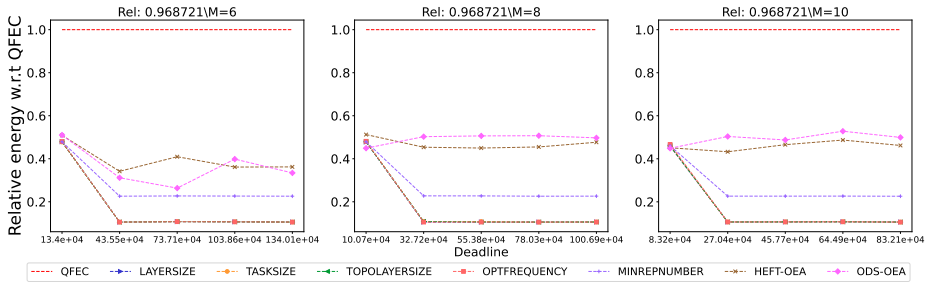

Figure 2836: Assessing the performance of different processor number on the BLAST workflow.

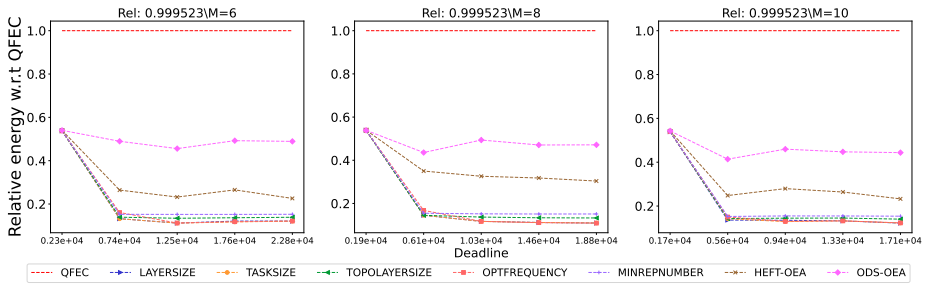

Figure 2837: Assessing the performance of different processor number on the BWA workflow.

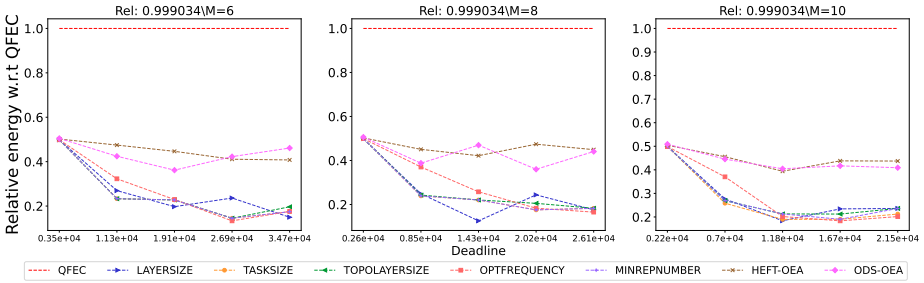

Figure 2838: Assessing the performance of different processor number on the Cholesky workflow.

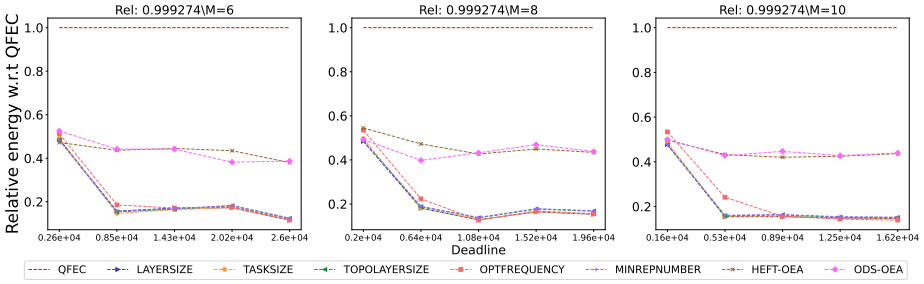

Figure 2839: Assessing the performance of different processor number on the Cycles workflow.

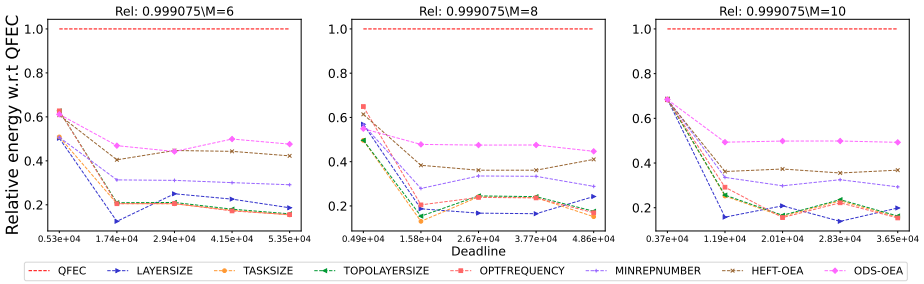

Figure 2840: Assessing the performance of different processor number on the Epigenomics workflow.

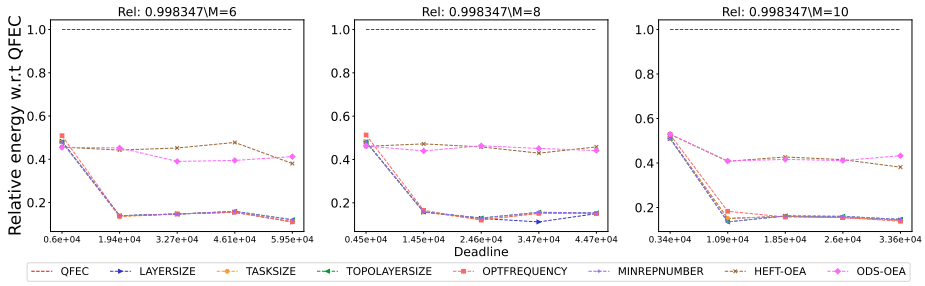

Figure 2841: Assessing the performance of different processor number on the Genome workflow.

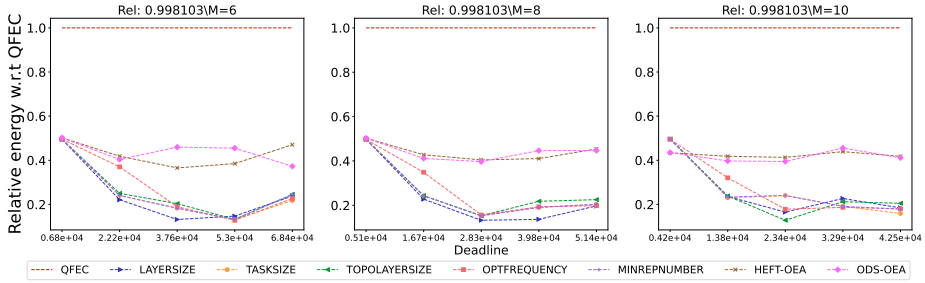

Figure 2842: Assessing the performance of different processor number on the LU.

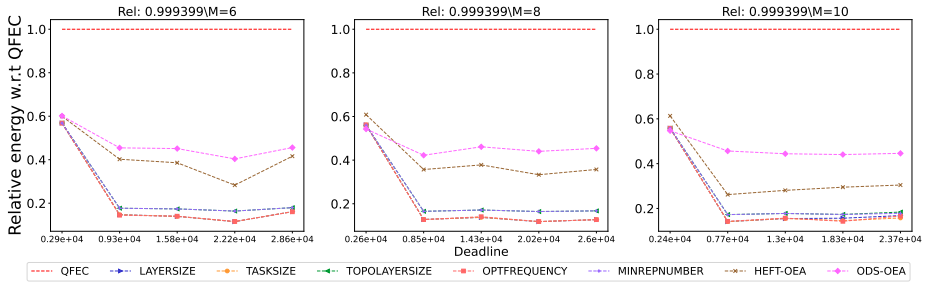

Figure 2843: Assessing the performance of different processor number on the Montage workflow.

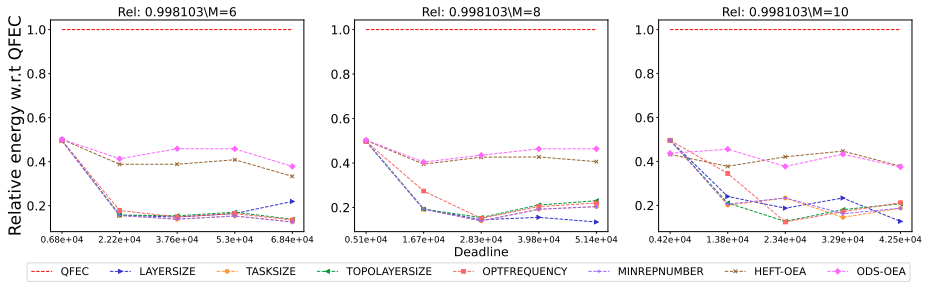

Figure 2844: Assessing the performance of different processor number on the QR.

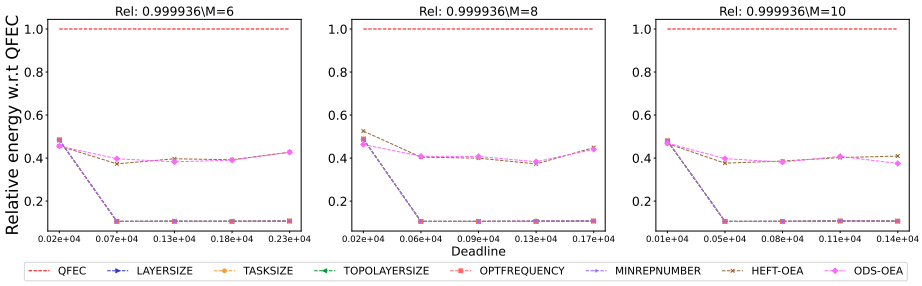

Figure 2845: Assessing the performance of different processor number on the Seismology workflow.

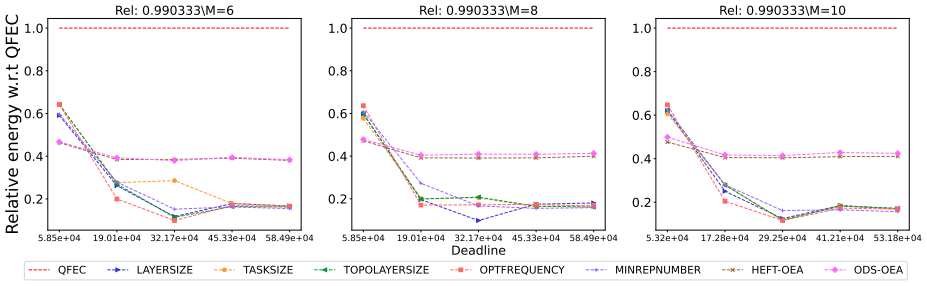

Figure 2846: Assessing the performance of different processor number on the SoyKB workflow.

I.5.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

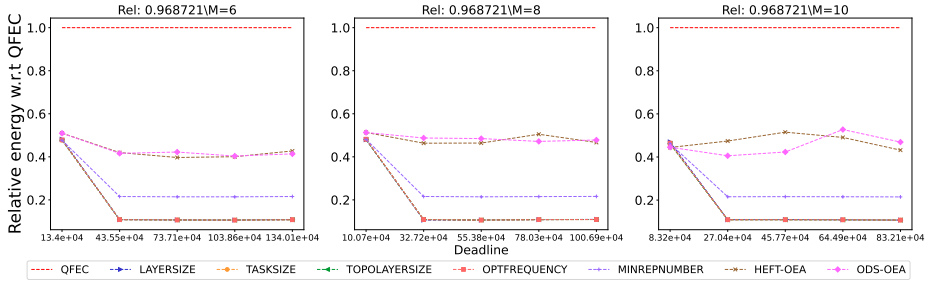

Figure 2847: Assessing the performance of different processor number on the BLAST workflow.

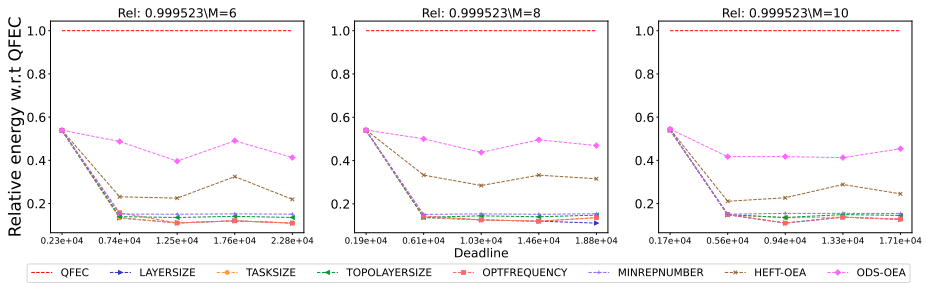

Figure 2848: Assessing the performance of different processor number on the BWA workflow.

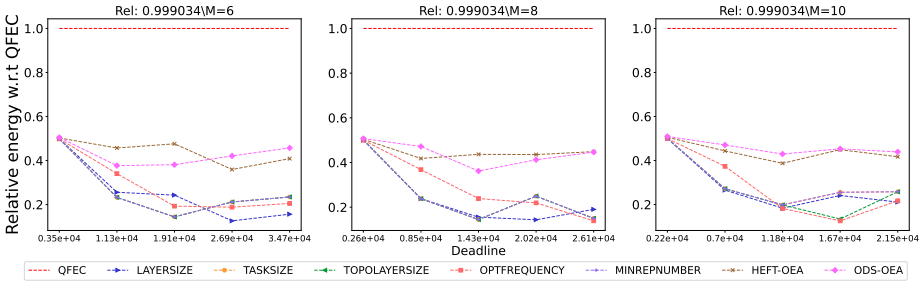

Figure 2849: Assessing the performance of different processor number on the Cholesky workflow.

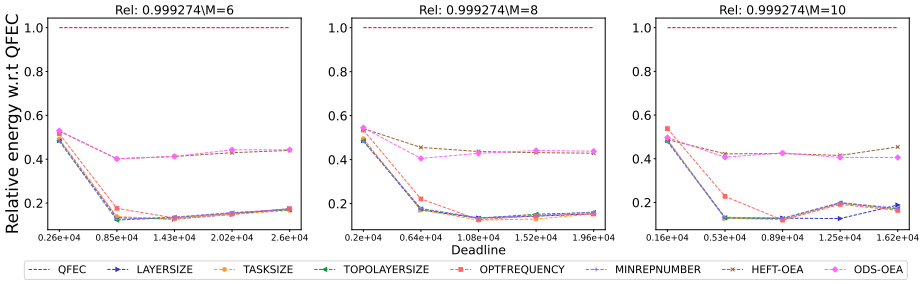

Figure 2850: Assessing the performance of different processor number on the Cycles workflow.

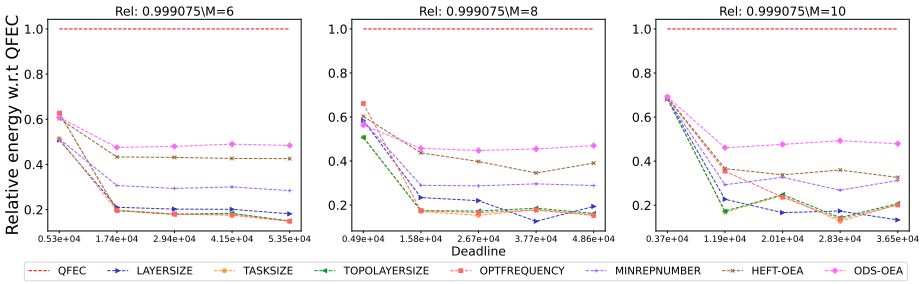

Figure 2851: Assessing the performance of different processor number on the Epigenomics workflow.

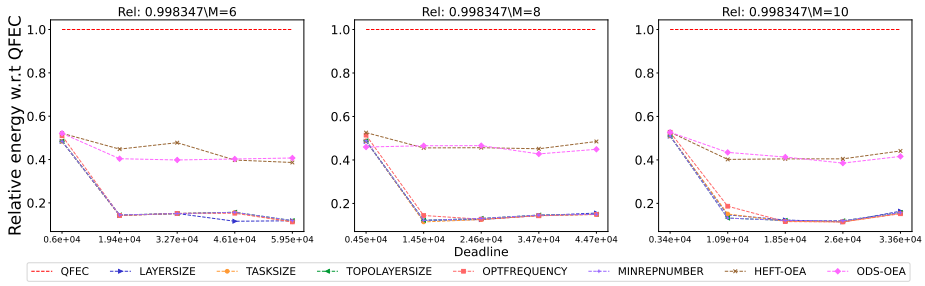

Figure 2852: Assessing the performance of different processor number on the Genome workflow.

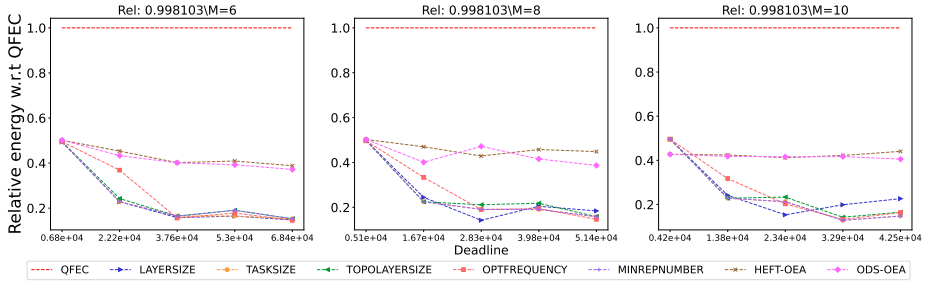

Figure 2853: Assessing the performance of different processor number on the LU.

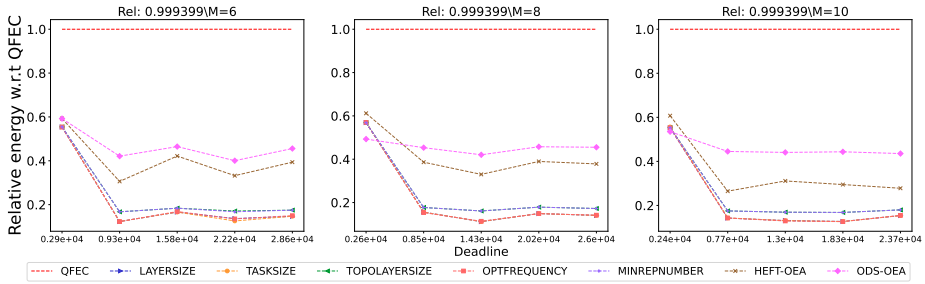

Figure 2854: Assessing the performance of different processor number on the Montage workflow.

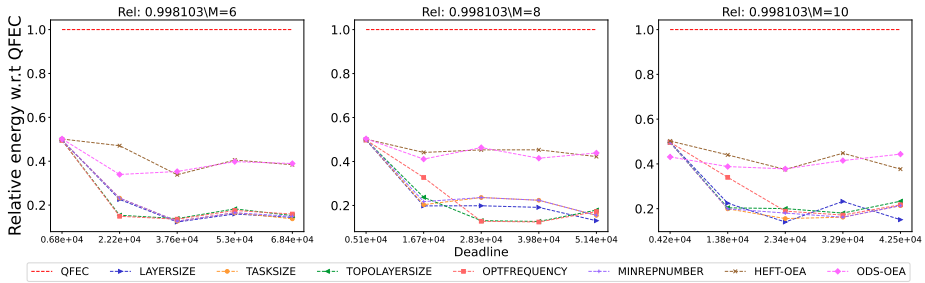

Figure 2855: Assessing the performance of different processor number on the QR.

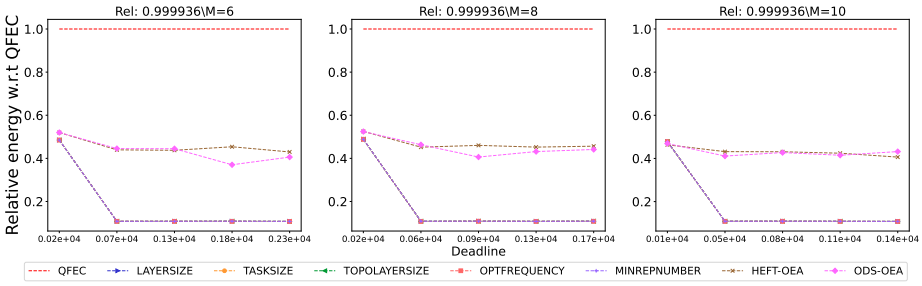

Figure 2856: Assessing the performance of different processor number on the Seismology workflow.

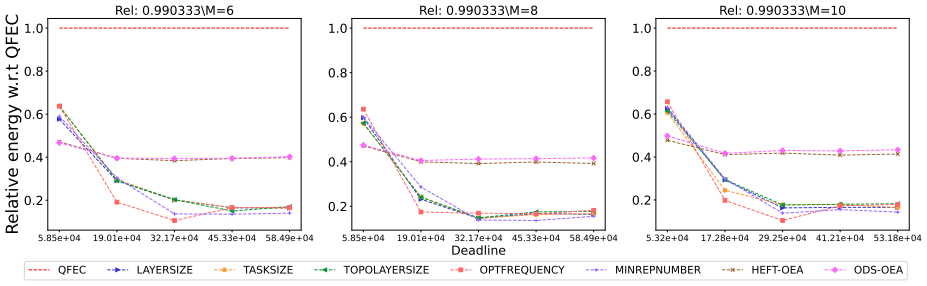

Figure 2857: Assessing the performance of different processor number on the SoyKB workflow.

I.6 $BC/WC = 0.6$

I.6.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

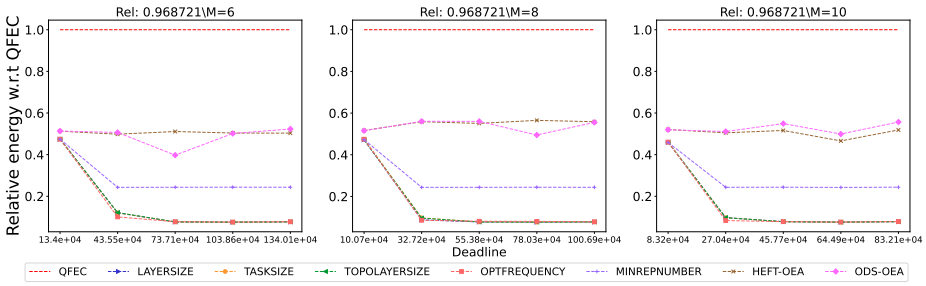

Figure 2858: Assessing the performance of different processor number on the BLAST workflow.

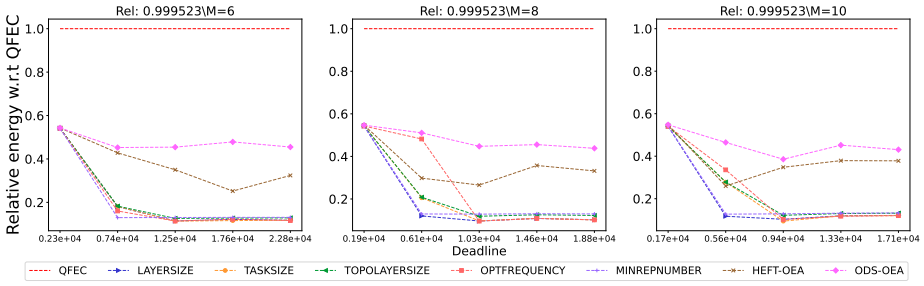

Figure 2859: Assessing the performance of different processor number on the BWA workflow.

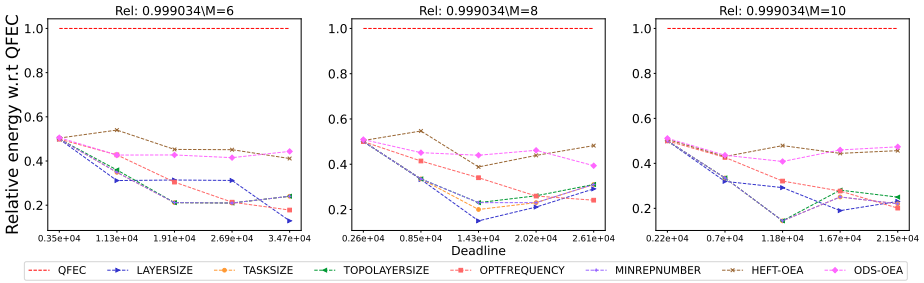

Figure 2860: Assessing the performance of different processor number on the Cholesky workflow.

