An Approximative Criterion for the Potential of Energetic Reasoning*

Timo Berthold[1], Stefan Heinz[1], and Jens Schulz[2]

[1] Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany
{berthold,heinz}@zib.de
[2] Technische Universität Berlin, Institut für Mathematik, Straße des 17. Juni 136, 10623 Berlin, Germany
jschulz@math.tu-berlin.de

Abstract. Energetic reasoning is one of the most powerful propagation algorithms in cumulative scheduling. In practice, however, it is commonly not used because it has a high running time and its success highly depends on the tightness of the variable bounds. In order to speed up energetic reasoning, we provide a necessary condition to detect infeasibilities, which can be tested efficiently.

We present an implementation of energetic reasoning that employs this condition and that can be parametrically adjusted to handle the trade-off between solving time and propagation overhead. Computational results on instances from the PSPLib are provided. They show that using the condition decreases the running time by more than a half, although more search nodes need to be explored.

1 Introduction

Many real-world scheduling problems rely on cumulative restrictions. In this paper, we consider a cumulative scheduling problem with non-preemptable jobs and fix resource demands. Such a problem is determined by earliest start and latest completion times to all jobs, the resource demands, and a resource capacity for each resource. Besides that precedence constraints between different jobs might be present. The goal is to find start times for each job, a *schedule*, such that the cumulative demands do not exceed the capacities and the precedence requirements are satisfied.

Computing such a schedule is known to be strongly $\mathcal{NP}$-hard [3].

Several exact approaches were developed that solve the problem by branch-and-bound, using techniques from constraint programming, integer programming, or satisfiability testing. In constraint programming, the

* Supported by the DFG Research Center Matheon *Mathematics for key technologies* in Berlin.

main task is to design efficient *propagation algorithms* that adjust variable bounds or detect infeasibility of a search node, in order to keep the search tree small. Such algorithms are usually executed more than once per search node. The most powerful and widely used algorithms in cumulative scheduling are time-tabling, edge-finding, and energetic reasoning.

This paper concentrates on the evaluation of the energetic reasoning algorithm. Its merit lies in a drastic reduction of the number of search nodes by detecting infeasible nodes early. It has, however, a cubic running time in the number of jobs and is only capable to find variable bound adjustments for rather tight variable bounds.

Related work. Baptiste et.al. [1] provide a detailed overview on the main constraint programming techniques for cumulative scheduling. Therein, several theoretical properties of energetic reasoning are proven. A more general idea of *interval capacity consistency tests* is given by Dorndorf et.al. [4]. In the same paper, unit-size intervals are considered as a special case, which leads to the time-tabling algorithm [6]. Recently, Kooli et.al. [7] used integer programming techniques in order to improve the energetic reasoning algorithm. This approach extends the method presented by Hidri et.al. [5], where the parallel machine scheduling problem has been considered. In both works only infeasibility of a subproblem is checked; variable bound adjustments are not performed.

Contribution. We derive a necessary condition for energetic reasoning to detect infeasibilities. The condition is based on a *relative energy histogram*, which can be computed efficiently. We show that this histogram underestimates the true energy requirement of an interval by a factor of at most $1/3$. We embed this approximative result in a parametrically adjustable propagation algorithm which detects variable bound adjustments and infeasibilities in the same run.

As our computational results reveal, the presented algorithm drastically reduces the total computation time for solving instances from the PSPLib [8] in contrast to the pure energetic reasoning algorithm.

Outline. We introduce the resource-constrained project scheduling problem (RCPSP) and the general idea of energetic reasoning in Section 2. In Section 3 we derive a necessary condition for energetic reasoning to be successful and embed it into a competitive propagation algorithm. Experimental results on instances from PSPLib [8] are presented in Section 4.

2 Problem description and Energetic Reasoning

In resource-constrained project scheduling (RCPSP) we are given a set $\mathcal{J}$ of non-preemptable jobs and a set $\mathcal{R}$ of renewable resources. Each resource $k \in \mathcal{R}$ has bounded capacity $C_k \in \mathbb{N}$. Every job $j \in \mathcal{J}$ has a processing time $p_j \in \mathbb{N}$ and resource demands $r_{jk} \in \mathbb{N}$ for each resource $k \in \mathcal{R}$. The start time S_j of job j is constrained by its predecessors that are given by a precedence graph $D = (V, A)$ with $V = \mathcal{J}$. An arc $(i, j) \in A$ represents a precedence relationship, i.e., job i must be finished before job j starts. The goal is to schedule all jobs with respect to resource and precedence constraints, such that the *makespan*, i.e., the latest completion time of all jobs, is minimized.

