



Hard-deadline constrained workflows scheduling using metaheuristic algorithms

Alexander Visheratin¹, Mikhail Melnik¹, Nikolay Butakov¹ and Denis Nasonov¹

¹ITMO University, St. Petersburg, Russia.

*alexvish91@gmail.com, mihail.melnik.ifmo@gmail.com, alipoov.nb@gmail.com,
denis.nasonov@gmail.com*

Abstract

An efficient scheduling is the essential part of complex scientific applications processing in computational distributed environments. The computational complexity comes as from environment heterogeneity as from the application structure that usually is represented as a workflow which contains different linked tasks. A lot of well-known techniques were proposed by different scientific groups. The most popular of them are based on greedy list-based heuristics or evolutionary metaheuristics. In this paper we investigate the applicability of previously developed metaheuristic algorithm – coevolutional genetic algorithm (CGA) for scheduling series of workflows with hard deadlines constraints.

Keywords: Scheduling, workflow, grid, genetic algorithm, coevolution, HEFT.

1 Introduction

Today complex computational systems based on Grids, clusters or computational clouds play very important role in applied scientific researches, which usually use composite applications for computational purposes. To execute these applications in distributed environments they are split into separated tasks, which can be run on different resources with left constrains on dependencies between those tasks. Formally these applications are called workflows and are represented by graph with defined tasks on the nodes and dependences as the edges.

Proper scheduling of composite applications' execution on available resources is essential part of efficient problems solving, it brings us to investigate optimization mechanisms of scheduling process. For different reasons different optimization criteria can be used, such as total execution time (makespan), cost, energy efficiency and etc. For cloud environments computational cost often is the most important measure, since users have to pay for the resources usage time. For the Grid computing the makespan of the workflow is most crucial, because execution results leads to research progress, and a lot of composite applications of many users wait for opportunity to start running. In our work we

address to a problem of urgent computing that specifies strong constraints on execution time called hard deadlines. The specifics come from the necessity to have the results exactly on set up time otherwise they become valueless. In example, it's crucial for the preventing system in the hazard of flooding, earthquake, epidemic, fire and etc.

Two main groups of algorithms are used for building workflow execution schedules. The first group is list-based heuristics that includes two main steps task ranking and resource allocation based on this ranking. One of the mostly used algorithms is heterogeneous earliest finish time (HEFT) algorithm (Arabnejad, 2013) due to its short computation time and ability to generate solutions with good quality. In this work we adopted it for hard deadline case. The second group is metaheuristic algorithms. Algorithms of this group usually require more time for execution, than heuristics, but can find much better solutions. The most popular algorithms in this area are genetic algorithm (GA), particle swarm optimization (PSO), gravitational search algorithm (GSA), ant-colony optimization (ACO) (Yu J. R., 2008) and so on. In our work we improved coevolutional genetic algorithm (CGA) for hard deadline cases.

On the other hand optimization can be conducted not only by efficient search of the tasks mapping on the computational resources but also by resources optimization for the certain set of task using virtualization technologies that allow to re-balance main resource configuration parameters (CPU, memory, etc.) according to the tasks needs. In this work we also amuse this optimization in coevolutional genetic algorithm.

2 Related works

Today, there are two widely used approaches that consider hard-deadline constraint for developing real-time systems – Rate Monotonic (RM) and Earliest Deadline First (EDF) algorithms. In (Buttazzo, 2005) comprehensive comparison of these algorithms was performed in order to figure out strong and weak points of both algorithms and dispel misconceptions related to them. It was shown that although RM have simpler implementation, in general it doesn't have properties often credited to it, such as better jitter control or predictability in overload conditions. EDF, on the other hand, allows better resources utilization and better aperiodic activities responsiveness. Regarding our work, the RM concept of assigning priorities for tasks inversely to their periods was used in HEFT modification to implement deadlines support.