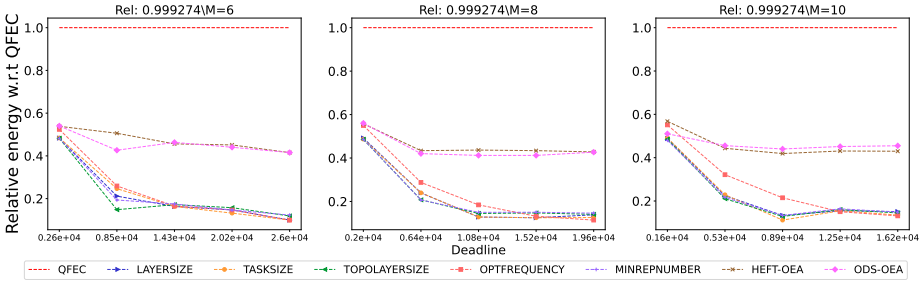

Figure 2861: Assessing the performance of different processor number on the Cycles workflow.

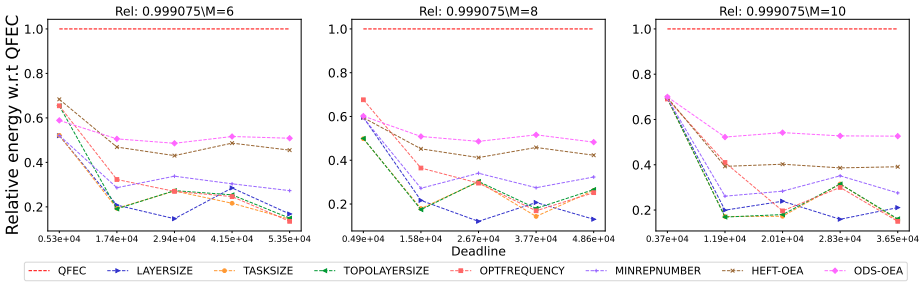

Figure 2862: Assessing the performance of different processor number on the Epigenomics workflow.

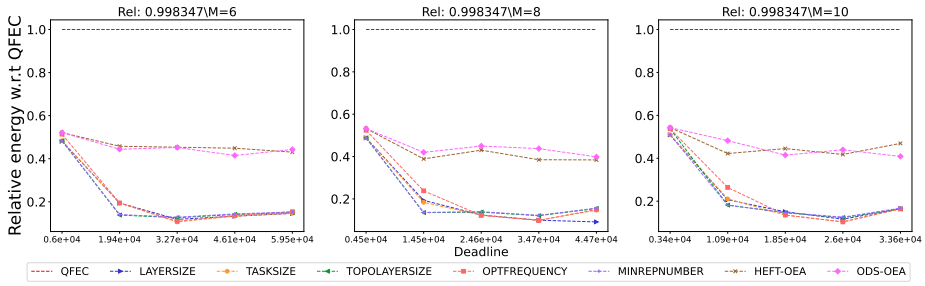

Figure 2863: Assessing the performance of different processor number on the Genome workflow.

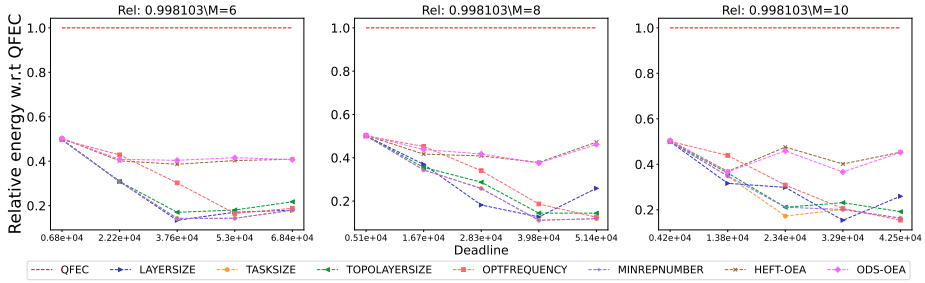

Figure 2864: Assessing the performance of different processor number on the LU.

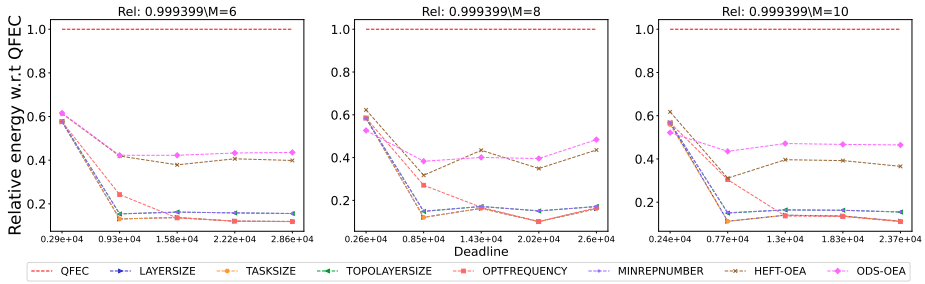

Figure 2865: Assessing the performance of different processor number on the Montage workflow.

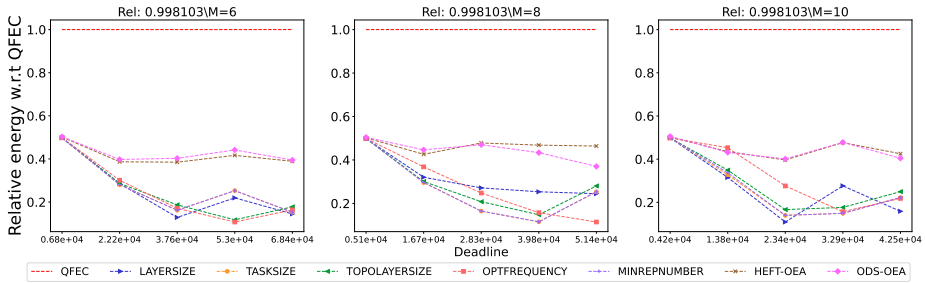

Figure 2866: Assessing the performance of different processor number on the QR.

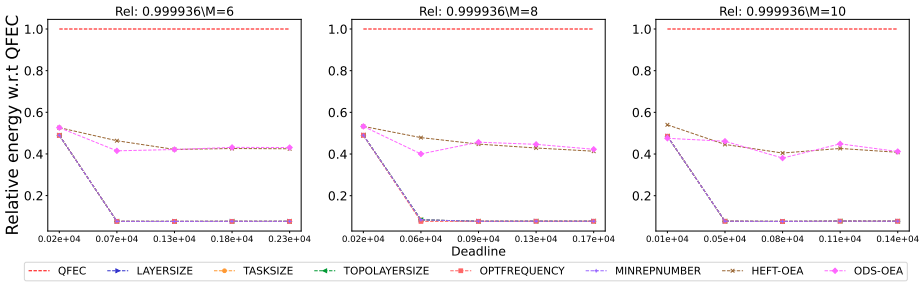

Figure 2867: Assessing the performance of different processor number on the Seismology workflow.

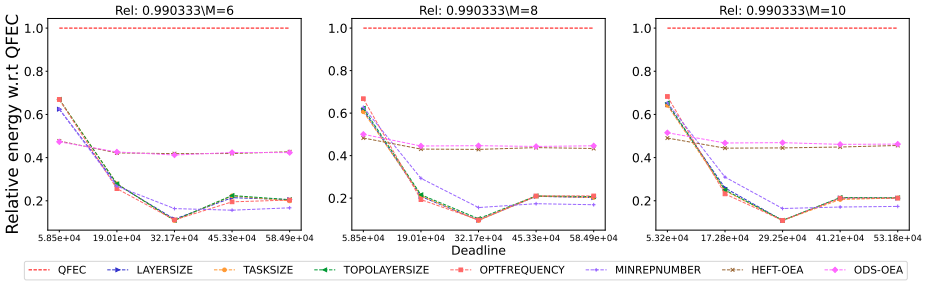

Figure 2868: Assessing the performance of different processor number on the SoyKB workflow.

I.6.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

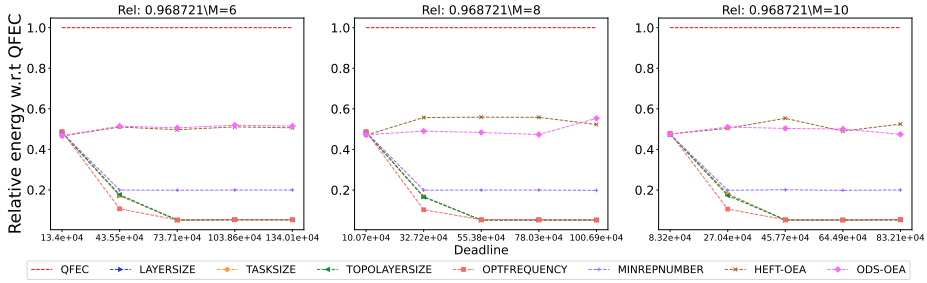

Figure 2869: Assessing the performance of different processor number on the BLAST workflow.

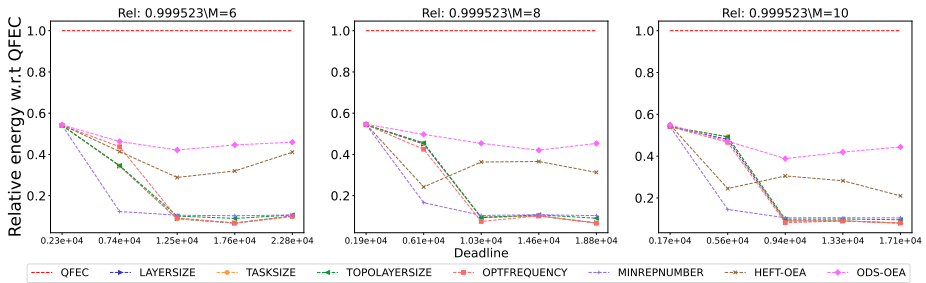

Figure 2870: Assessing the performance of different processor number on the BWA workflow.

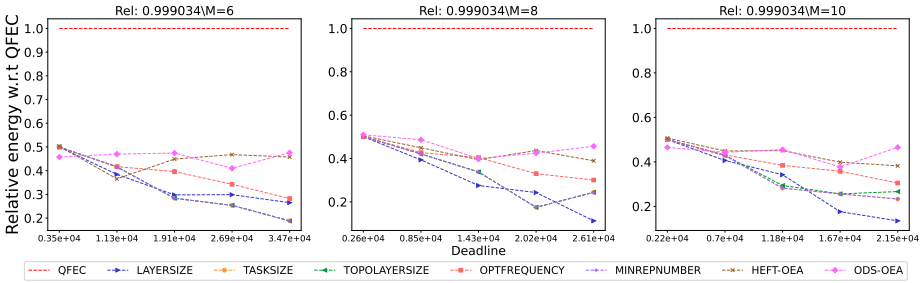

Figure 2871: Assessing the performance of different processor number on the Cholesky workflow.

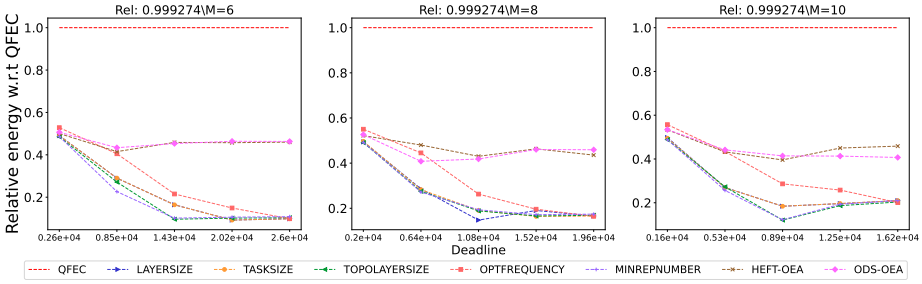

Figure 2872: Assessing the performance of different processor number on the Cycles workflow.

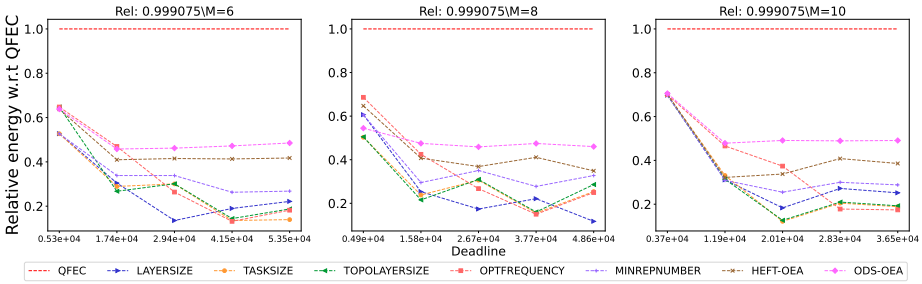

Figure 2873: Assessing the performance of different processor number on the Epigenomics workflow.

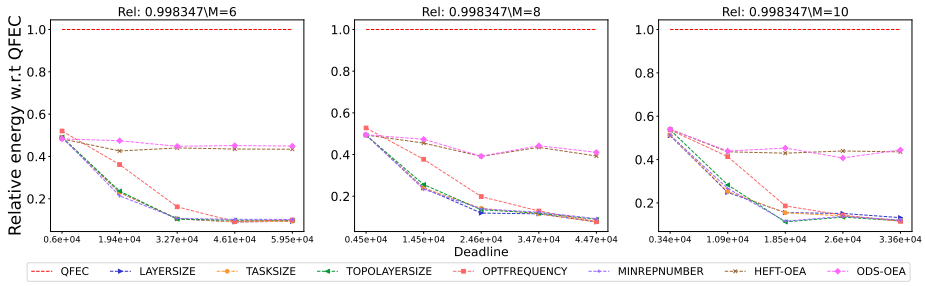

Figure 2874: Assessing the performance of different processor number on the Genome workflow.

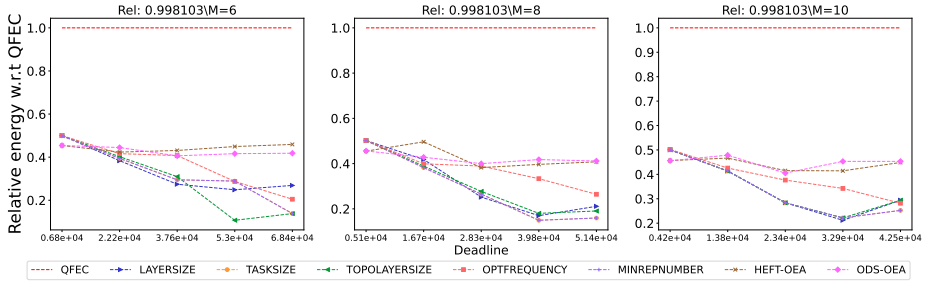

Figure 2875: Assessing the performance of different processor number on the LU.

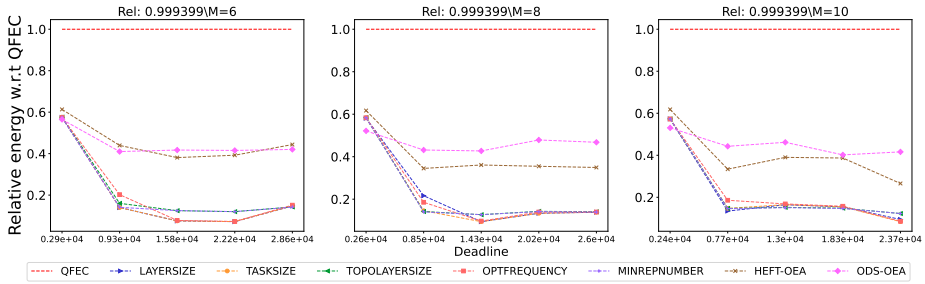

Figure 2876: Assessing the performance of different processor number on the Montage workflow.

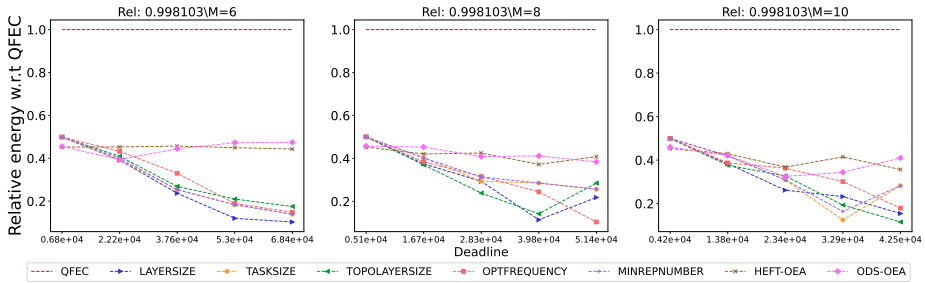

Figure 2877: Assessing the performance of different processor number on the QR.

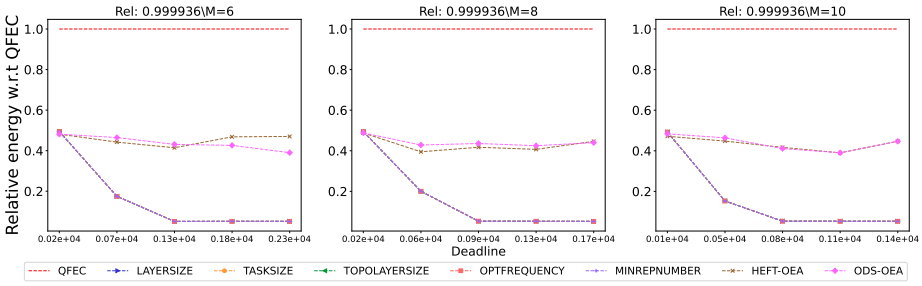

Figure 2878: Assessing the performance of different processor number on the Seismology workflow.

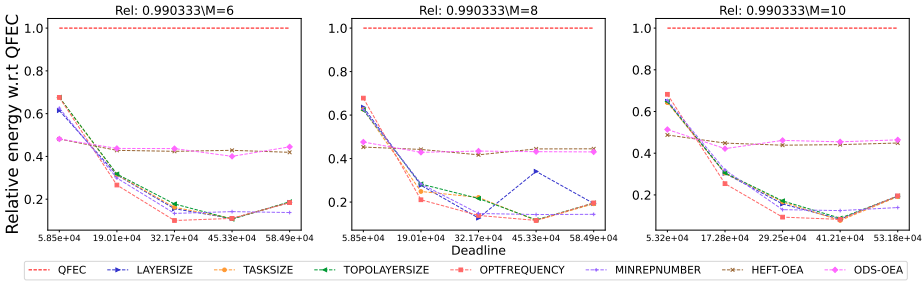

Figure 2879: Assessing the performance of different processor number on the SoyKB workflow.

I.6.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

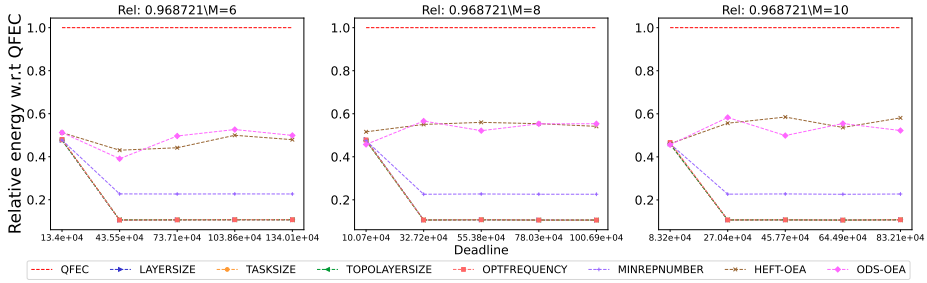

Figure 2880: Assessing the performance of different processor number on the BLAST workflow.

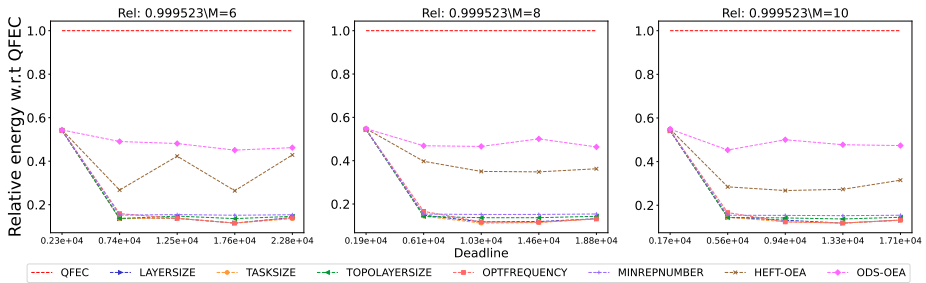

Figure 2881: Assessing the performance of different processor number on the BWA workflow.

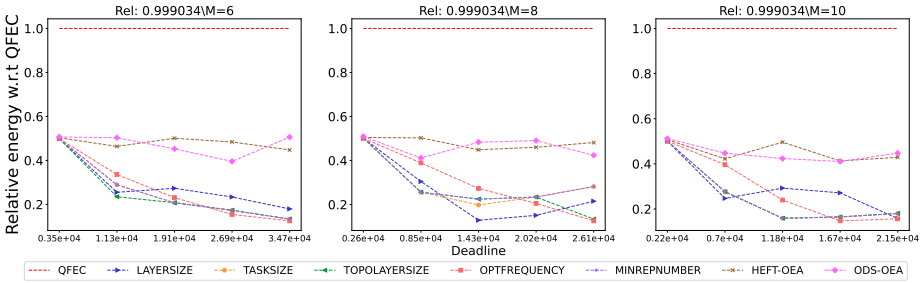

Figure 2882: Assessing the performance of different processor number on the Cholesky workflow.

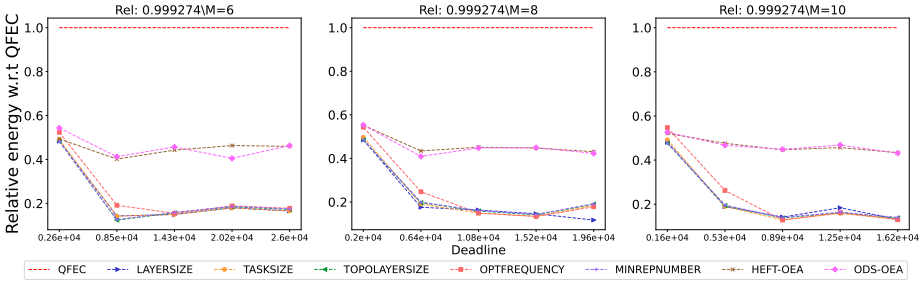

Figure 2883: Assessing the performance of different processor number on the Cycles workflow.

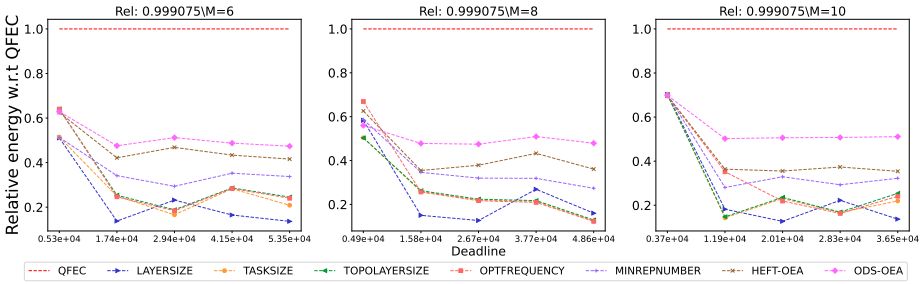

Figure 2884: Assessing the performance of different processor number on the Epigenomics workflow.

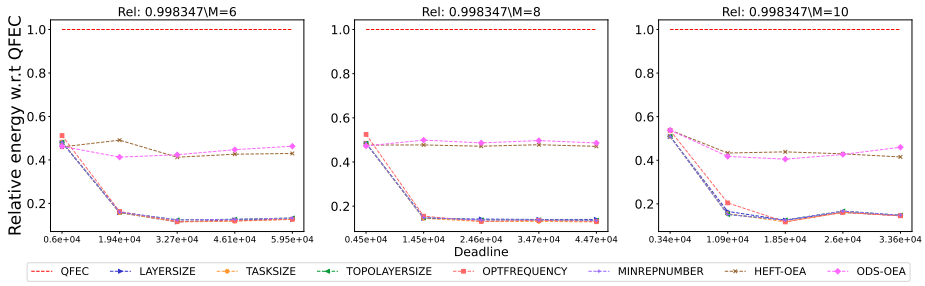

Figure 2885: Assessing the performance of different processor number on the Genome workflow.