Currently, a large number of scientific researches is dedicated to workflows execution optimization with regard to one or several quality of service (QoS) parameters, including deadlines. For example, Yeo and Buyya in (Yeo, 2005) present an approach to handle service level agreements (SLA) in distributed environments in order to enhance the utility of the computational cluster. Authors propose technique for scheduling tasks with regard to user defined quality of service parameters, such as computational cost and execution deadline. Authors show efficiency of the algorithm in comparison with older algorithm for different QoS characteristics, but proposed approach can't be directly used in our case since it does not take into account several workflows with different deadlines and hard option as well.

Abrishami et al. in (Abrishami, 2013) present and compare algorithms for scheduling, considering deadlines and computation cost. Authors aim their algorithms to minimize execution cost of workflows as well as to keep their deadlines. The main drawback of proposed algorithms regarding our use case is that they consider only one workflow with deadlines, when in our approach several workflows with different deadlines can be used with hard deadline feature. However in the future we suppose to adapt the most efficient of described algorithms (i.e. IC-PCP) for the case of several deadlines with hard deadline option to compare it to our solution.

In (Bochenina, 2014) Bochenina performs comparative study of different scheduling algorithms for heterogeneous systems considering deadlines. Author proposes two approaches for dealing with workflow execution time constraints, which are able to take into account workflow soft deadlines in order to build proper schedule.

In (Nasonov D. a., 2014) authors try to meet hard deadline of predefined early warning system's scenarios by background scheduling optimization with accepted resource reliability parameters as well as low scenario changes in workload that could be easily adapted to already formed plan.

Mao and Humphrey in (Mao, 2011) describe the approach for workflows execution optimization by dynamically managing virtual machines in cloud environment. The main goal of the research was to minimize execution cost as well as try to keep workflow's deadline. This work is related to processing single workflows with soft deadlines in a cost-efficient way, when we try to maintain several workflows with hard deadlines.

In (Nebro, 2009) authors propose cellular genetic algorithm based on neighborhood interaction, where an individual can only cooperate with its nearby neighbors in the breeding loop. Although comparison against NSGA-II and SPEA2 shown good results, algorithm applicability to series of workflows with possible deadlines is the matter of future research.

3 Problem statement

In this section we describe our assumptions regarding workflows, their tasks and computational resources, define the problem and propose our solution for solving it.

3.1 Workflows

Usually workflows are presented in the form of directed acyclic graph (DAG) where nodes stand for tasks and edges represent dependencies between tasks (in most cases data dependencies). Detailed description of this approach is given in (Sinnen, 2007). In our work we define total execution time (makespan) as the main parameter of optimization, limited with deadline constrains. Every workflow can have user specified hard deadline - time interval, in which workflow has to be executed by all means. We cannot start execution until all deadlines are kept in the schedule.

In our work we use a number of synthetically generated workflows replicated real scientific applications from different scientific fields: Montage (astronomy), CyberShake (earthquake science), Epigenomics (biology), Inspiral (gravitational physics) and SIPHT (biology) (Bharathi, 2008). They provide detailed information about workflow structure, relative execution time of tasks, input and output data for each task and data dependencies. Bharathi et al. developed workflow generator, which allows to generate versions of described workflows with different number of tasks in form of DAX (Directed Acyclic Graph in XML). Example of workflow structure can be found on Fig. 1.

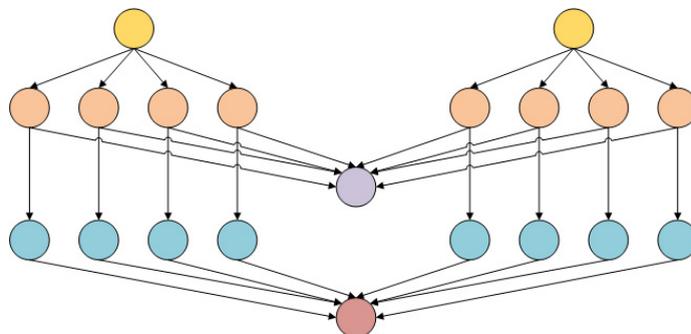


Figure 1: Synthetic workflow structure (CyberShake)

Task completion time on execution phase is estimated by normal distribution for each task t with mean value m and variance d . On scheduling phase completion time considered constant and equal to mean value.