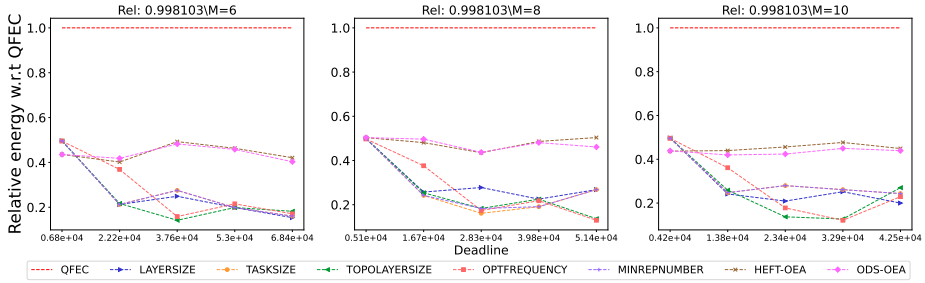

Figure 2886: Assessing the performance of different processor number on the LU.

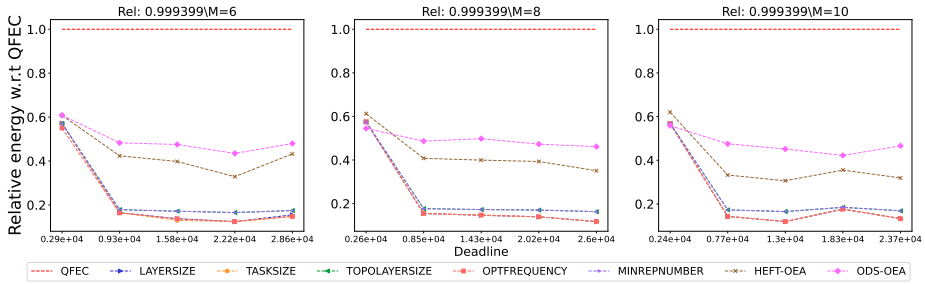

Figure 2887: Assessing the performance of different processor number on the Montage workflow.

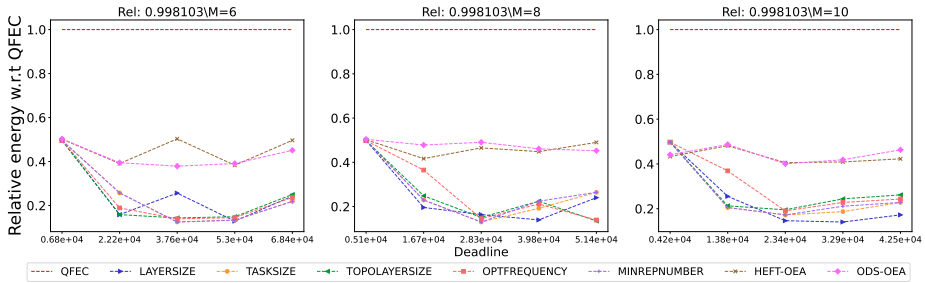

Figure 2888: Assessing the performance of different processor number on the QR.

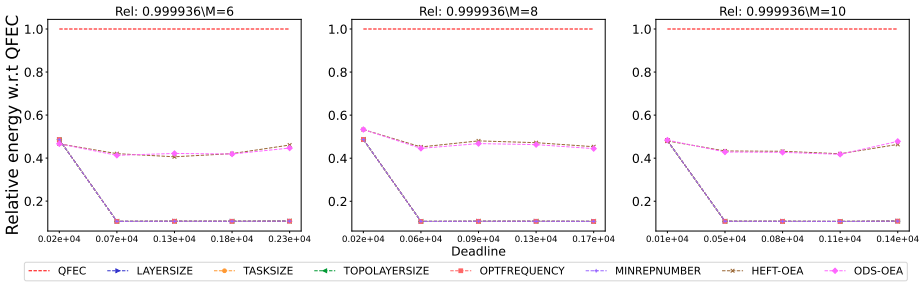

Figure 2889: Assessing the performance of different processor number on the Seismology workflow.

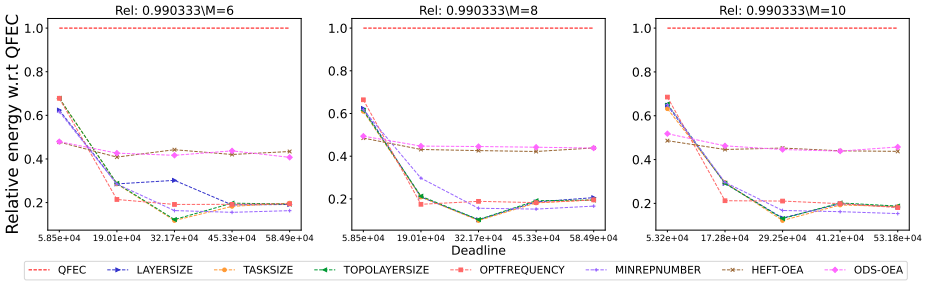

Figure 2890: Assessing the performance of different processor number on the SoyKB workflow.

I.6.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

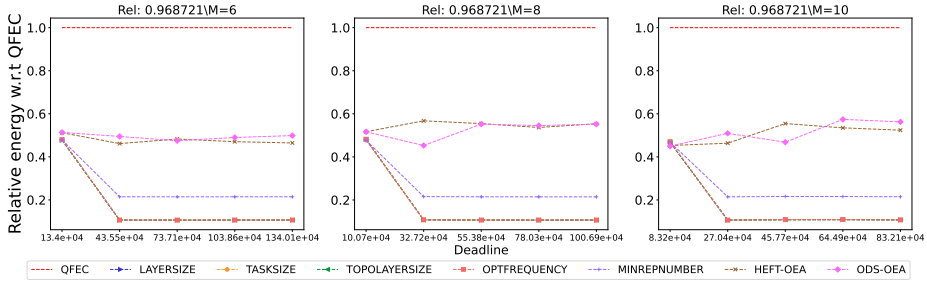

Figure 2891: Assessing the performance of different processor number on the BLAST workflow.

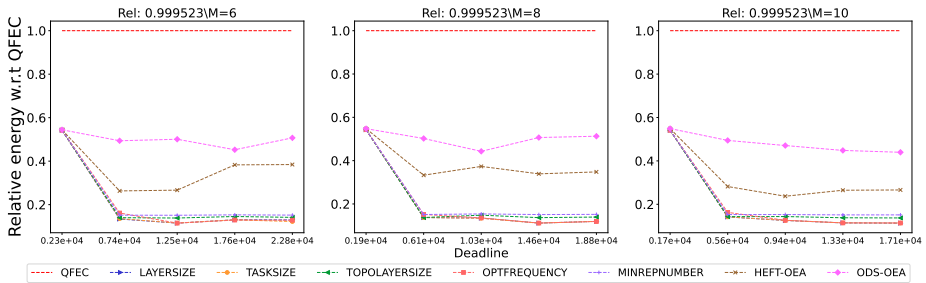

Figure 2892: Assessing the performance of different processor number on the BWA workflow.

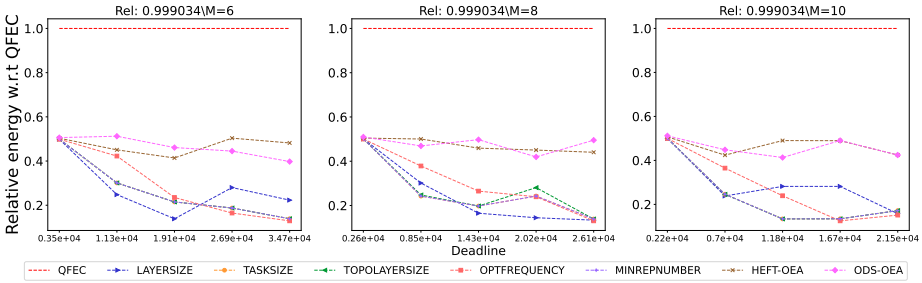

Figure 2893: Assessing the performance of different processor number on the Cholesky workflow.

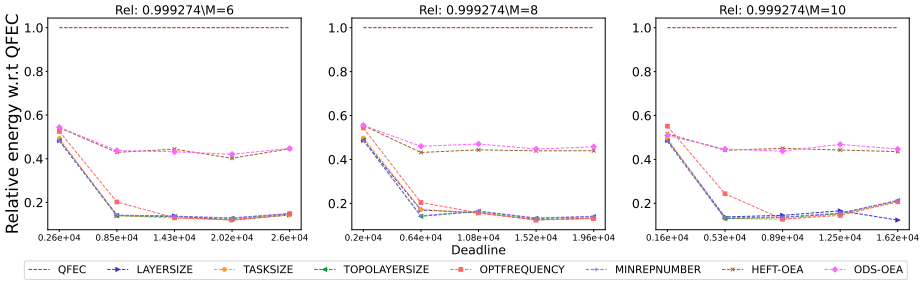

Figure 2894: Assessing the performance of different processor number on the Cycles workflow.

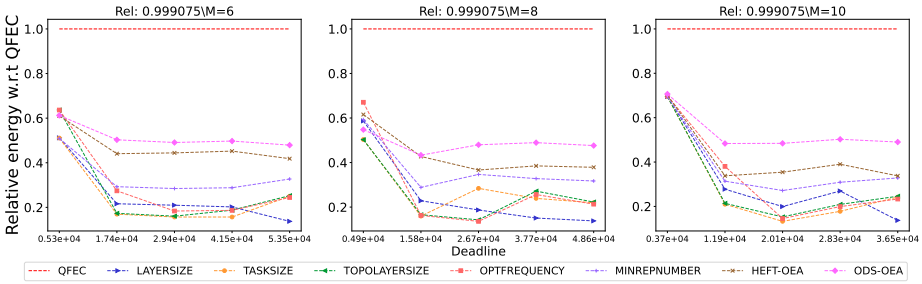

Figure 2895: Assessing the performance of different processor number on the Epigenomics workflow.

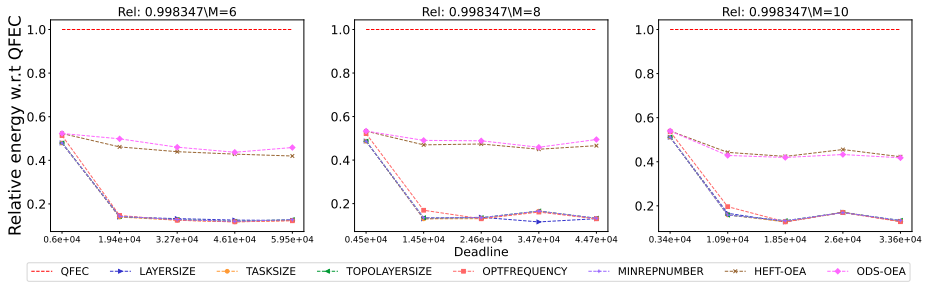

Figure 2896: Assessing the performance of different processor number on the Genome workflow.

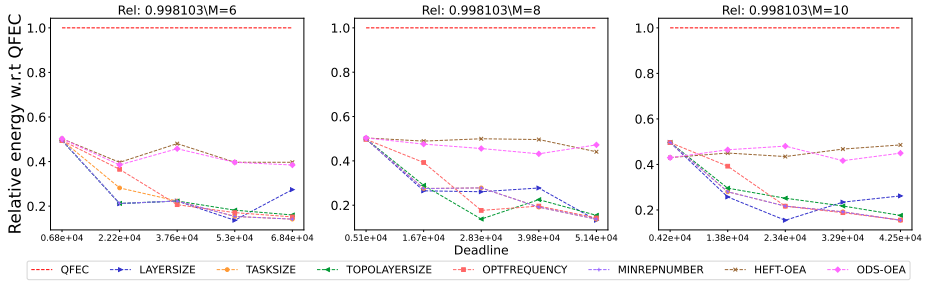

Figure 2897: Assessing the performance of different processor number on the LU.

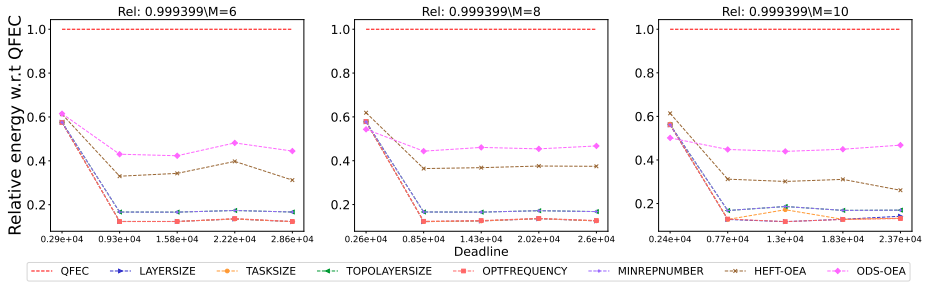

Figure 2898: Assessing the performance of different processor number on the Montage workflow.

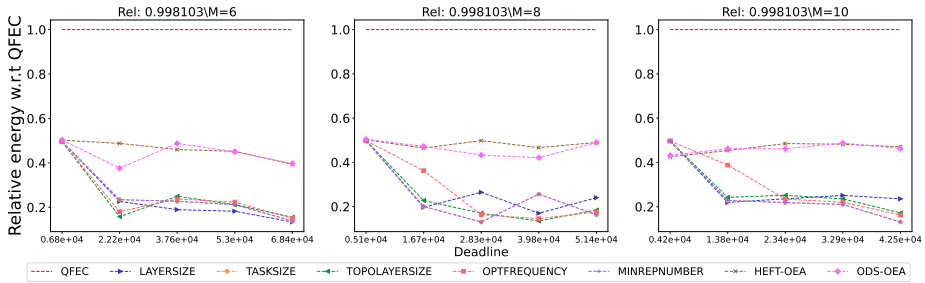

Figure 2899: Assessing the performance of different processor number on the QR.

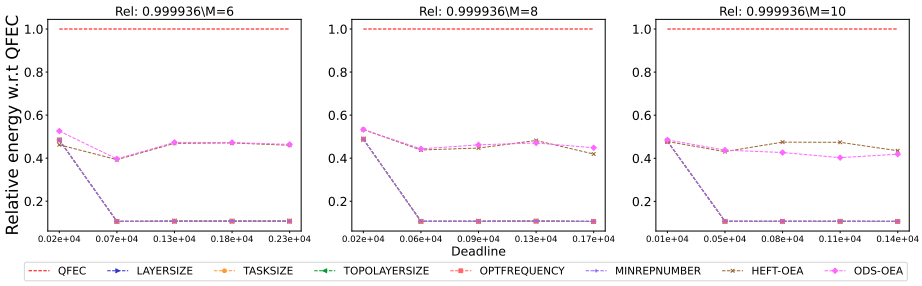

Figure 2900: Assessing the performance of different processor number on the Seismology workflow.

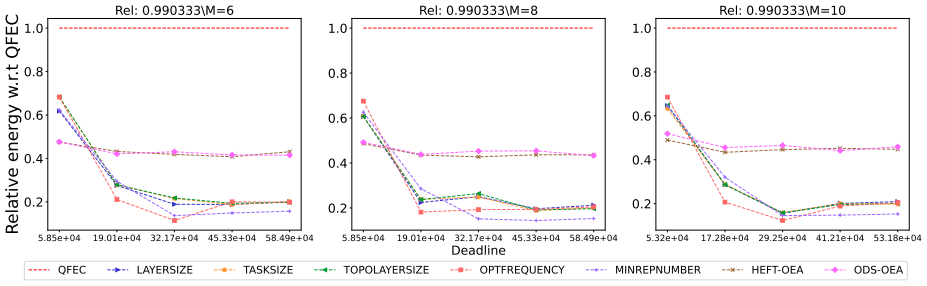

Figure 2901: Assessing the performance of different processor number on the SoyKB workflow.

I.7 $BC/WC = 0.7$

I.7.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

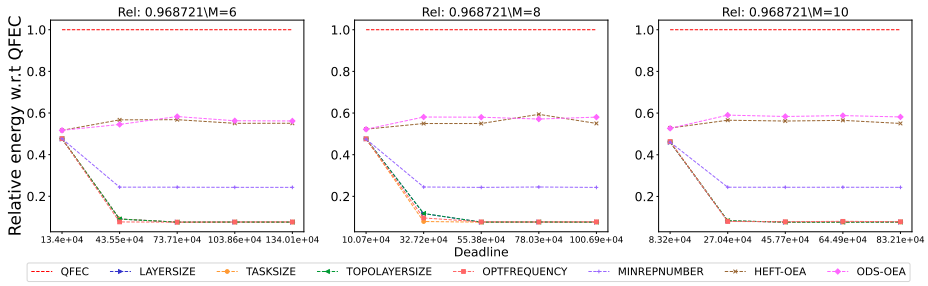

Figure 2902: Assessing the performance of different processor number on the BLAST workflow.

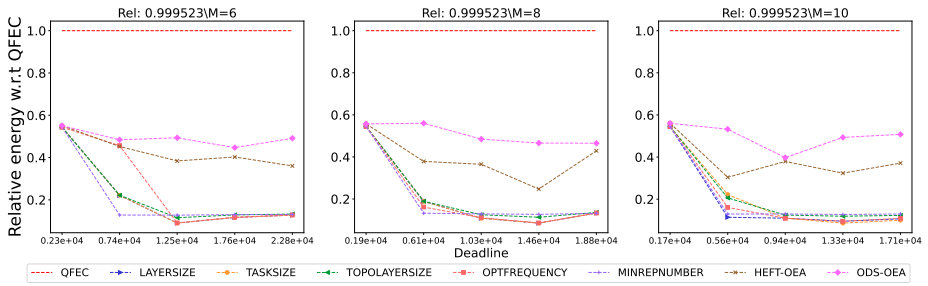

Figure 2903: Assessing the performance of different processor number on the BWA workflow.

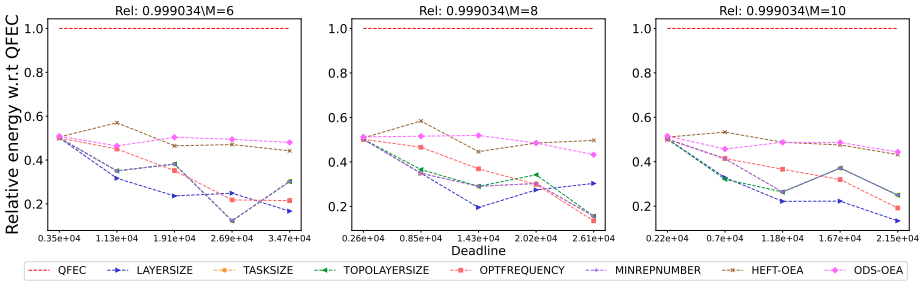

Figure 2904: Assessing the performance of different processor number on the Cholesky workflow.

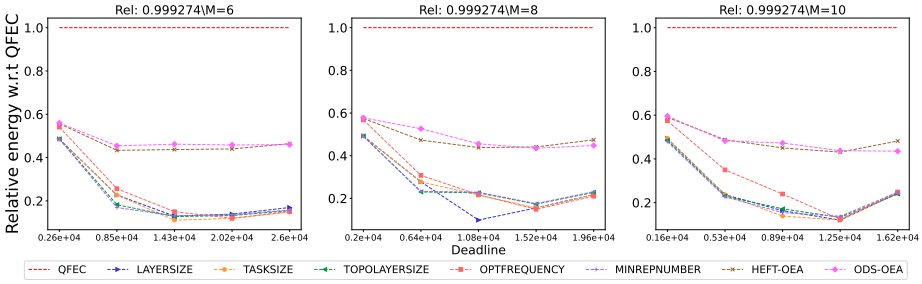

Figure 2905: Assessing the performance of different processor number on the Cycles workflow.

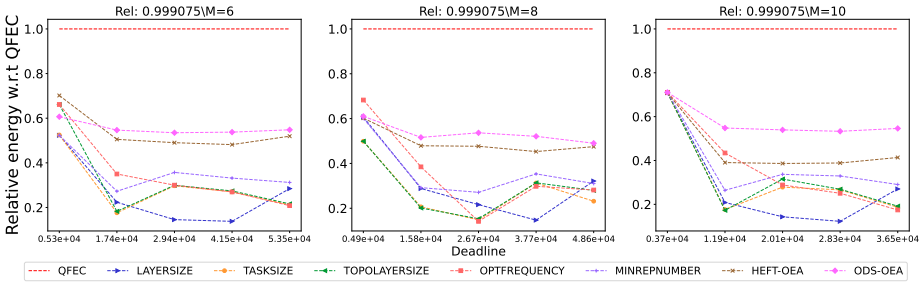

Figure 2906: Assessing the performance of different processor number on the Epigenomics workflow.

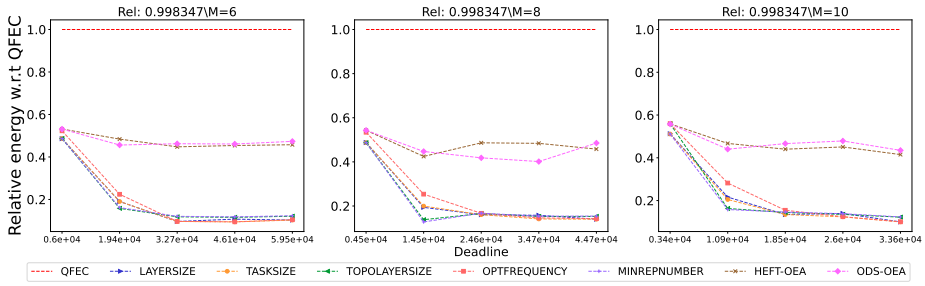

Figure 2907: Assessing the performance of different processor number on the Genome workflow.

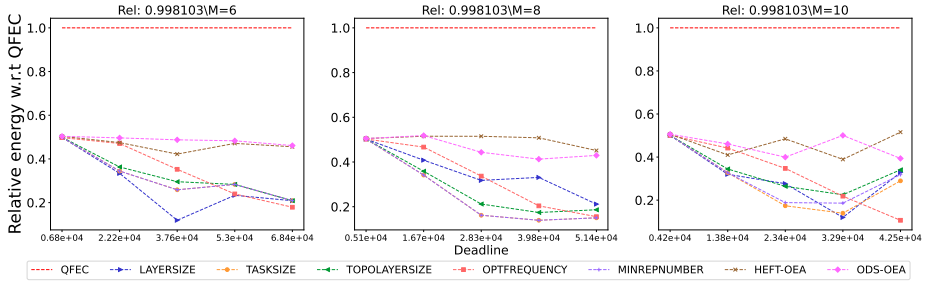

Figure 2908: Assessing the performance of different processor number on the LU.

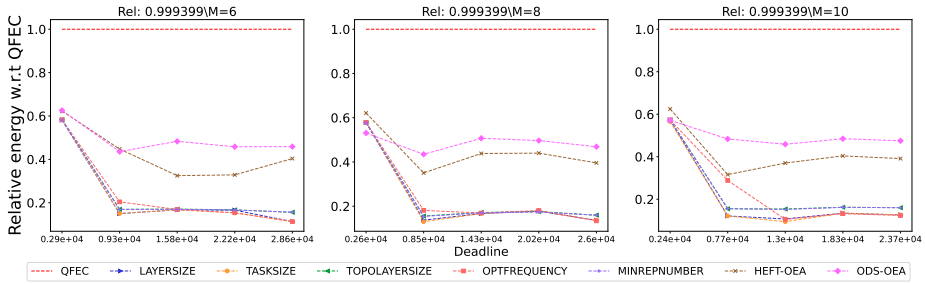

Figure 2909: Assessing the performance of different processor number on the Montage workflow.

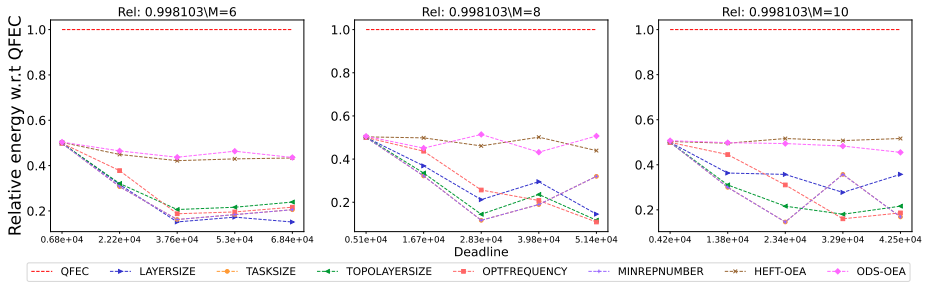

Figure 2910: Assessing the performance of different processor number on the QR.

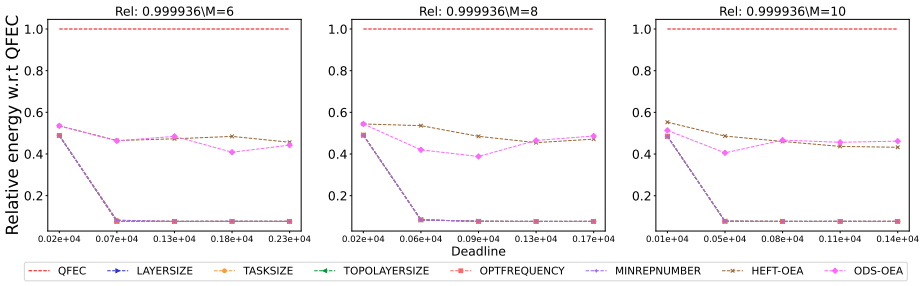

Figure 2911: Assessing the performance of different processor number on the Seismology workflow.

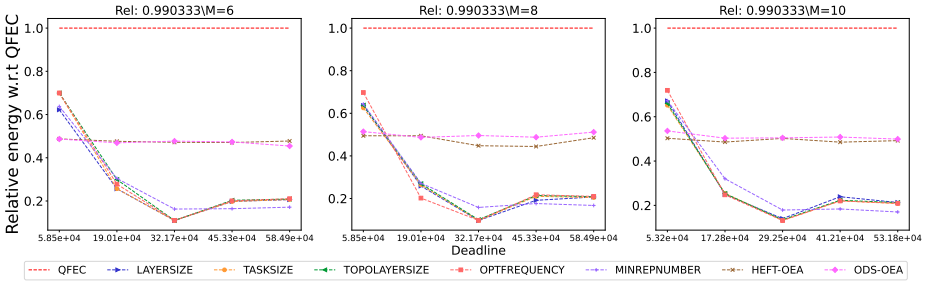

Figure 2912: Assessing the performance of different processor number on the SoyKB workflow.

I.7.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

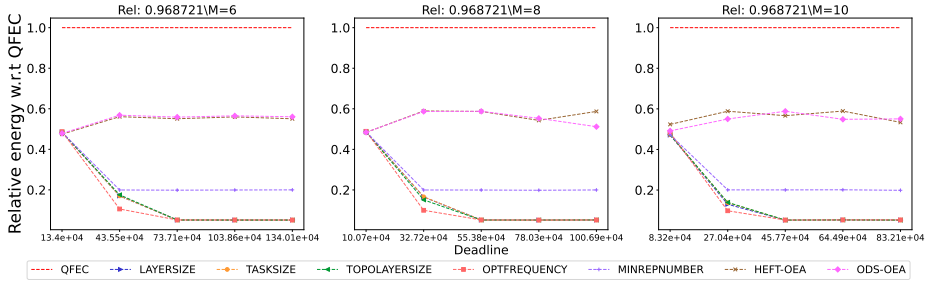

Figure 2913: Assessing the performance of different processor number on the BLAST workflow.

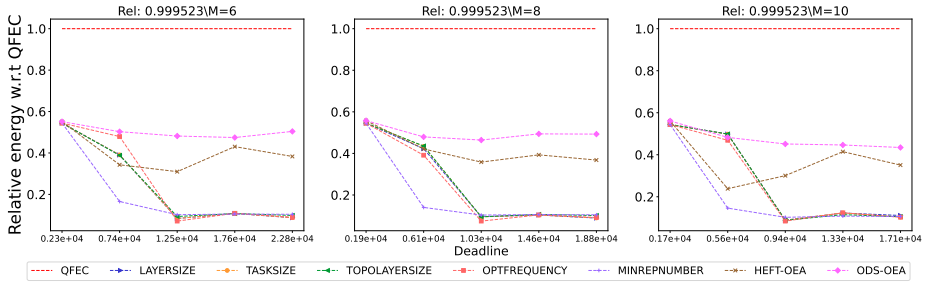

Figure 2914: Assessing the performance of different processor number on the BWA workflow.

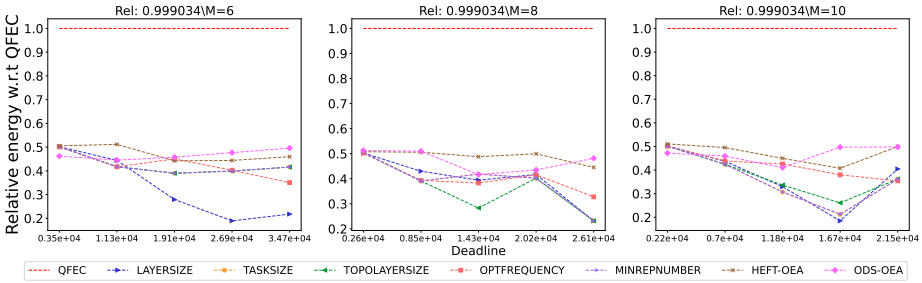

Figure 2915: Assessing the performance of different processor number on the Cholesky workflow.

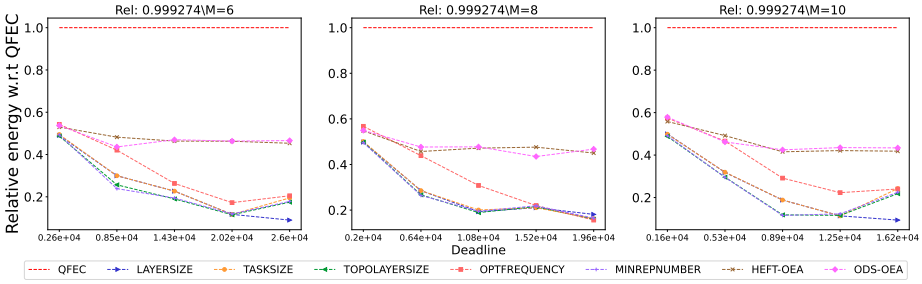

Figure 2916: Assessing the performance of different processor number on the Cycles workflow.

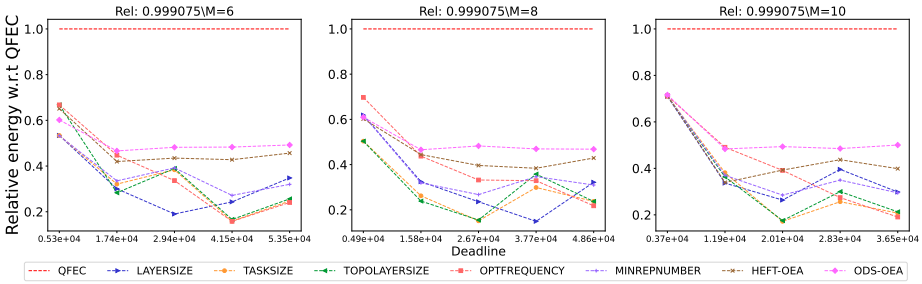

Figure 2917: Assessing the performance of different processor number on the Epigenomics workflow.

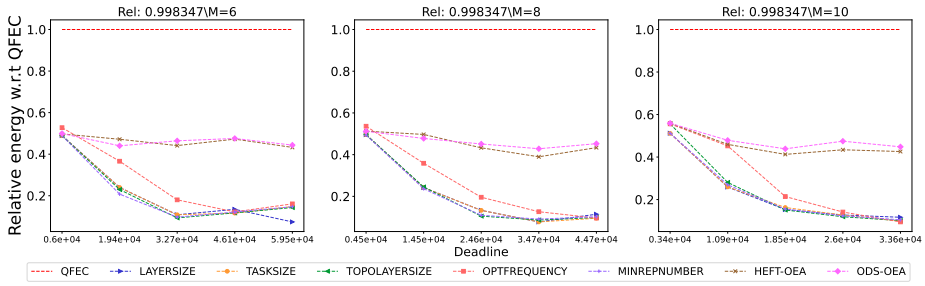

Figure 2918: Assessing the performance of different processor number on the Genome workflow.

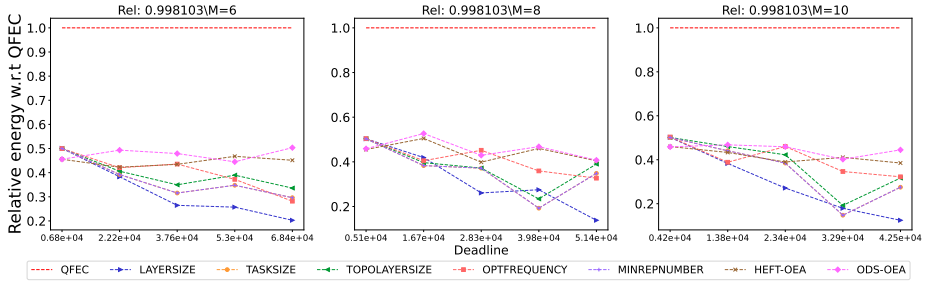

Figure 2919: Assessing the performance of different processor number on the LU.

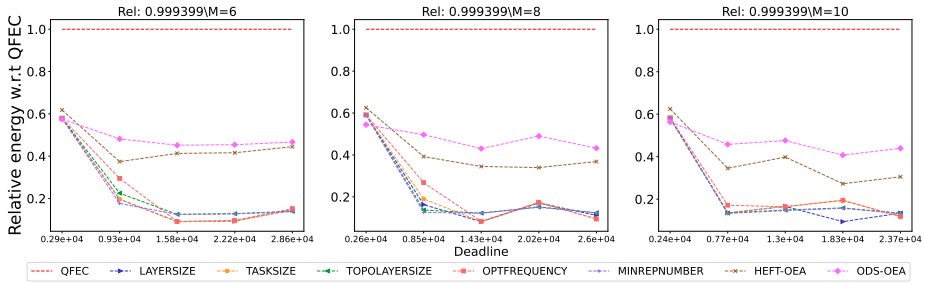

Figure 2920: Assessing the performance of different processor number on the Montage workflow.

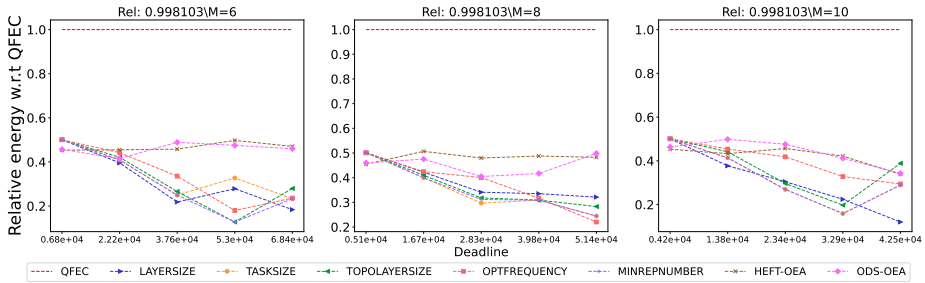

Figure 2921: Assessing the performance of different processor number on the QR.

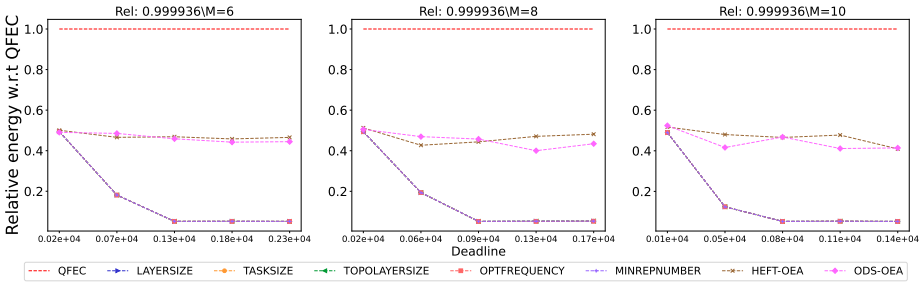

Figure 2922: Assessing the performance of different processor number on the Seismology workflow.

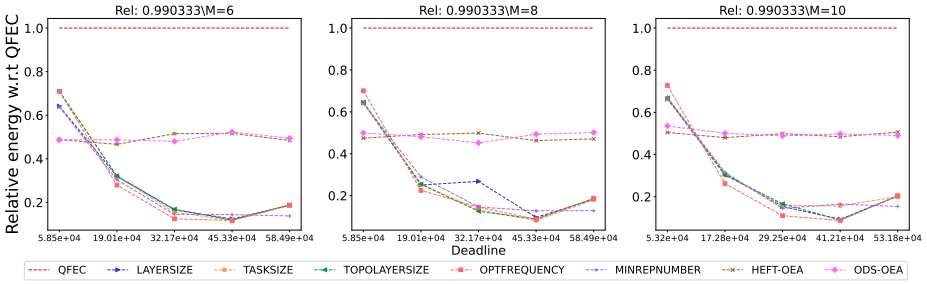

Figure 2923: Assessing the performance of different processor number on the SoyKB workflow.

I.7.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

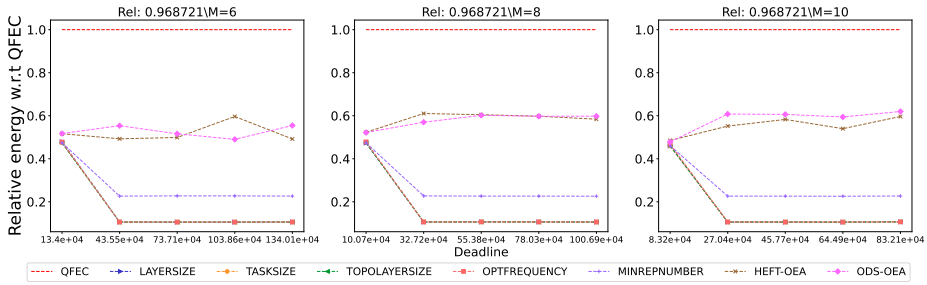

Figure 2924: Assessing the performance of different processor number on the BLAST workflow.

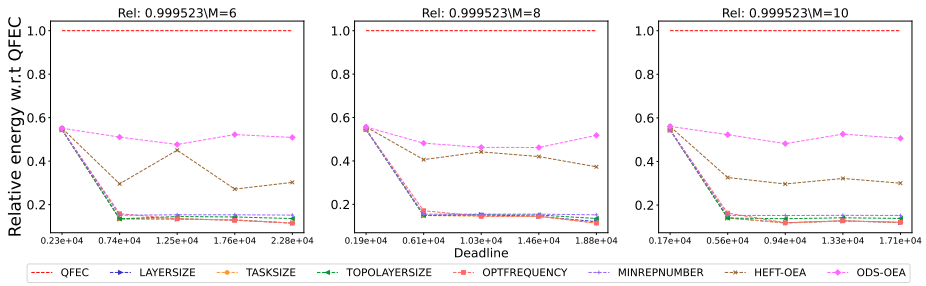

Figure 2925: Assessing the performance of different processor number on the BWA workflow.

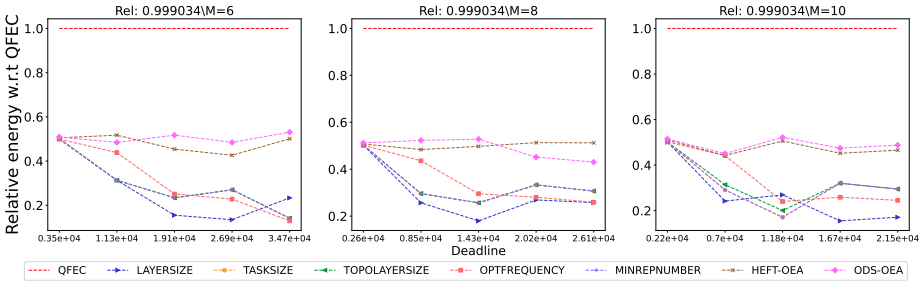

Figure 2926: Assessing the performance of different processor number on the Cholesky workflow.

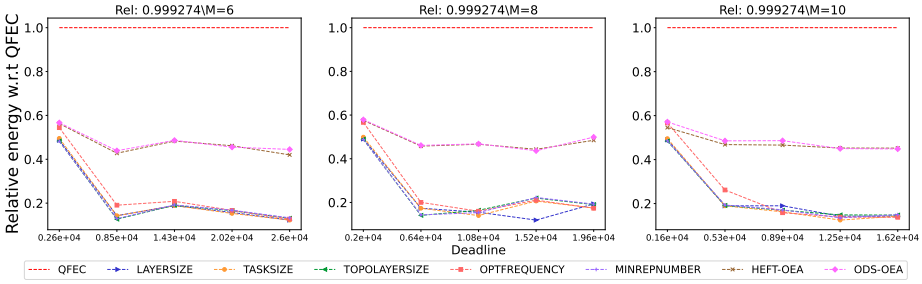

Figure 2927: Assessing the performance of different processor number on the Cycles workflow.

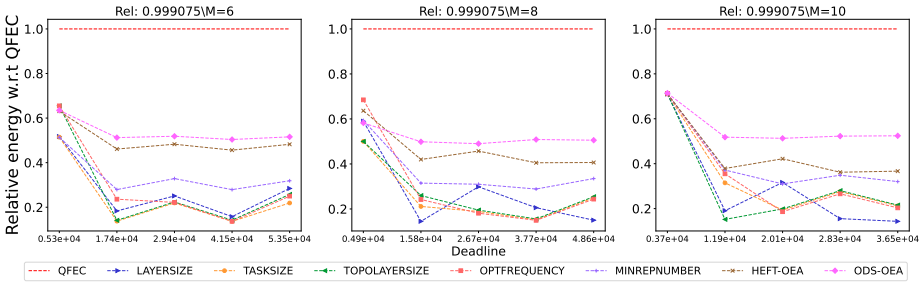

Figure 2928: Assessing the performance of different processor number on the Epigenomics workflow.

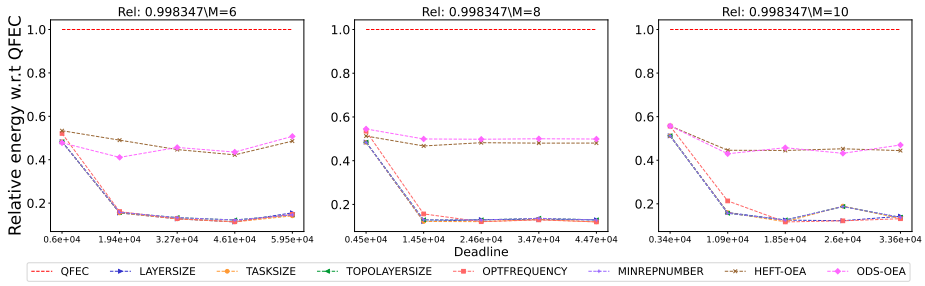

Figure 2929: Assessing the performance of different processor number on the Genome workflow.

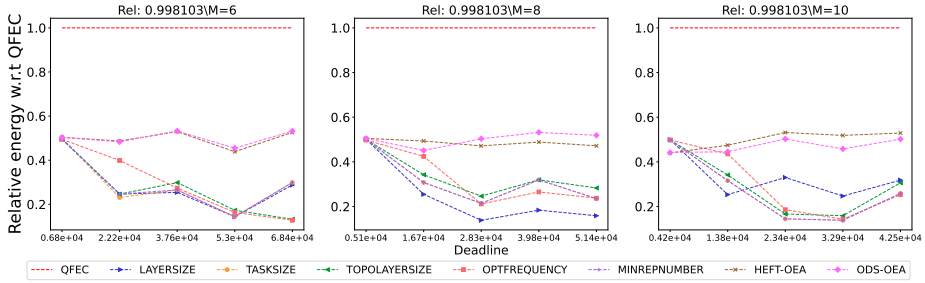

Figure 2930: Assessing the performance of different processor number on the LU.

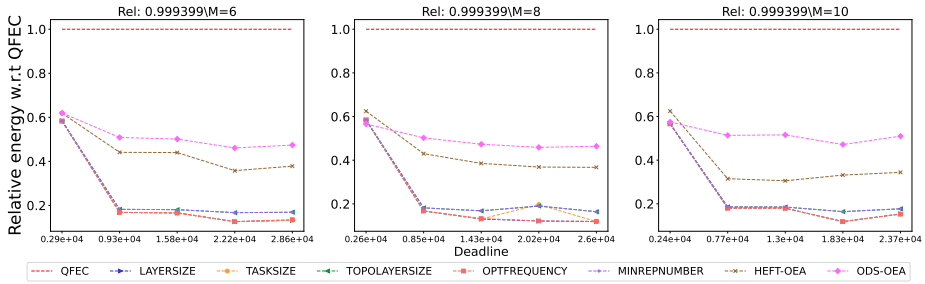

Figure 2931: Assessing the performance of different processor number on the Montage workflow.

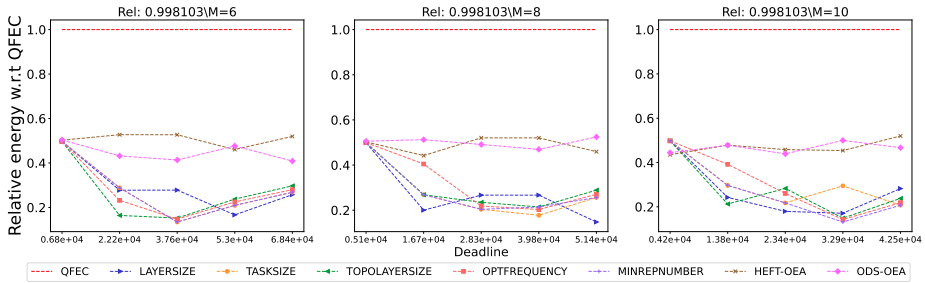

Figure 2932: Assessing the performance of different processor number on the QR.

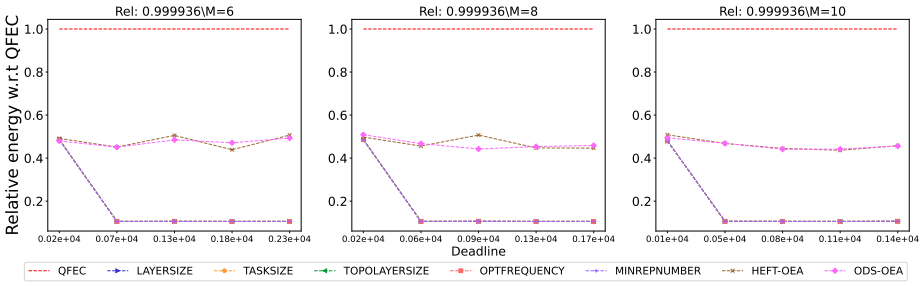

Figure 2933: Assessing the performance of different processor number on the Seismology workflow.

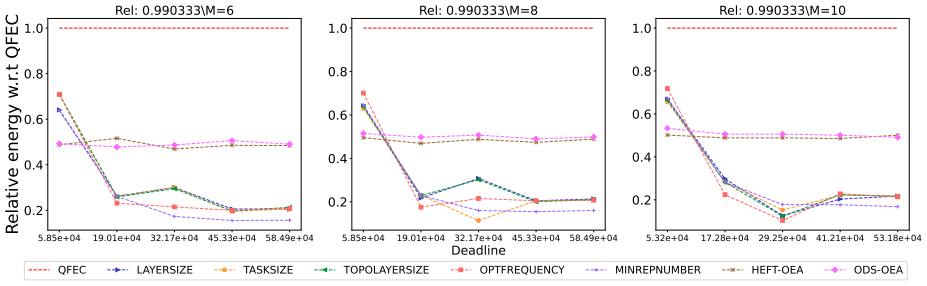

Figure 2934: Assessing the performance of different processor number on the SoyKB workflow.