3.2 Computational resources

All computational resources have a certain reliability and we have to take this fact into consideration to make the failure probability of workflow with deadline as low as possible. For these purposes we try to guarantee every task of such workflows 99% execution reliability through task replication. For example, if we map task on resource with reliability of 0.8 we have to duplicate this task on another resource with reliability at least of 0.95 to ensure 99% execution reliability, because $P(AB) = P(A) \cdot P(B)$, where $P(A)$ – failure rate of resource 1 and $P(B)$ – failure rate of resource 2.

In our work we consider computational environment to implement virtualization techniques. It means that instead of real computational resources we map workflow tasks on a number of virtualized resources, which are based on real resources.

3.3 Problem statement

As it was mentioned above, there are a lot of cases when hard deadline constrains are critical for final results and urgent computing techniques are required, It forces us to change priorities during optimization and to adapt existed algorithms to the strong limitations at first and only then optimize makespan itself considering dynamic changes during execution process in the environment, For such algorithm the most important is to get results in fixed time Moreover there is need to process several workflows at once, where each of them can have urgent criteria and have to be completed in certain amount of time, in other words has hard execution deadline. Our research is aimed to check the applicability of metaheuristic algorithms, in particular, coevolutional genetic algorithm (Butakov, 2014), for scheduling workflows with hard deadlines.

4 Proposed solution

In this section proposed coevolution metaheuristic and modified heuristic algorithms are presented.

4.1 Hard deadline coevolutionary genetic algorithm (HDCGA)

We consider virtualization as one of the ways for scheduling performance increase. In (Butakov, 2014) concept of coevolutional approach for building schedules in heterogeneous virtualized environments. The idea is based on simultaneous evolutionary processes for tasks schedule and computational environment. Tasks mapping evolution is organized in the classic way of crossover, mutation and selection using fitness function. Resources evolution is performed with virtual resources management. In every population real resources are split into set of virtual machines. These resources form an evolution particle, as it is shown on Figure 2. Principal schema of the HDCGA algorithm is shown on Figure 3. On each step of the process configuration and scheduling population are merged. Then for each pair of particles we build schedule, evaluate it with fitness function and perform selection strategy based on fitness function results. In our experiments it was noticed that usage of this algorithm allow to outperform list-based heuristics, such as HEFT, up to 84%.

For implementation of hard-deadline workflows support, fitness function computation was modified in the following way: if makespan estimation of the workflow exceeds its deadline, fitness function result for corresponding pair of particles gets very high penalty, which makes it impossible for such individuals to take part in the evolutionary process.

To simulate dynamic environments we used the concept described in (Nasonov D. B., 2014). The key principle of this approach is combined application of heuristic and metaheuristic approaches for processing of system events, such as resource failure. On such event appearance rescheduling procedure is started. It consists of two steps. On the first step we generate schedule using HEFT and immediately apply it to the environment. At the same time we calculate time interval for GA execution. If GA could find solution better than HEFT, we apply it to the environment.