I.7.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

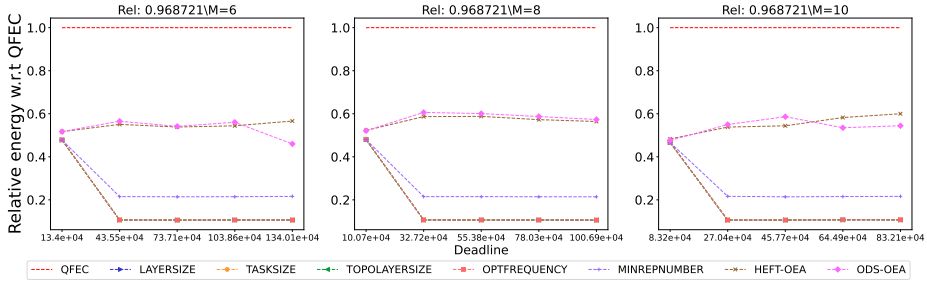

Figure 2935: Assessing the performance of different processor number on the BLAST workflow.

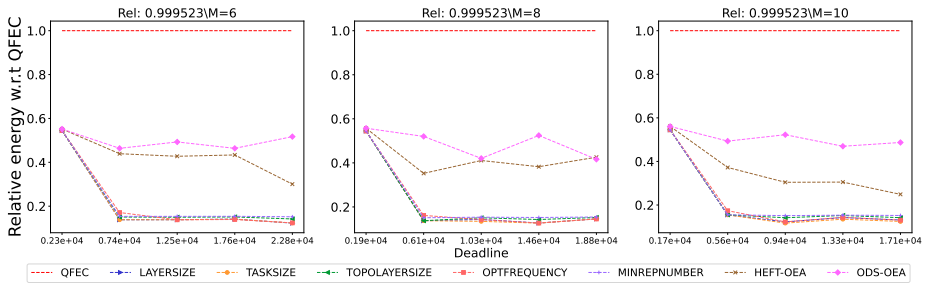

Figure 2936: Assessing the performance of different processor number on the BWA workflow.

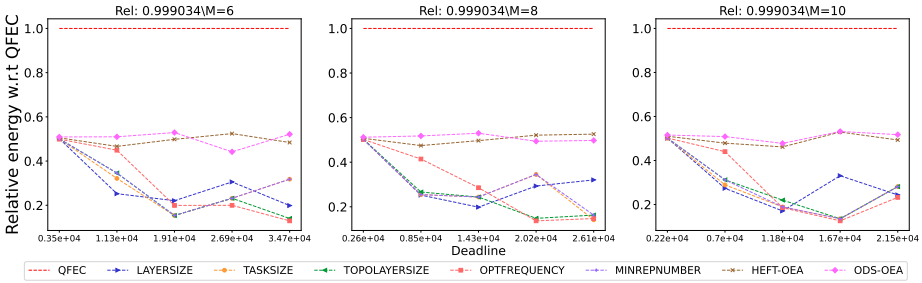

Figure 2937: Assessing the performance of different processor number on the Cholesky workflow.

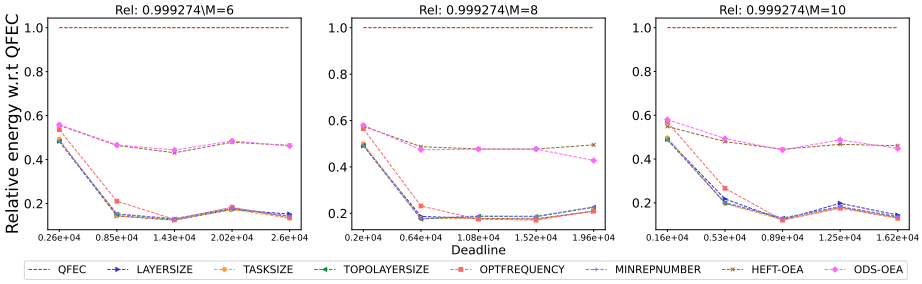

Figure 2938: Assessing the performance of different processor number on the Cycles workflow.

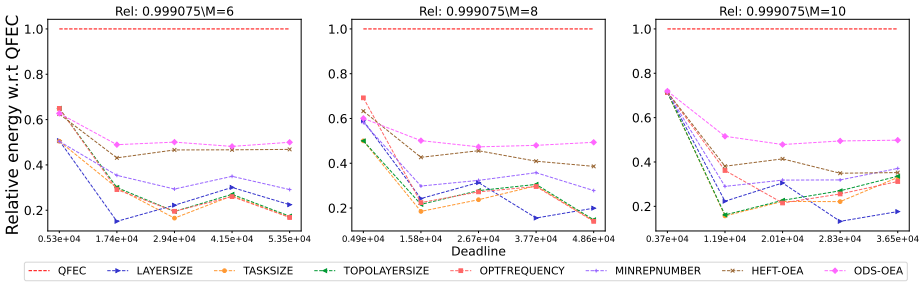

Figure 2939: Assessing the performance of different processor number on the Epigenomics workflow.

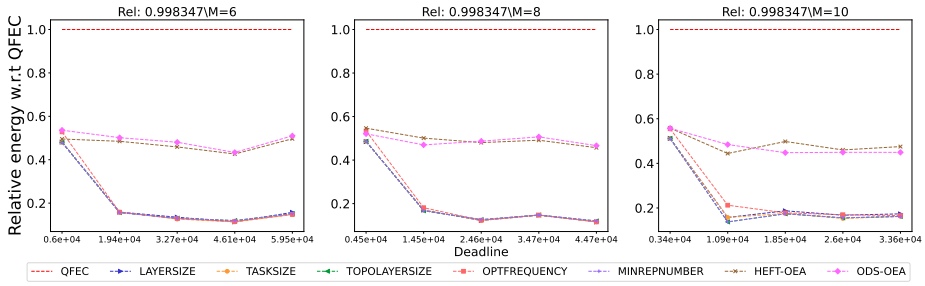

Figure 2940: Assessing the performance of different processor number on the Genome workflow.

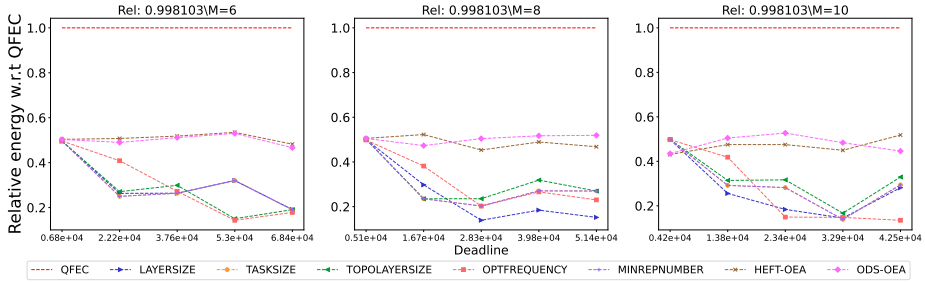

Figure 2941: Assessing the performance of different processor number on the LU.

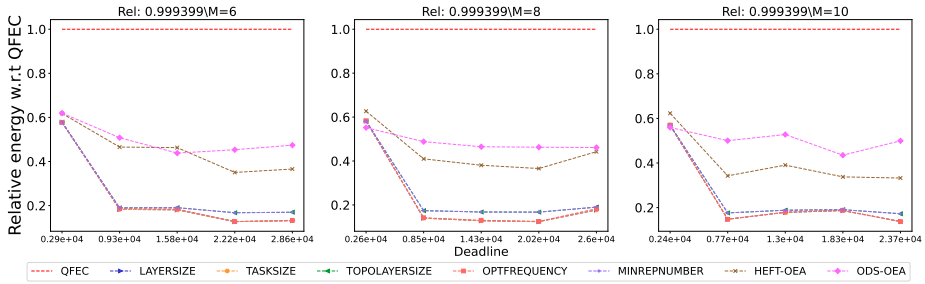

Figure 2942: Assessing the performance of different processor number on the Montage workflow.

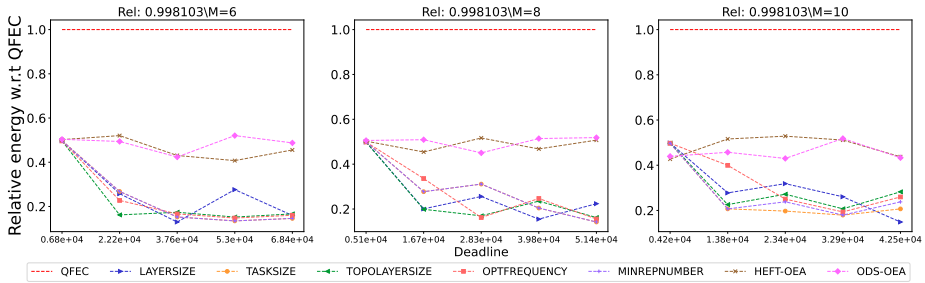

Figure 2943: Assessing the performance of different processor number on the QR.

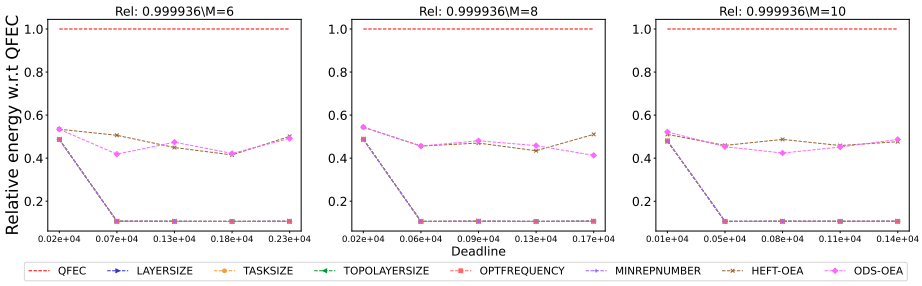

Figure 2944: Assessing the performance of different processor number on the Seismology workflow.

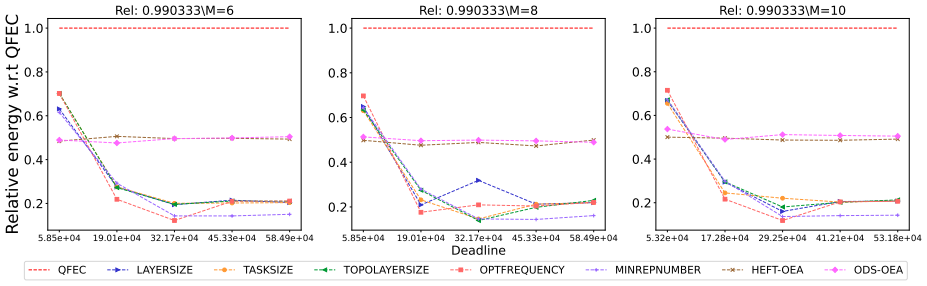

Figure 2945: Assessing the performance of different processor number on the SoyKB workflow.

I.8 $BC/WC = 0.8$

I.8.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

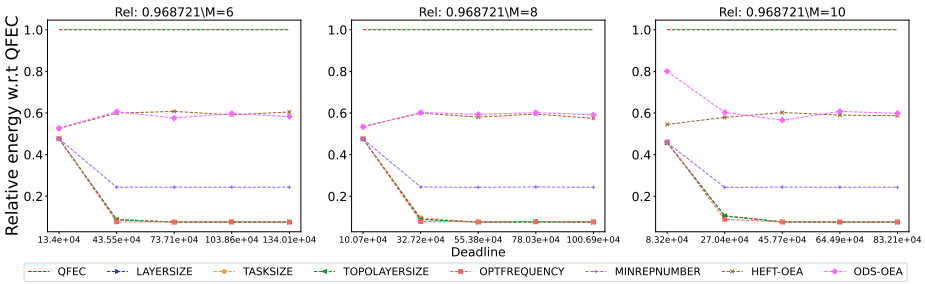

Figure 2946: Assessing the performance of different processor number on the BLAST workflow.

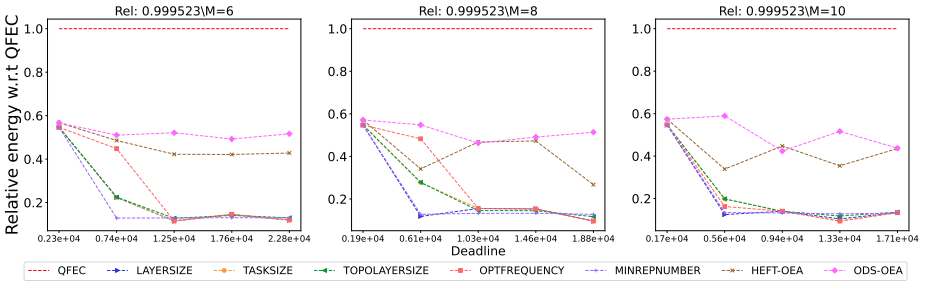

Figure 2947: Assessing the performance of different processor number on the BWA workflow.

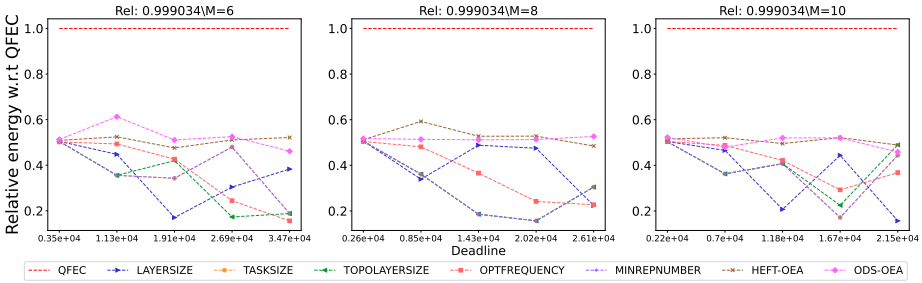

Figure 2948: Assessing the performance of different processor number on the Cholesky workflow.

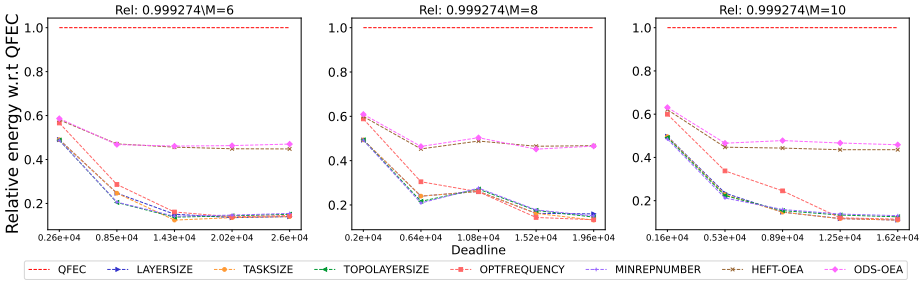

Figure 2949: Assessing the performance of different processor number on the Cycles workflow.

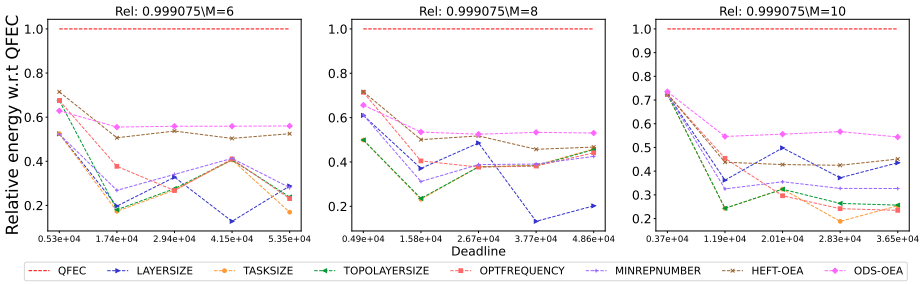

Figure 2950: Assessing the performance of different processor number on the Epigenomics workflow.

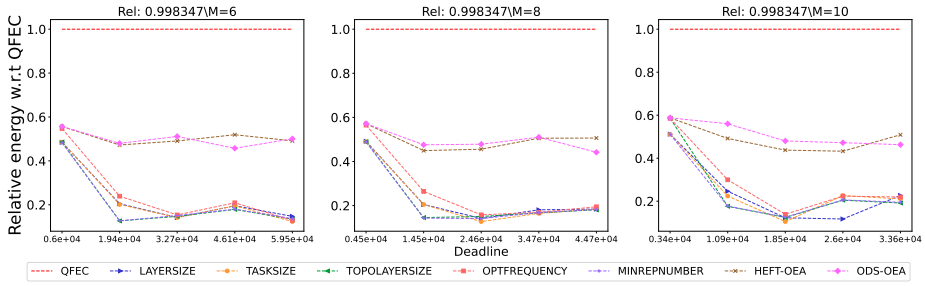

Figure 2951: Assessing the performance of different processor number on the Genome workflow.

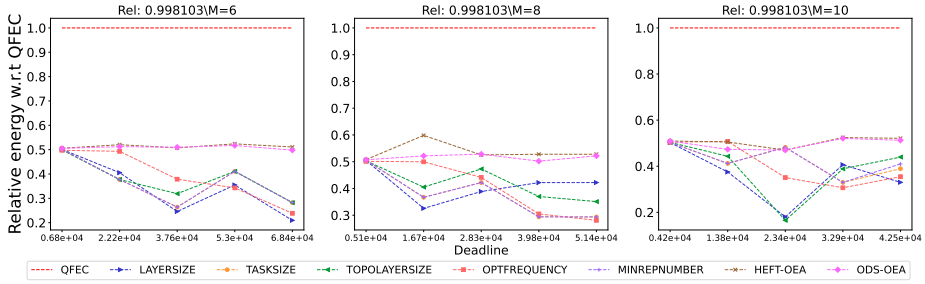

Figure 2952: Assessing the performance of different processor number on the LU.

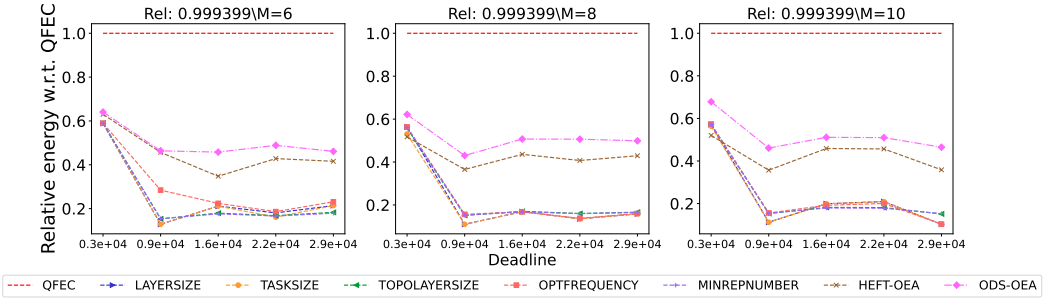

Figure 2953: Assessing the performance of different processor number on the Montage workflow.

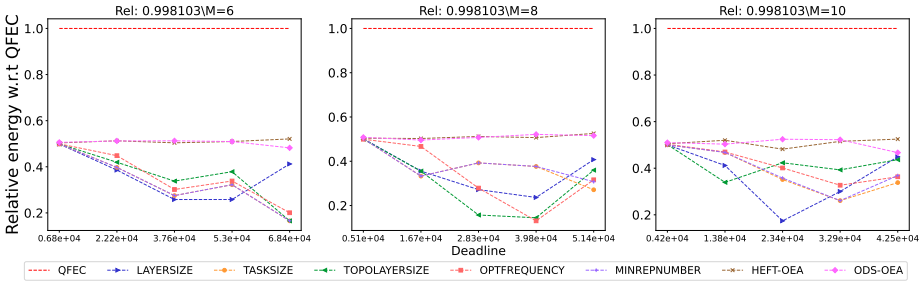

Figure 2954: Assessing the performance of different processor number on the QR.

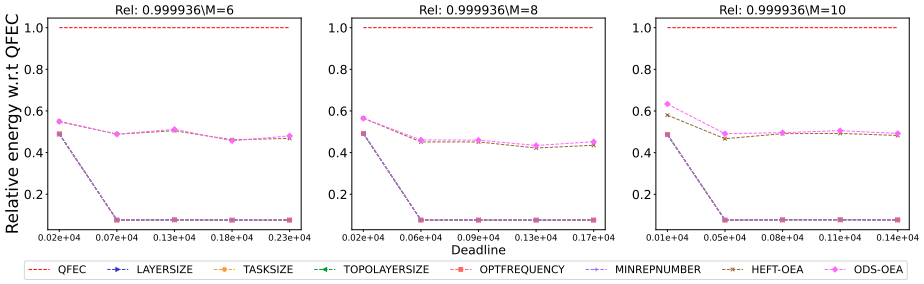

Figure 2955: Assessing the performance of different processor number on the Seismology workflow.

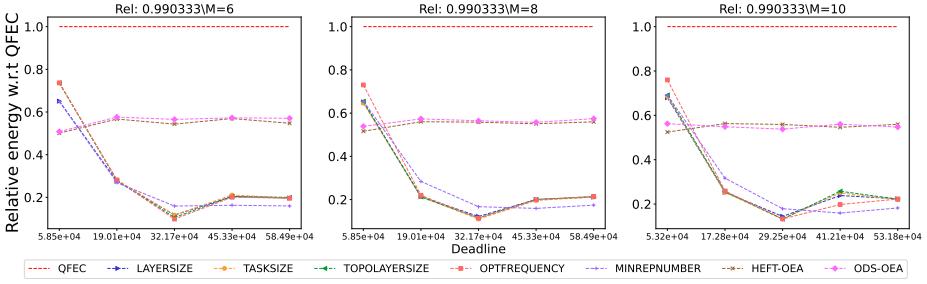

Figure 2956: Assessing the performance of different processor number on the SoyKB workflow.

I.8.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

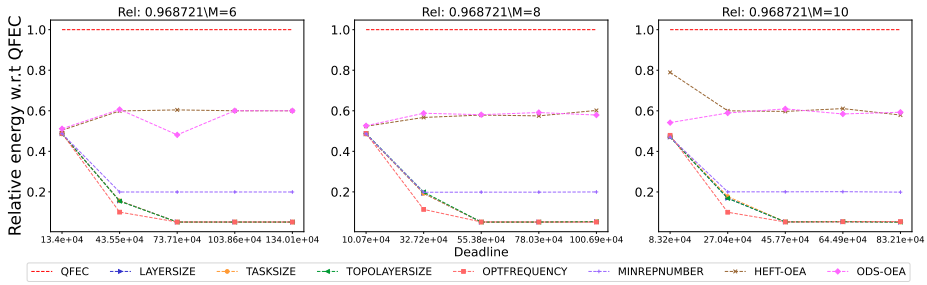

Figure 2957: Assessing the performance of different processor number on the BLAST workflow.

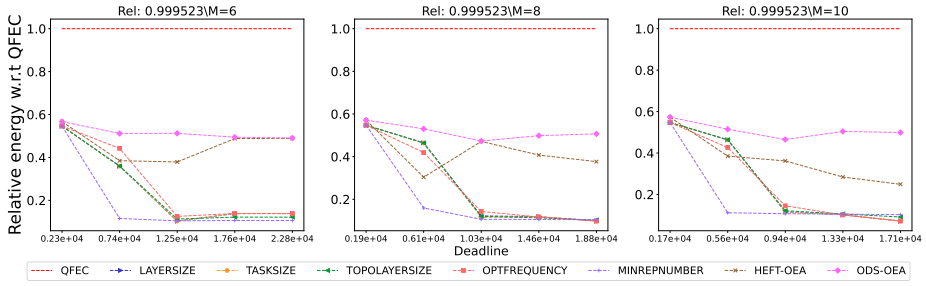

Figure 2958: Assessing the performance of different processor number on the BWA workflow.

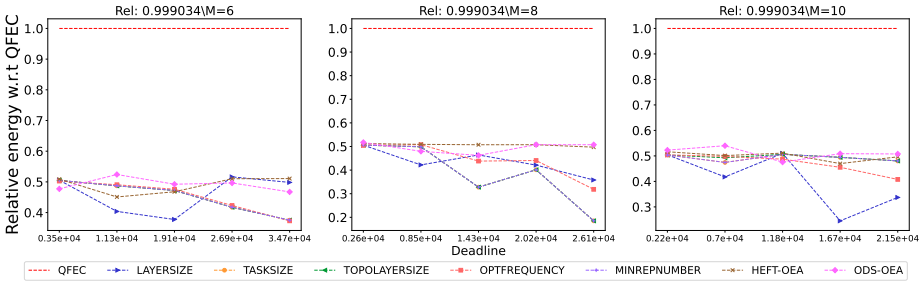

Figure 2959: Assessing the performance of different processor number on the Cholesky workflow.