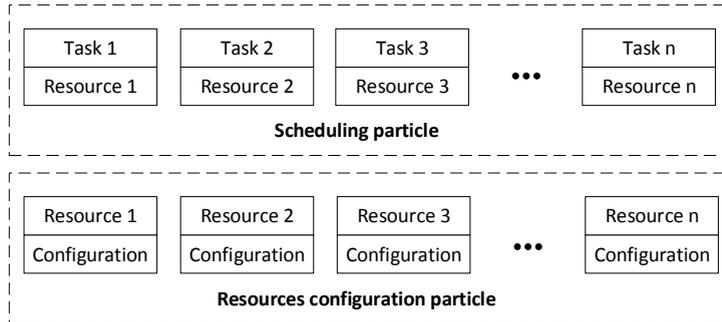


Figure 2: Coevolution particles schema

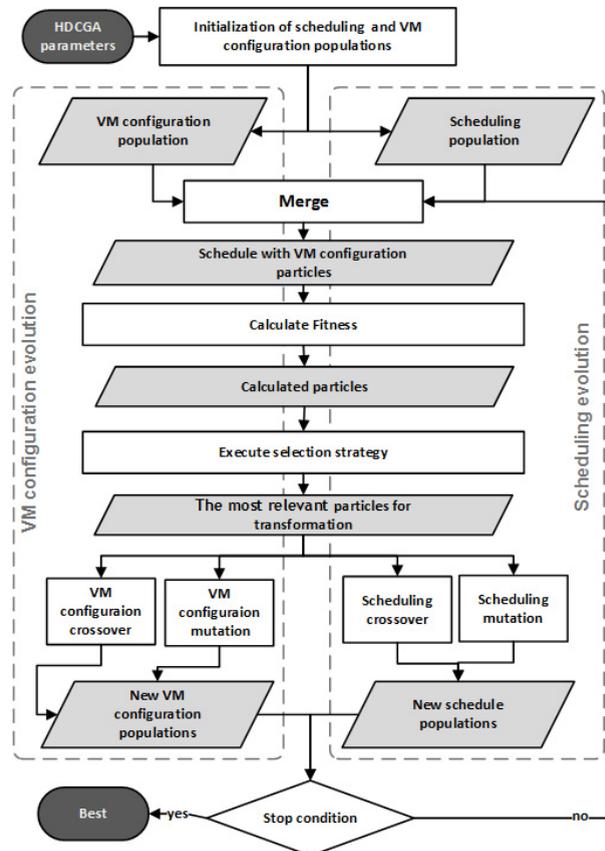


Figure 3: HDCGA algorithm schema

4.2 Deadline Heterogeneous Earliest Finish Time (DHEFT)

Competitive algorithm for this set of experiments is modified HEFT – deadline HEFT (DHEFT). The DHEFT algorithm is presented on the Figure 4. Original algorithm does not take into account possible workflow deadlines so we made slight changes in ranking in the way that tasks of workflows with smaller deadline got much higher ranking and were processed earlier. For this purpose for tasks of workflows with deadlines we assign priorities inversely to workflows' deadlines. On ranking phase tasks with priorities obtain additional rank according to the formula:

$$r_{add} = inc^p$$

Where r_{add} – additional rank, p – task priority, inc – increase parameter.

Described approach doesn't guarantee keeping deadlines but at least it takes into consideration workflows' deadlines and is good enough for generation of initial populations for metaheuristic algorithms.

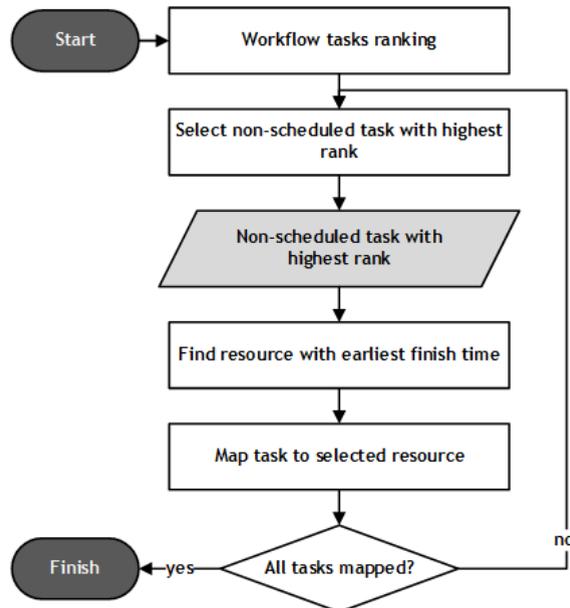


Figure 4: DHEFT algorithm schema

5 Experiments

For experimental part of the research we used computational environment simulator. It allows to simulate workflows execution on the set of resources using different scheduling strategies taking into account resources reliability and virtualization. We determine computational cost for each workflow task as multiplication of runtime content attribute from the file and predefined constant value 20. We also assume that transfer cost between resources is a constant greater than zero. Each task is computed only on one computational resource at a particular moment of time. Every computational resource has predefined value of power in flops. We have fixed set of resources consisted of four instances with corresponding computational powers: 10, 15, 25, 30. For HDCGA we used following parameters: mutation probability – 0.8, crossover probability – 0.6, population size – 50, populations' interactions number – 200, iterations number – 200.