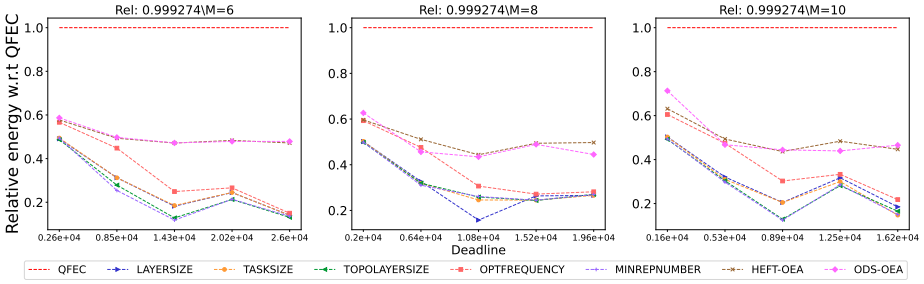

Figure 2960: Assessing the performance of different processor number on the Cycles workflow.

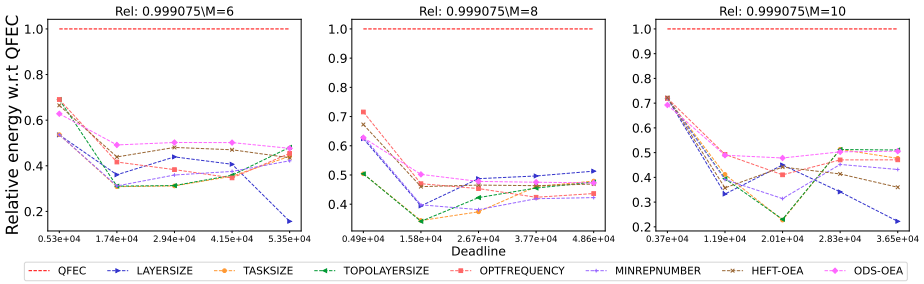

Figure 2961: Assessing the performance of different processor number on the Epigenomics workflow.

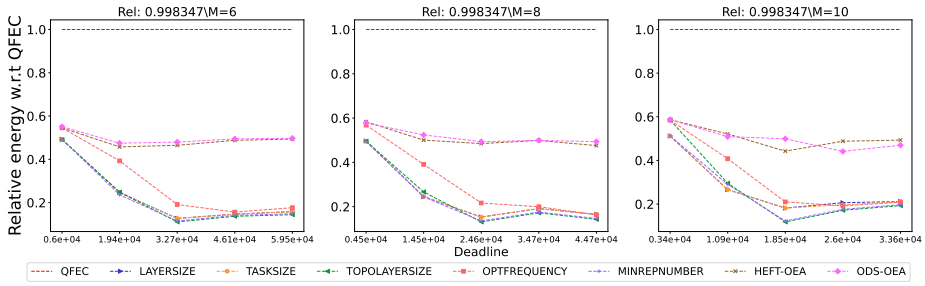

Figure 2962: Assessing the performance of different processor number on the Genome workflow.

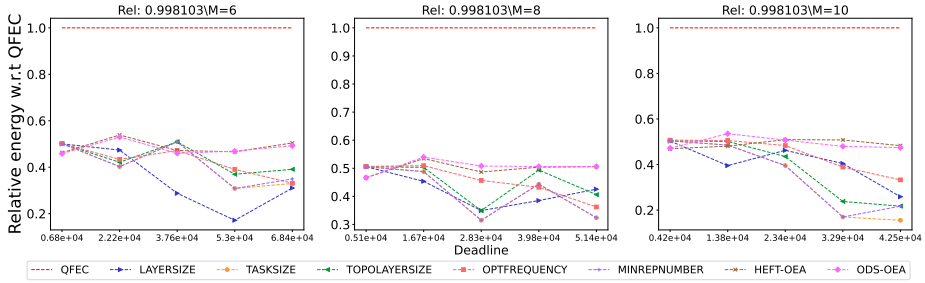

Figure 2963: Assessing the performance of different processor number on the LU.

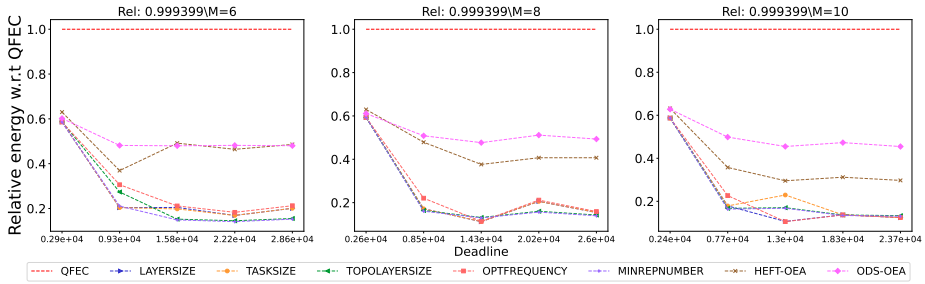

Figure 2964: Assessing the performance of different processor number on the Montage workflow.

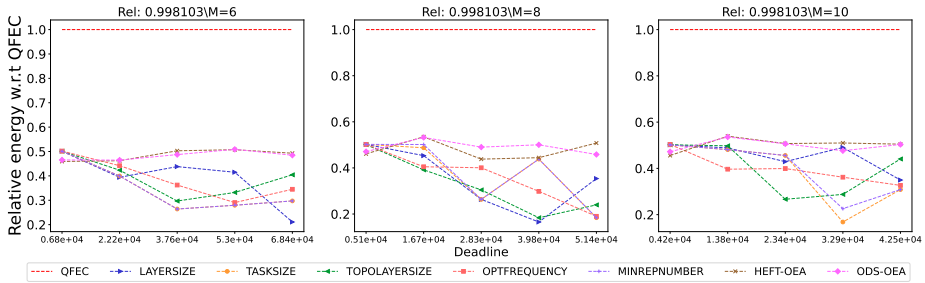

Figure 2965: Assessing the performance of different processor number on the QR.

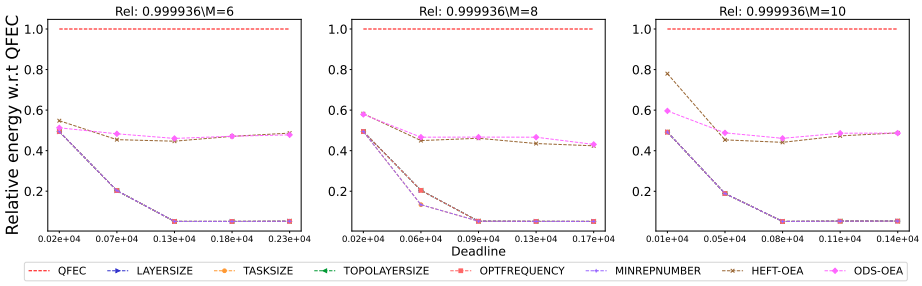

Figure 2966: Assessing the performance of different processor number on the Seismology workflow.

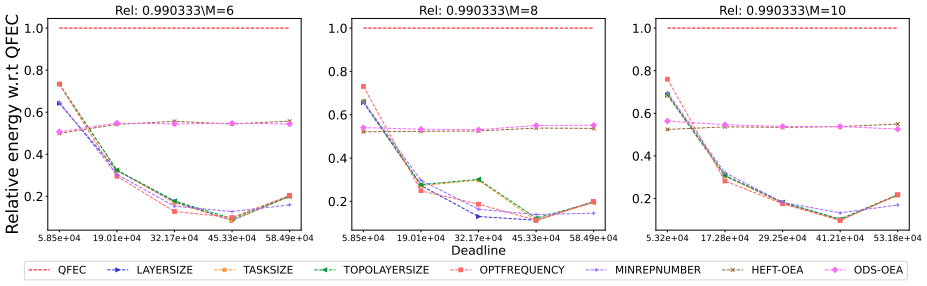

Figure 2967: Assessing the performance of different processor number on the SoyKB workflow.

I.8.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

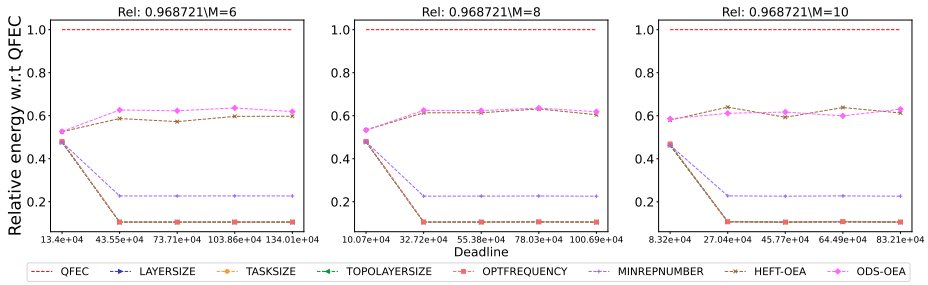

Figure 2968: Assessing the performance of different processor number on the BLAST workflow.

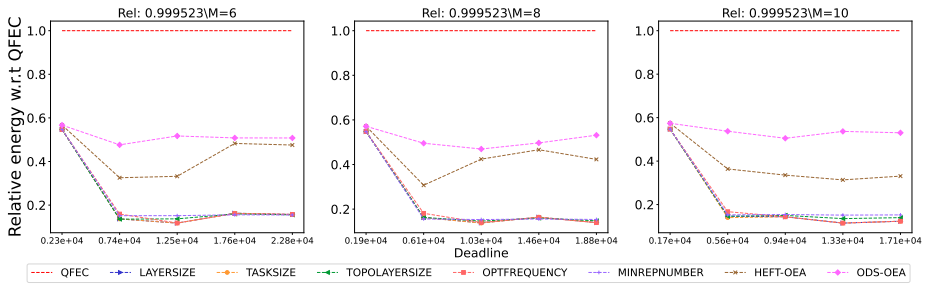

Figure 2969: Assessing the performance of different processor number on the BWA workflow.

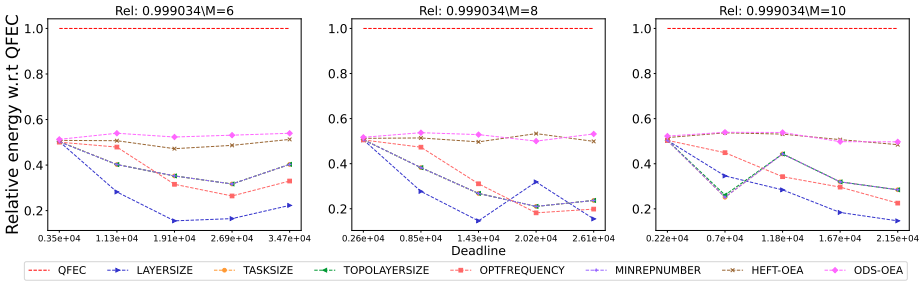

Figure 2970: Assessing the performance of different processor number on the Cholesky workflow.

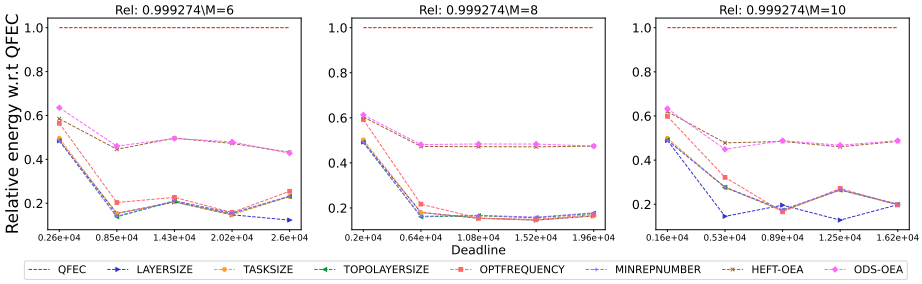

Figure 2971: Assessing the performance of different processor number on the Cycles workflow.

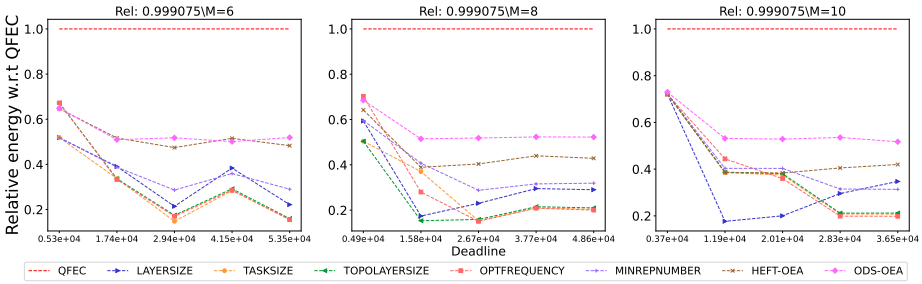

Figure 2972: Assessing the performance of different processor number on the Epigenomics workflow.

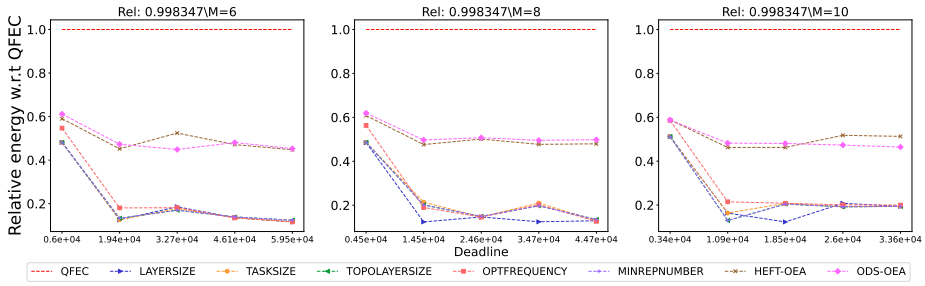

Figure 2973: Assessing the performance of different processor number on the Genome workflow.

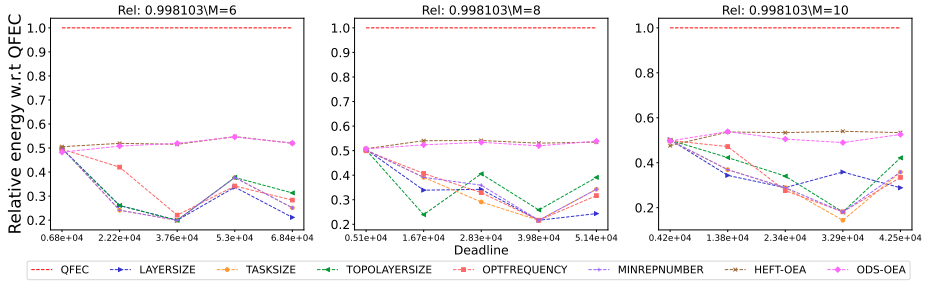

Figure 2974: Assessing the performance of different processor number on the LU.

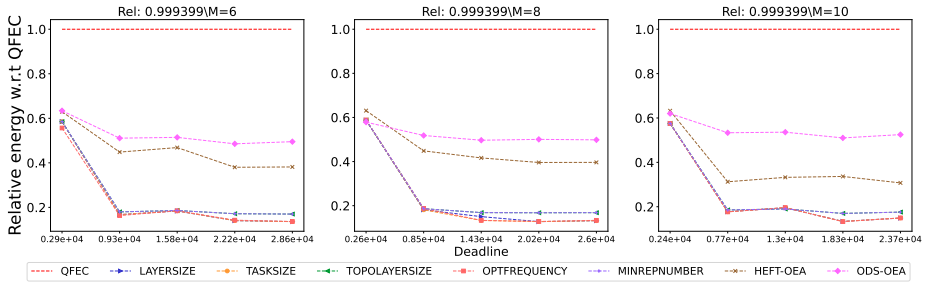

Figure 2975: Assessing the performance of different processor number on the Montage workflow.

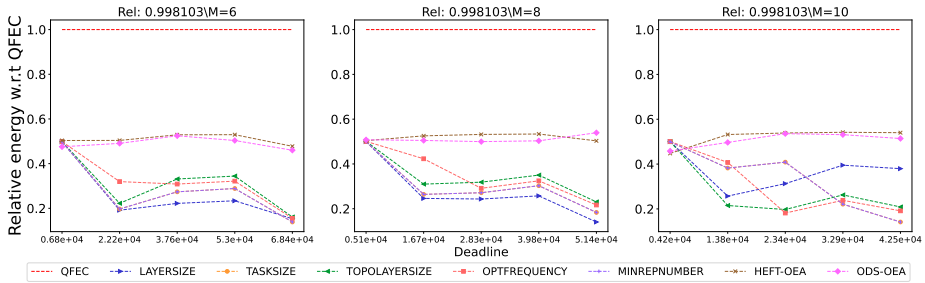

Figure 2976: Assessing the performance of different processor number on the QR.

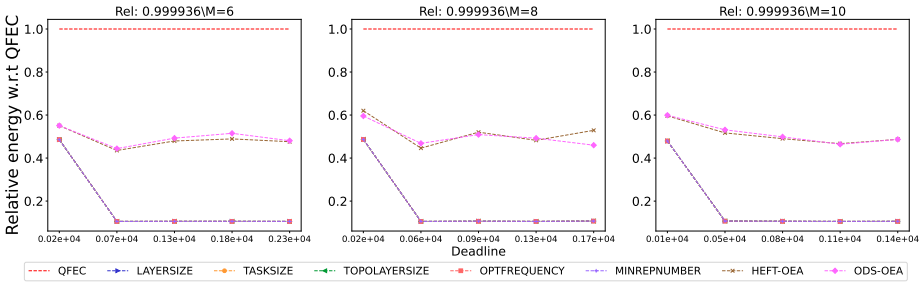

Figure 2977: Assessing the performance of different processor number on the Seismology workflow.

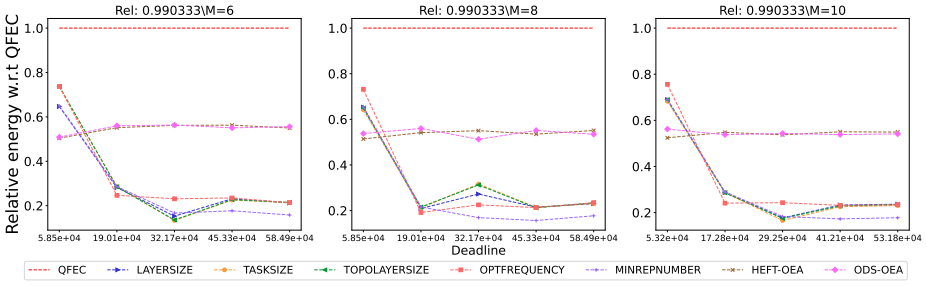

Figure 2978: Assessing the performance of different processor number on the SoyKB workflow.

I.8.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

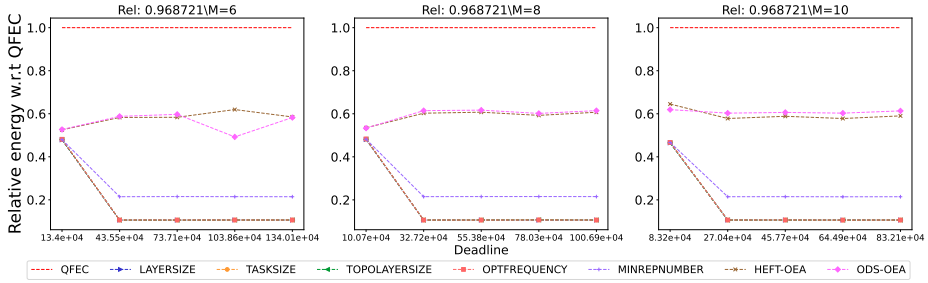

Figure 2979: Assessing the performance of different processor number on the BLAST workflow.

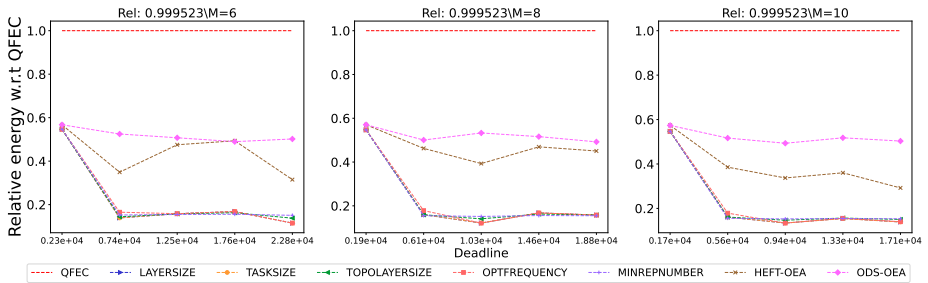

Figure 2980: Assessing the performance of different processor number on the BWA workflow.

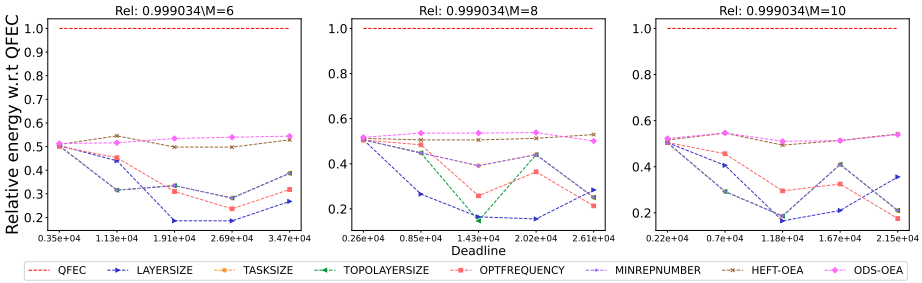

Figure 2981: Assessing the performance of different processor number on the Cholesky workflow.

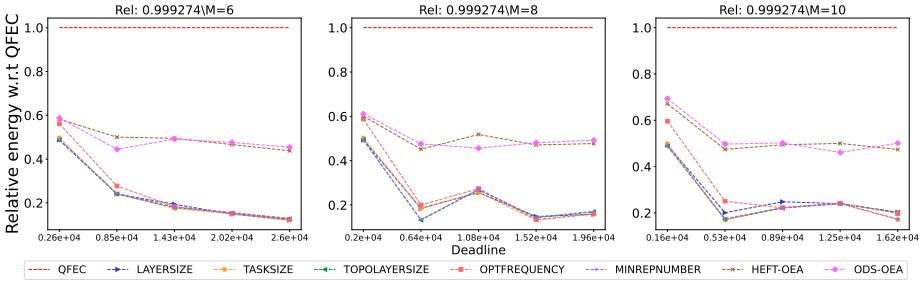

Figure 2982: Assessing the performance of different processor number on the Cycles workflow.

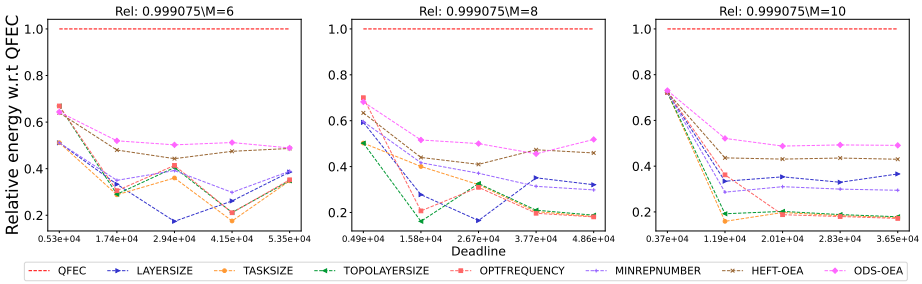

Figure 2983: Assessing the performance of different processor number on the Epigenomics workflow.

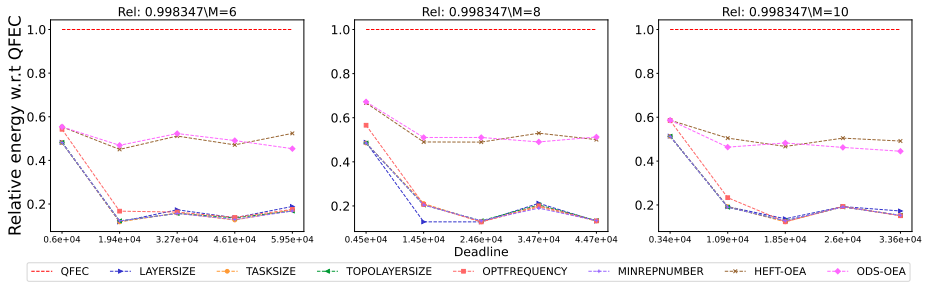

Figure 2984: Assessing the performance of different processor number on the Genome workflow.