For experimental study of the proposed algorithm we conducted series of experiments for following synthetic workflows – Montage, CyberShake and Inspiral. In each series we used four workflows of the same type to create workflows queue. Every series involves four sets of experiments: one, two, three and all four workflows have hard execution deadline. Resulted makespans of workflows with deadlines were then normalized according to their deadlines.

On figure 5 kernel densities of normalized makespans are presented. As we can see, for all cases HEFT density is distributed quite evenly through all solution space. The negative side of this distribution is that it very much exceeds deadline value, which is depicted by black vertical line on the plot. CGA density, on the other side, shows that the algorithm generated solutions exceeding deadline in a very small number of cases. In Table 1 deadline exceed rates of algorithms for different number of deadlines is presented.

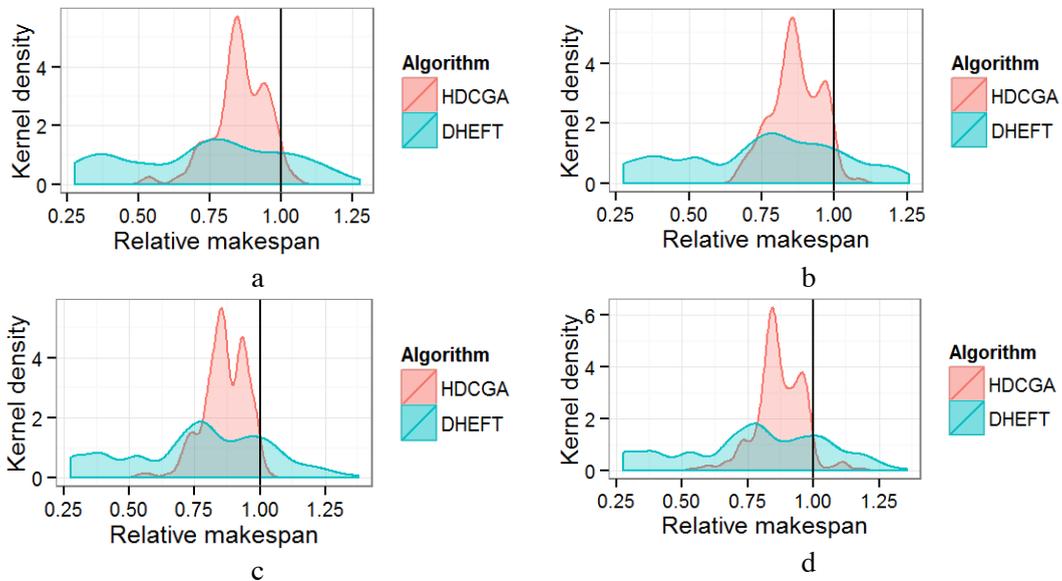


Figure 5: Relative makespan density for the case with one (a), two (b), three (c) and four (d) deadlines

Number of deadlines	HDCGA deadline exceed rate (%)	DHEFT deadline exceed rate (%)
1	1.53	19.33
2	2.31	18.66
3	1.02	20
4	3.47	23.66

Table 1: Deadline exceed rate

Another very important criteria for scheduling algorithm performance is an overall makespan. Makespan increase of HDCGA over DHEFT is presented in figure 6. It can be observed that HDCGA outperforms DHEFT at least at 5% in all cases.

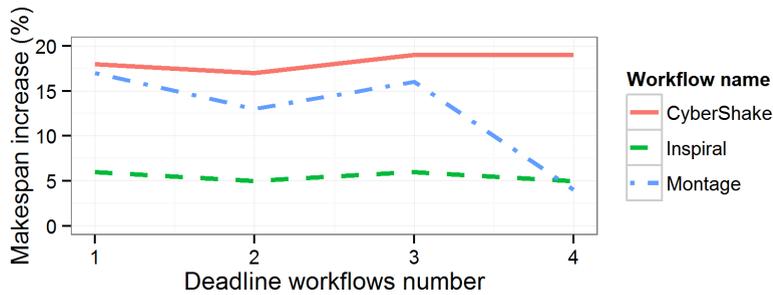


Figure 6: HDCGA total makespan increase over DHEFT

6 Conclusion

In this work we investigated the applicability of metaheuristic algorithms for scheduling several workflows with possible hard deadlines constraints in heterogeneous computational environment implementing virtualization techniques. Experimental results show efficiency of proposed HDCGA algorithm in comparison with DHEFT both from the point of keeping hard deadlines and from the point of generating solutions with lower overall makespan.

During experiments we discovered a number of interesting observations: HDCGA deadline exceed rate on the case of three deadlines is lower than in cases of two and four deadlines; kernel density distribution for all experiments is multi-modal; Montage workflow makespan increase drops sharply from 15% to 5% in the case of four workflows. In our future work we are going to make detailed research and get explanations of these phenomena.

7 Acknowledgements

This paper is supported by Russian Foundation for Basic Research - 15-29-07034 "Technology of distributed cloud computing for the simulation of a large city».

References

- Abrishami, S. M. (2013). Deadline-constrained workflow scheduling algorithms for Infrastructure as a Service Clouds. *Future Generation Computer Systems* 29.1, 158-169.
- Arabnejad, H. (2013). List Based Task Scheduling Algorithms on Heterogeneous Systems-An overview.
- Bharathi, S. e. (2008). Characterization of scientific workflows. *Workflows in Support of Large-Scale Science*.
- Bochenina, K. (2014). A Comparative Study of Scheduling Algorithms for the Multiple Deadline-constrained Workflows in Heterogeneous Computing Systems with Time Windows. *Procedia Computer Science* 29, 509-522.
- Butakov, N. a. (2014). Co-evolutional genetic algorithm for workflow scheduling in heterogeneous distributed environment. *Application of Information and Communication Technologies (AICT)*.
- Buttazzo, G. C. (2005). Rate monotonic vs. EDF: judgment day. *Real-Time Systems* 29.1, 5-26.

- Mao, M. a. (2011). Auto-scaling to minimize cost and meet application deadlines in cloud workflows. *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*. ACM.
- Nasonov, D. a. (2014). Hybrid Scheduling Algorithm in Early Warning Systems. *Procedia Computer Science* 29, 1677-1687.
- Nasonov, D. B. (2014). Hybrid Evolutionary Workflow Scheduling Algorithm for Dynamic Heterogeneous Distributed Computational Environment. *International Joint Conference SOCO'14-CISIS'14-ICEUTE'14*.
- Nebro, A. J. (2009). Mocell: A cellular genetic algorithm for multiobjective optimization. *International Journal of Intelligent Systems* 24.7, 726-746.
- Sinnen, O. (2007). Task scheduling for parallel systems. *Wiley-Interscience*, 108.
- Son, J. H. (1999). Hard/soft deadline assignment for high workflow throughput. *Database Applications in Non-Traditional Environments*.
- Yazir, Y. O. (2010). Dynamic resource allocation in computing clouds using distributed multiple criteria decision analysis. *Cloud Computing (CLOUD)*.
- Yeo, C. S. (2005). Service level agreement based allocation of cluster resources: Handling penalty to enhance utility. *Cluster Computing*. IEEE International.
- Yu, J. a. (2006). A budget constrained scheduling of workflow applications on utility grids using genetic algorithms. *Workflows in Support of Large-Scale Science*.
- Yu, J. M. (2007). Multi-objective planning for workflow execution on grids. *Proceedings of the 8th IEEE/ACM International conference on Grid Computing*.
- Yu, J. R. (2008). Workflow scheduling algorithms for grid computing. *Metaheuristics for scheduling in distributed computing environments*, 173-214.