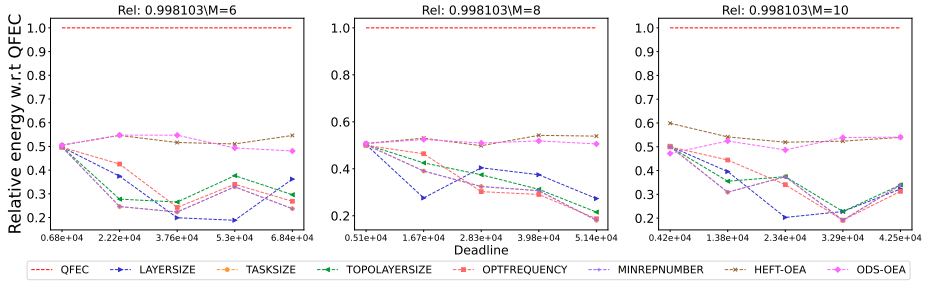

Figure 2985: Assessing the performance of different processor number on the LU.

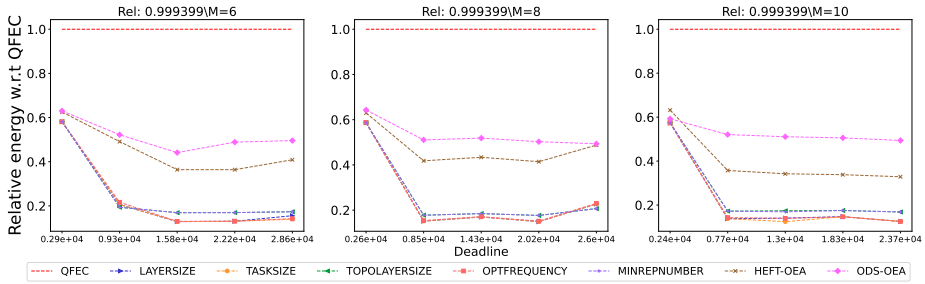

Figure 2986: Assessing the performance of different processor number on the Montage workflow.

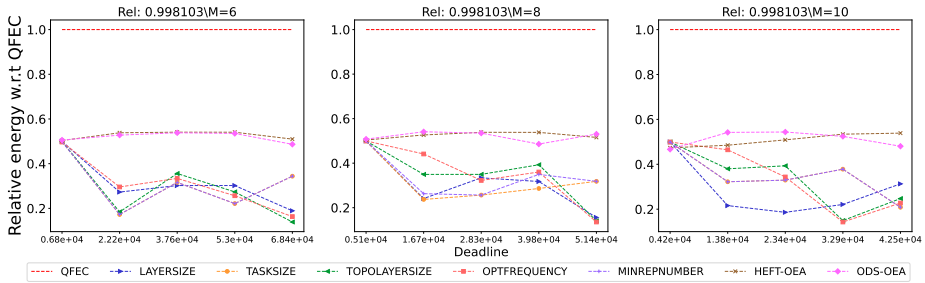

Figure 2987: Assessing the performance of different processor number on the QR.

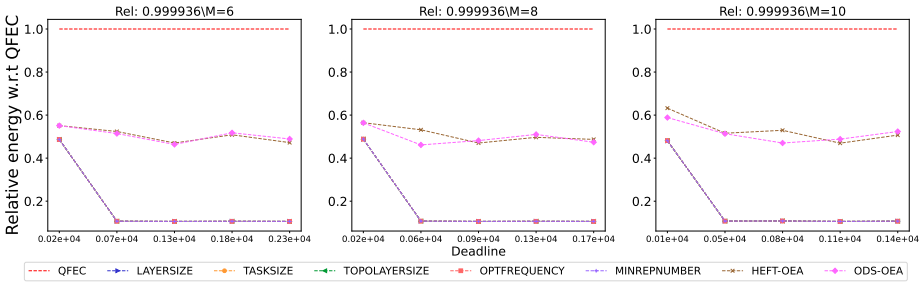

Figure 2988: Assessing the performance of different processor number on the Seismology workflow.

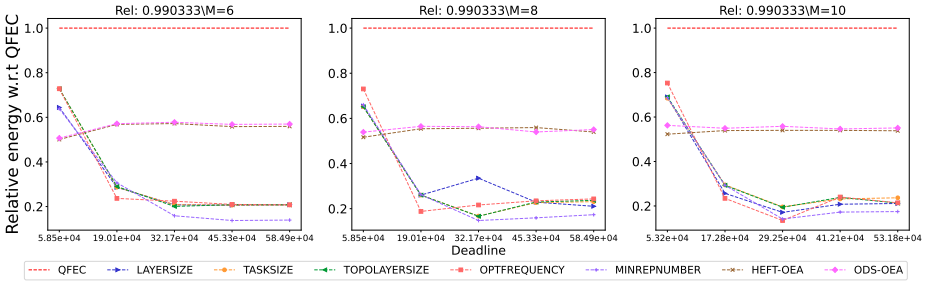

Figure 2989: Assessing the performance of different processor number on the SoyKB workflow.

I.9 $BC/WC = 0.9$

I.9.1 Frequency set $f_1 = \{1.0, 0.8, 0.6, 0.4, 0.15\}$

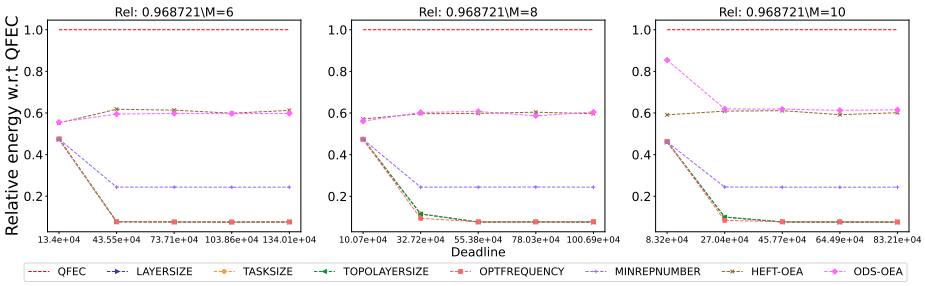

Figure 2990: Assessing the performance of different processor number on the BLAST workflow.

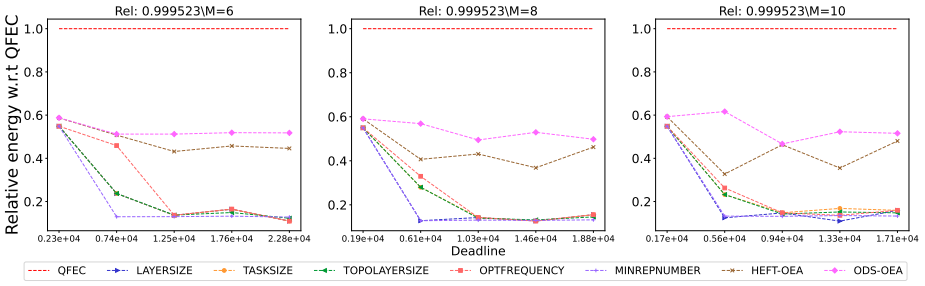

Figure 2991: Assessing the performance of different processor number on the BWA workflow.

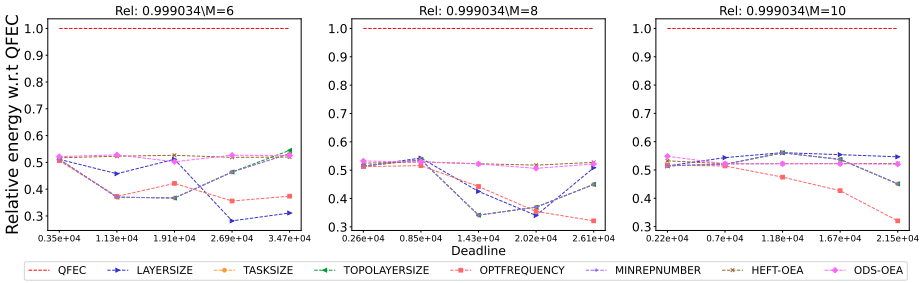

Figure 2992: Assessing the performance of different processor number on the Cholesky workflow.

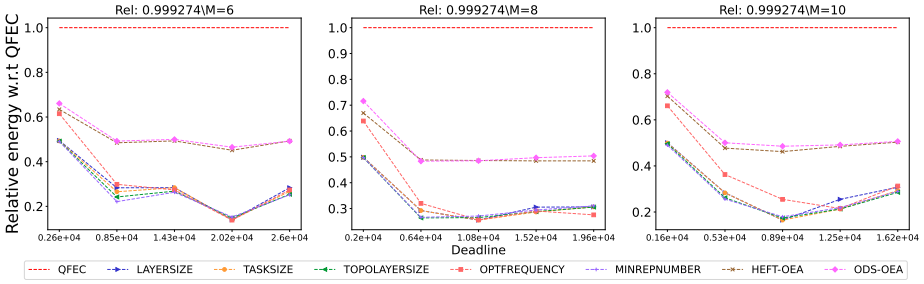

Figure 2993: Assessing the performance of different processor number on the Cycles workflow.

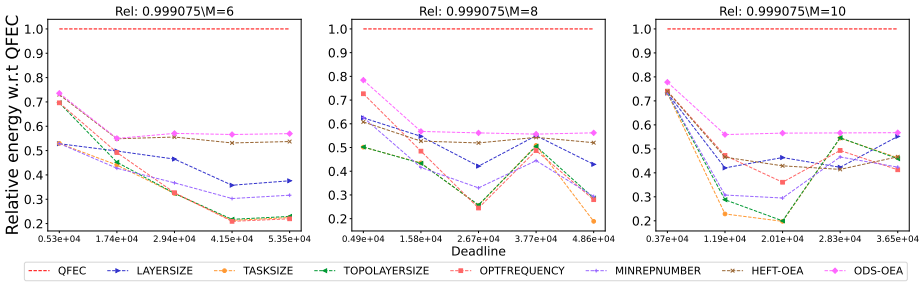

Figure 2994: Assessing the performance of different processor number on the Epigenomics workflow.

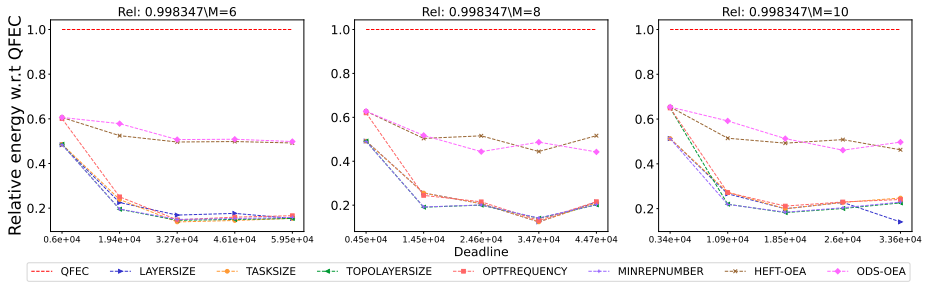

Figure 2995: Assessing the performance of different processor number on the Genome workflow.

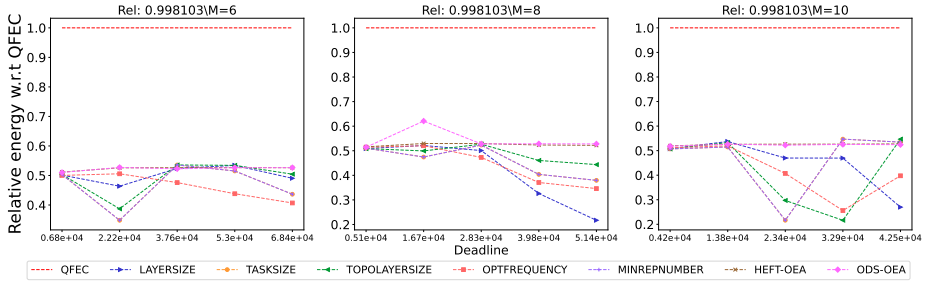

Figure 2996: Assessing the performance of different processor number on the LU.

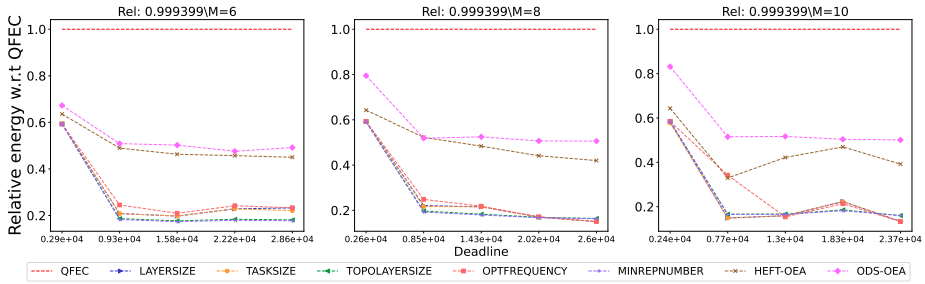

Figure 2997: Assessing the performance of different processor number on the Montage workflow.

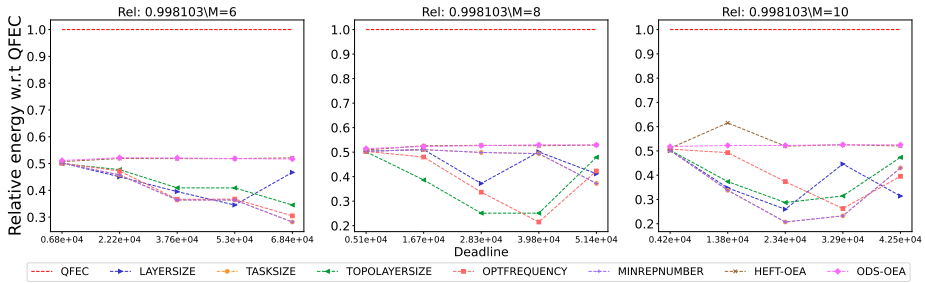

Figure 2998: Assessing the performance of different processor number on the QR.

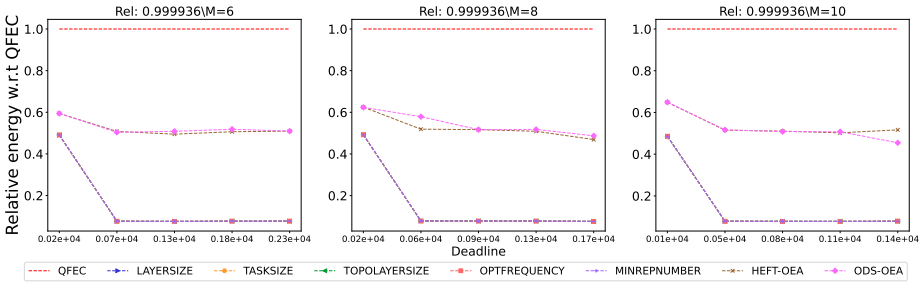

Figure 2999: Assessing the performance of different processor number on the Seismology workflow.

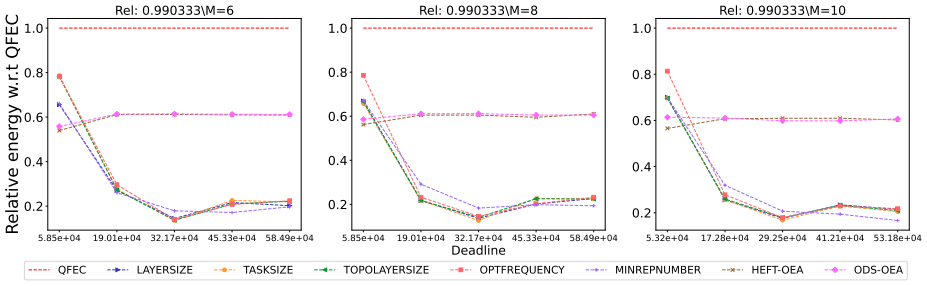

Figure 3000: Assessing the performance of different processor number on the SoyKB workflow.

I.9.2 Frequency set $f_2 = \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1\}$

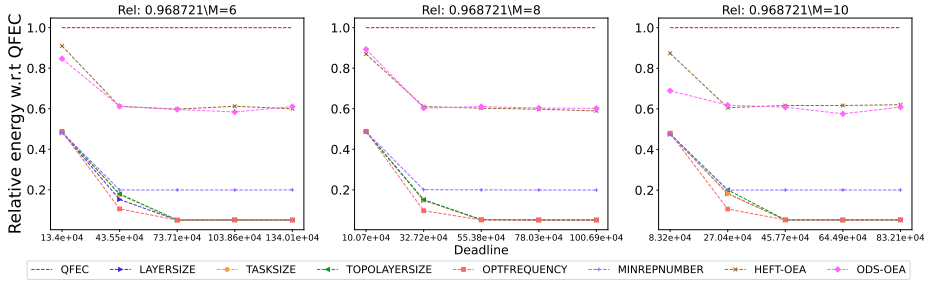

Figure 3001: Assessing the performance of different processor number on the BLAST workflow.

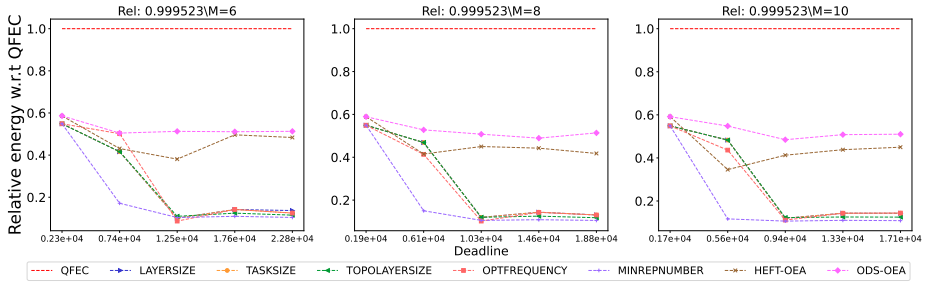

Figure 3002: Assessing the performance of different processor number on the BWA workflow.

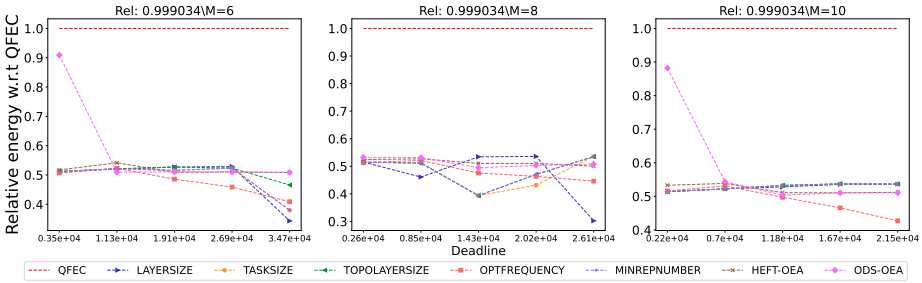

Figure 3003: Assessing the performance of different processor number on the Cholesky workflow.

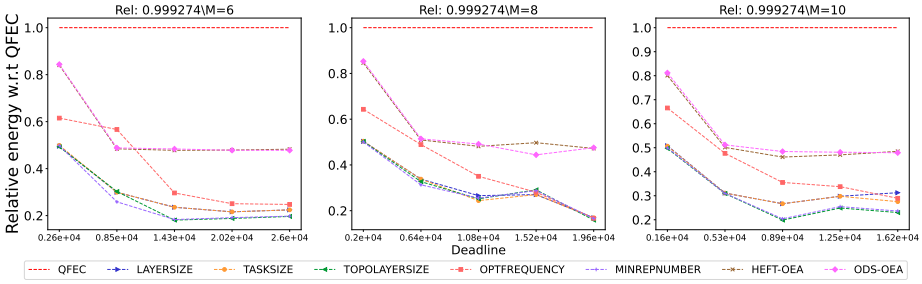

Figure 3004: Assessing the performance of different processor number on the Cycles workflow.

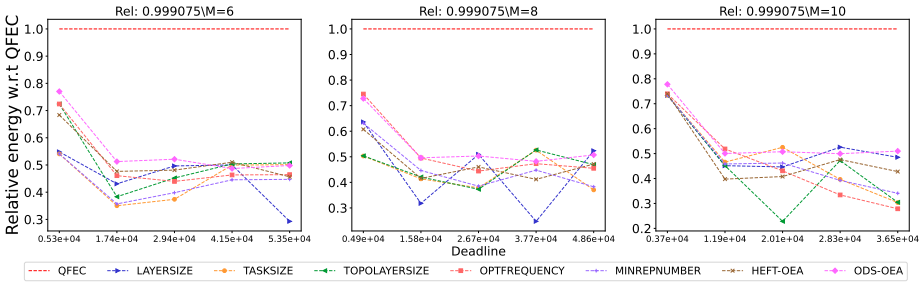

Figure 3005: Assessing the performance of different processor number on the Epigenomics workflow.

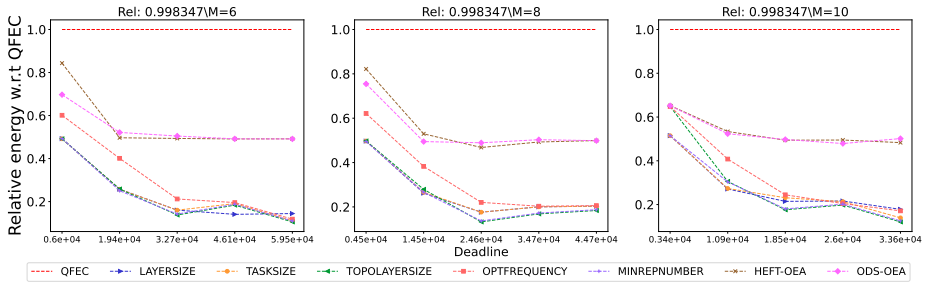

Figure 3006: Assessing the performance of different processor number on the Genome workflow.

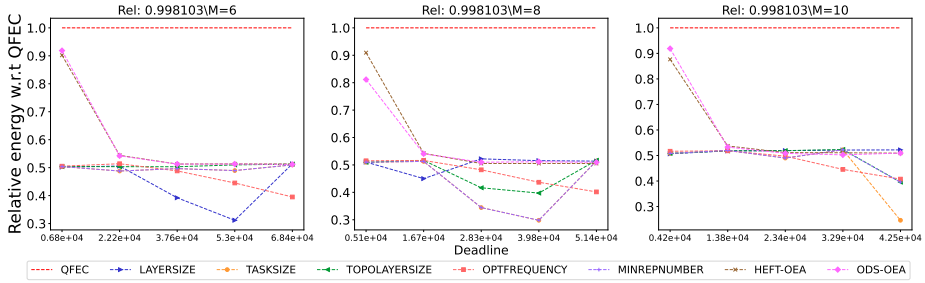

Figure 3007: Assessing the performance of different processor number on the LU.

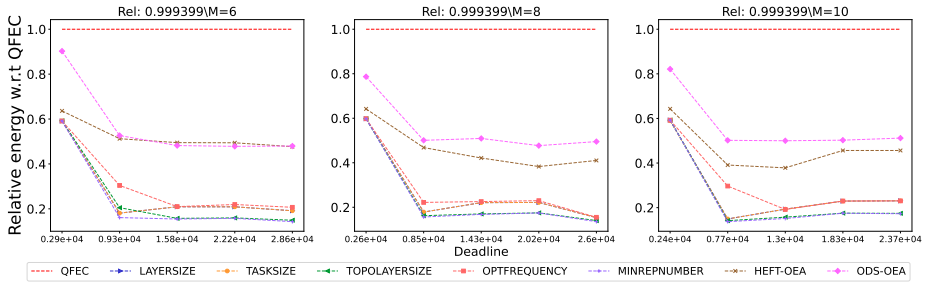

Figure 3008: Assessing the performance of different processor number on the Montage workflow.

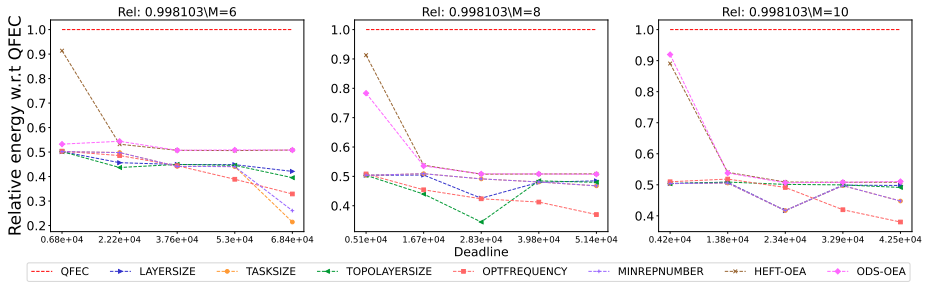

Figure 3009: Assessing the performance of different processor number on the QR.

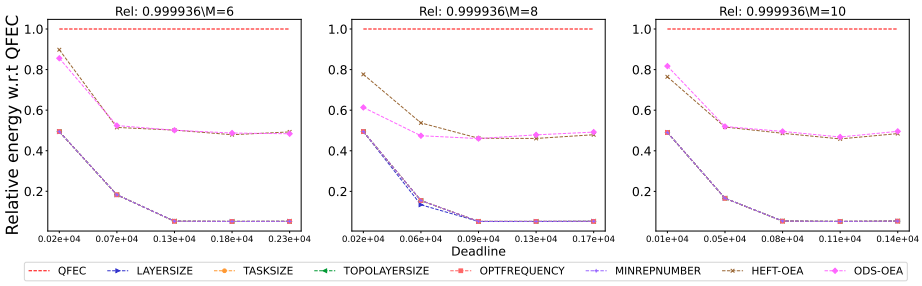

Figure 3010: Assessing the performance of different processor number on the Seismology workflow.

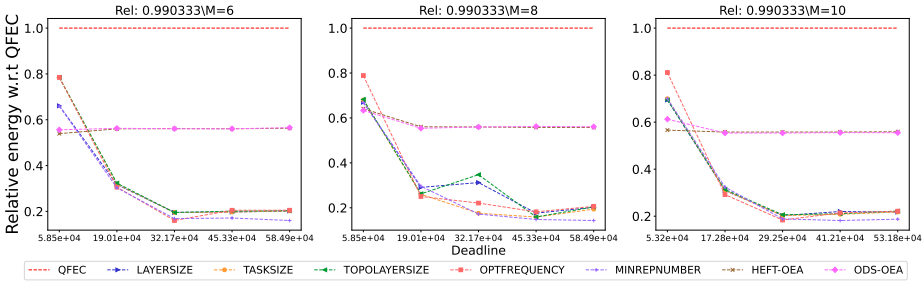

Figure 3011: Assessing the performance of different processor number on the SoyKB workflow.

I.9.3 Frequency set $f_3 = \{1.0, 0.86, 0.71, 0.57, 0.46, 0.32, 0.21\}$

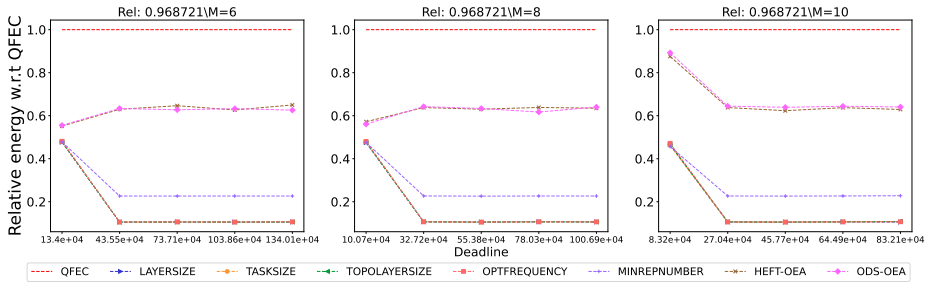

Figure 3012: Assessing the performance of different processor number on the BLAST workflow.

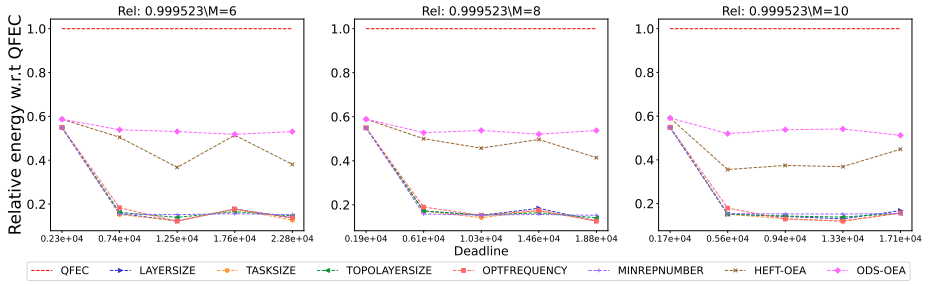

Figure 3013: Assessing the performance of different processor number on the BWA workflow.

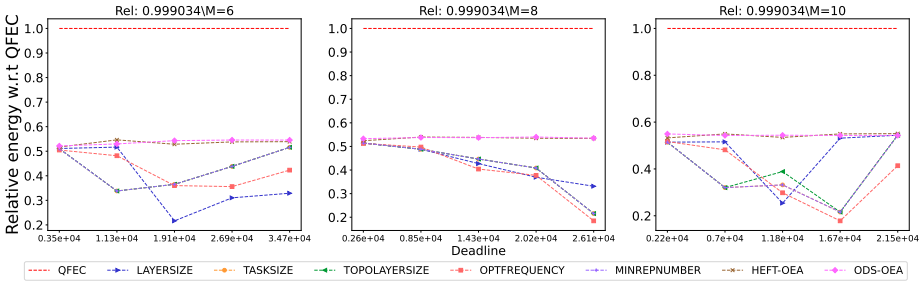

Figure 3014: Assessing the performance of different processor number on the Cholesky workflow.

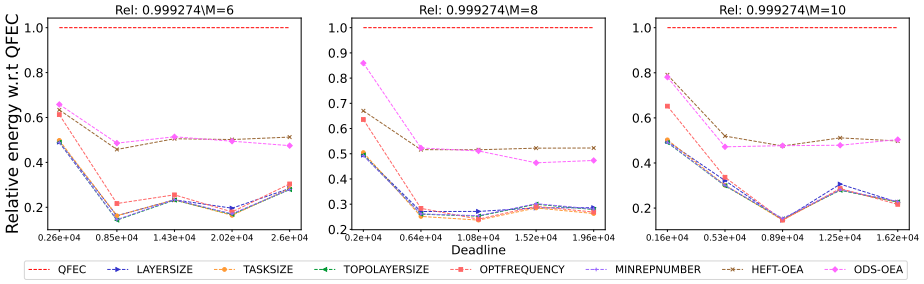

Figure 3015: Assessing the performance of different processor number on the Cycles workflow.

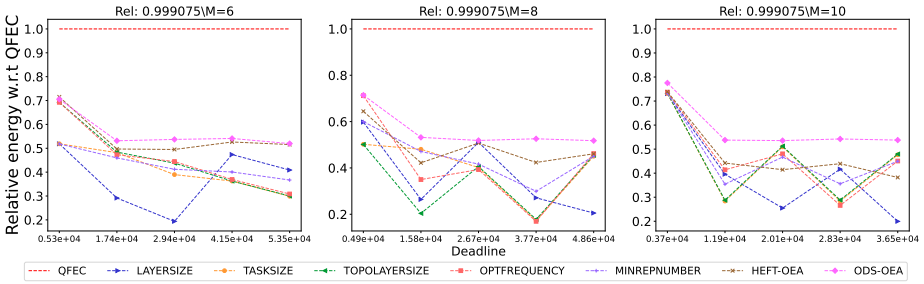

Figure 3016: Assessing the performance of different processor number on the Epigenomics workflow.

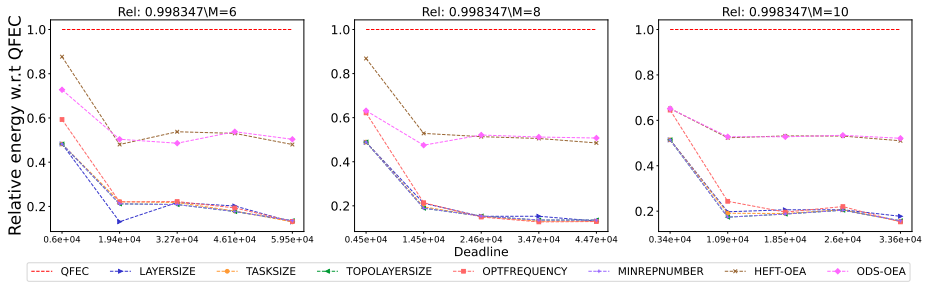

Figure 3017: Assessing the performance of different processor number on the Genome workflow.

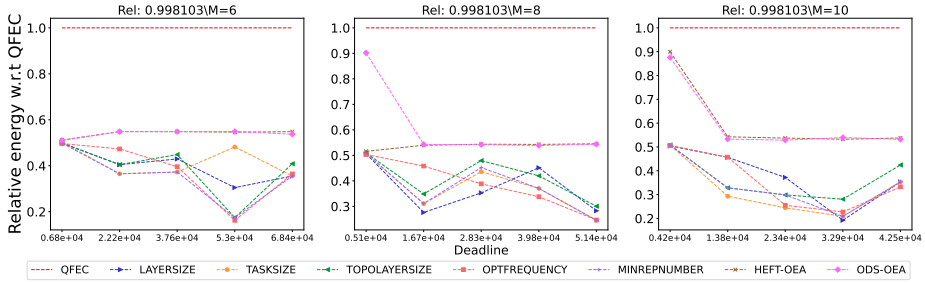

Figure 3018: Assessing the performance of different processor number on the LU.

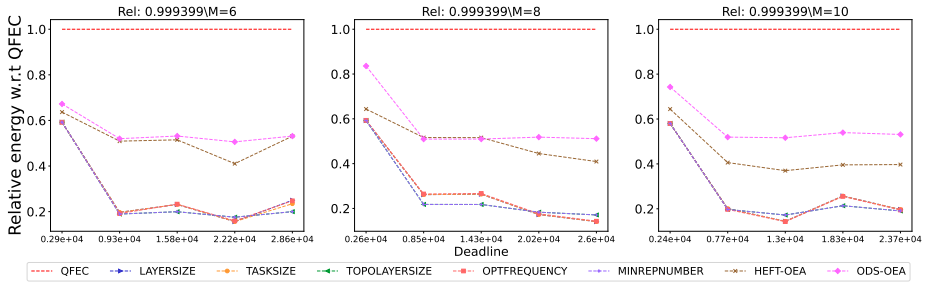

Figure 3019: Assessing the performance of different processor number on the Montage workflow.

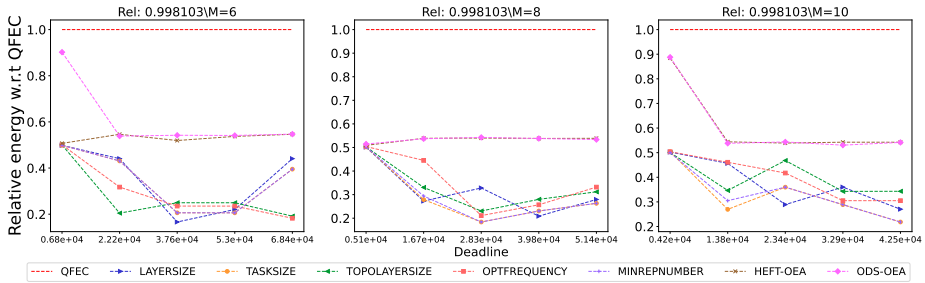

Figure 3020: Assessing the performance of different processor number on the QR.

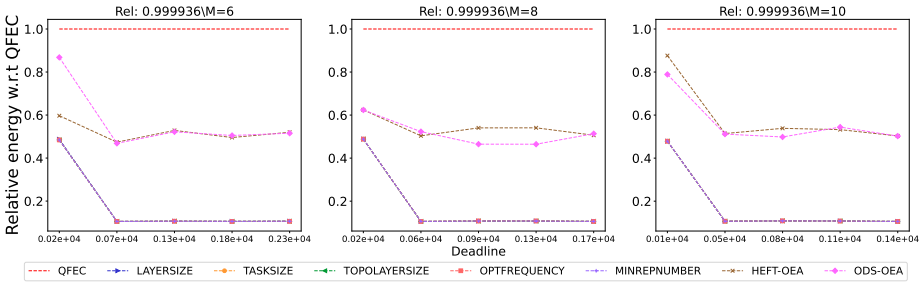

Figure 3021: Assessing the performance of different processor number on the Seismology workflow.

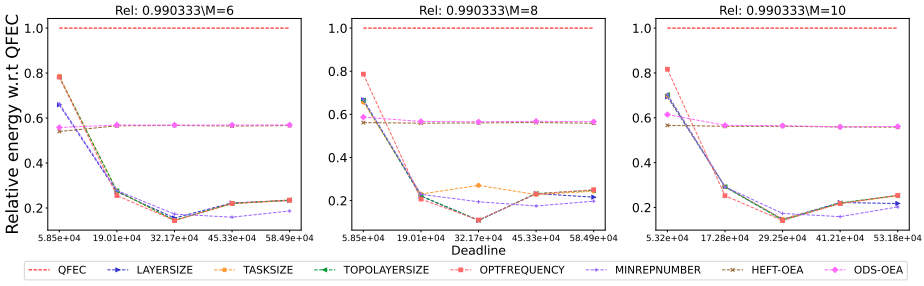

Figure 3022: Assessing the performance of different processor number on the SoyKB workflow.

I.9.4 Frequency set $f_4 = \{1.0, 0.844, 0.750, 0.633, 0.562, 0.50, 0.421, 0.375, 0.316, 0.281, 0.250, 0.211\}$

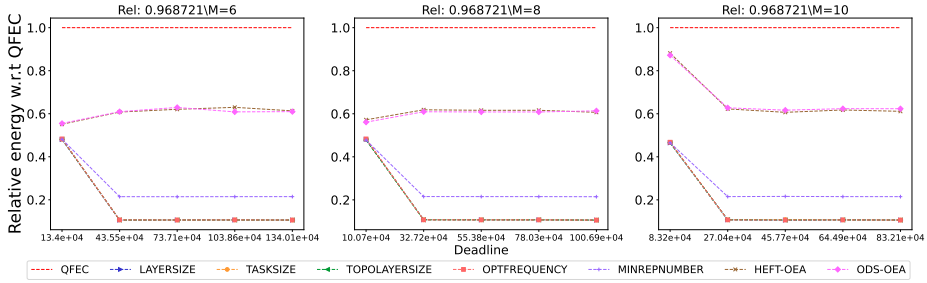

Figure 3023: Assessing the performance of different processor number on the BLAST workflow.

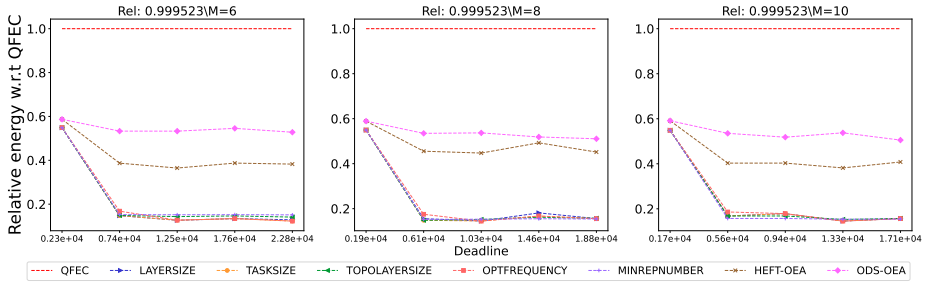

Figure 3024: Assessing the performance of different processor number on the BWA workflow.

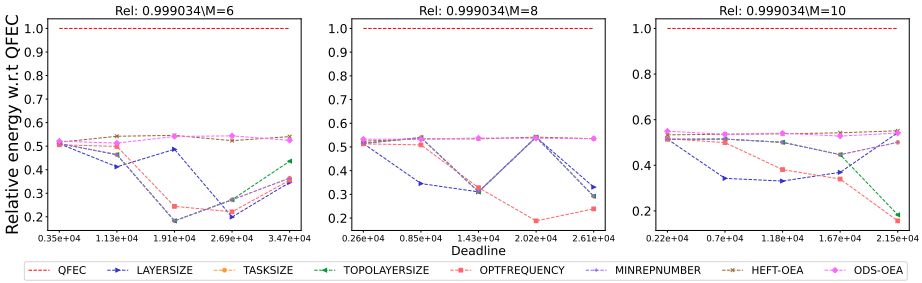

Figure 3025: Assessing the performance of different processor number on the Cholesky workflow.

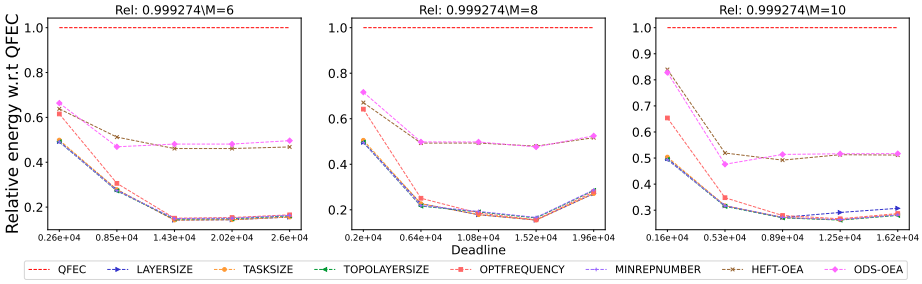

Figure 3026: Assessing the performance of different processor number on the Cycles workflow.

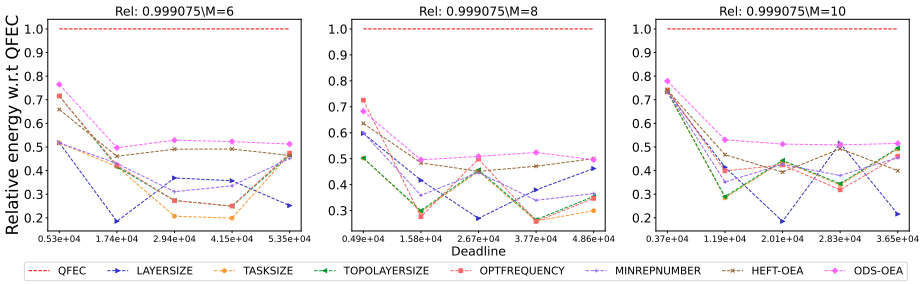

Figure 3027: Assessing the performance of different processor number on the Epigenomics workflow.

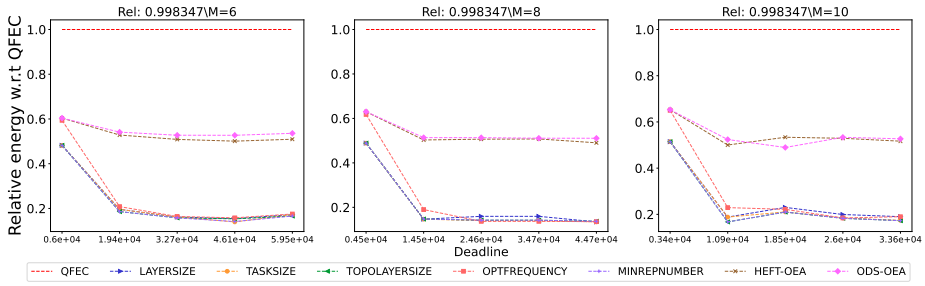

Figure 3028: Assessing the performance of different processor number on the Genome workflow.

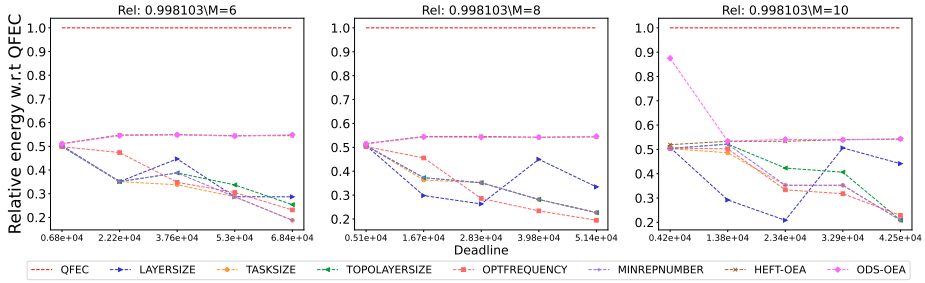

Figure 3029: Assessing the performance of different processor number on the LU.

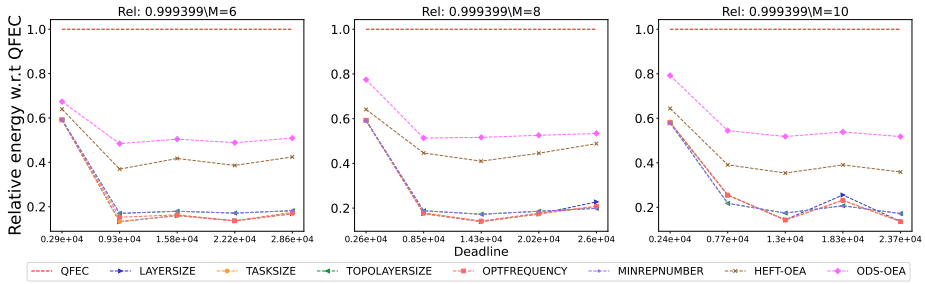

Figure 3030: Assessing the performance of different processor number on the Montage workflow.

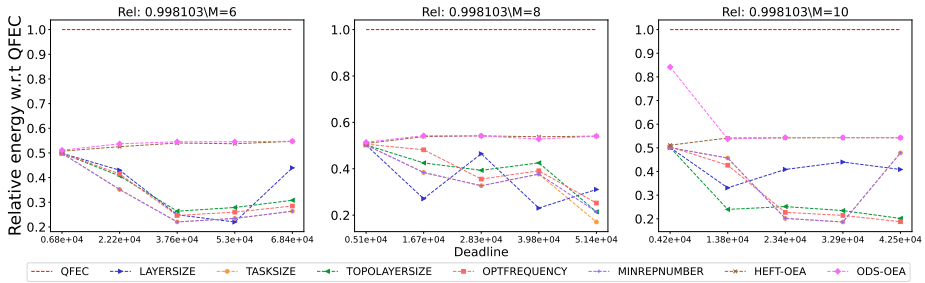

Figure 3031: Assessing the performance of different processor number on the QR.

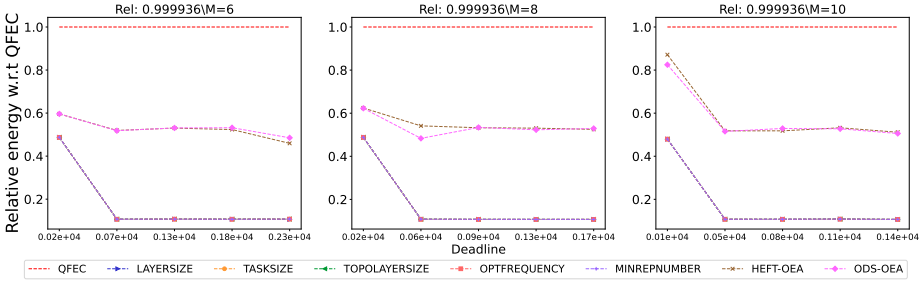

Figure 3032: Assessing the performance of different processor number on the Seismology workflow.

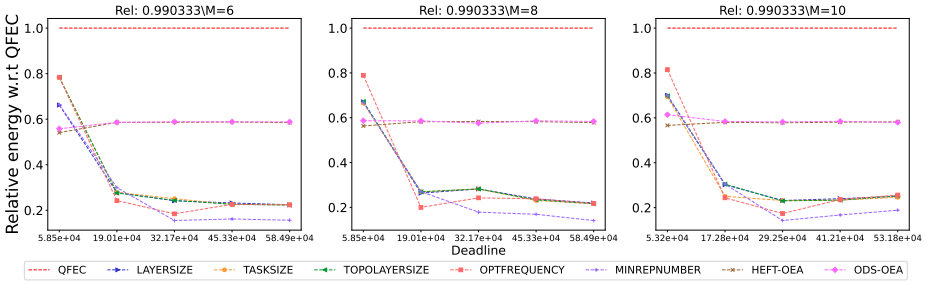

Figure 3033: Assessing the performance of different processor number on the SoyKB workflow.

RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399