The RCPSP can be modeled as a constraint integer program as follows:

$$\begin{aligned} \min \quad & \max_{j \in \mathcal{J}} S_j + p_j & & \\ \text{subject to} \quad & S_i + p_i \leq S_j & \text{for all}\,(i,j) \in A \quad & (1) \\ & \texttt{cumulative}(\boldsymbol{S}, \boldsymbol{p}, \boldsymbol{r}_{.k}, C_k) & \forall\, k \in \mathcal{R} \quad & (2) \end{aligned}$$

The constraints (1) represent the *precedence* conditions. The *cumulative* constraints (2) enforce that at each point in time t, the cumulated demand of the set of jobs running at that point, does not exceed the given capacities, i.e.,

$$\sum_{j \in \mathcal{J}: t \in [S_j, S_j + p_j[} r_{jk} \leq C_k \qquad \text{for all } k \in \mathcal{R}.$$

Energetic reasoning is a technique to detect infeasibility or to adjust variable bounds for one cumulative constraint $k \in \mathcal{R}$, based on the amount of work that must be executed in a specified time interval. The term energetic reasoning has been defined for partially or fully elastic scheduling problems [1]. This procedure is also known as *Left-Shift/Right-Shift* technique in case of cumulative scheduling with non-interruptible jobs. Given an upper bound on the latest completion time of all jobs, we obtain earliest start times est_j, earliest completion times ect_j, latest start times lst_j, and latest completion times lct_j for each job $j \in \mathcal{J}$. The *required energy* $E(a, b)$ of all jobs in interval $[a, b[$ is given by $E(a, b) := \sum_{j \in \mathcal{J}} e_j(a, b)$, with

$$e_j(a, b) := \max\{0, \min\{b - a, p_j, \mathrm{ect}_j - a, b - \mathrm{lst}_j\}\} \cdot r_j.$$

Hence, $e_j(a, b)$ is the non-negative minimum of (i) the volume in the interval $[a, b[$, (ii) the energy of job j, (iii) the left shifted energy, and (iv)

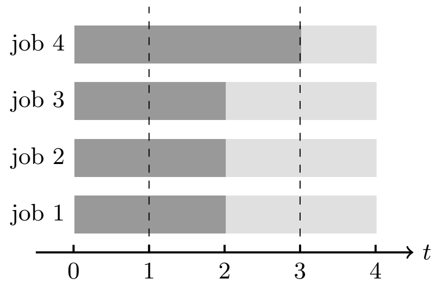


Fig. 1. Problem setup of Example 1.

the right shifted energy. With respect to $e_j(a,b)$, a problem is infeasible if more energy is required than available, so for $a < b$ and a resource capacity C we can deduce:

Corollary 1 ([1]). *If $E(a,b) > (b-a) \cdot C$, then the problem is infeasible.*

Example 1. Consider a cumulative resource of capacity 2 and four jobs each with a resource demand of 1, an earliest start time of 0 and a latest completion time of 4. Three of these jobs have a processing time of 2. The fourth job instead has a processing time of 3. Figure 1 illustrates this setup. For the interval $[1,3[$, the available energy is $(3-1)\cdot 2 = 4$. The first three jobs contribute 1 units each where as the fourth jobs adds 2 to the required energy. This sums up to $E(1,3) = 5$. This shows that that these jobs cannot be scheduled.

In order to detect infeasibility, $O(n^2)$ time-intervals need to be considered [1]. Those intervals are determined in the following way:

$$\begin{aligned}
O_1 &:= \bigcup_j \big(\{\mathrm{est}_j\} \cup \{\mathrm{est}_j + p_j\} \cup \{\mathrm{lct}_j - p_j\}\big), \\
O_2 &:= \bigcup_j \big(\{\mathrm{lct}_j\} \cup \{\mathrm{est}_j + p_j\} \cup \{\mathrm{lct}_j - p_j\}\big) \text{ and} \\
O(t) &:= \bigcup_j \{\mathrm{est}_j + \mathrm{lct}_j - t\}.
\end{aligned}$$

The relevant intervals to be checked for energetic tests are given by $(a,b) \in O_1 \times O_2$ and for fixed $a \in O_1 : (a,b) \in O_1 \times O(a)$ and for fixed $b \in O_2$: $(a,b) \in O(b) \times O_2$. These are $O(n^2)$ many.

Besides detecting infeasibilities, variable bounds can be adjusted by energetic reasoning. Due to symmetry reasons, we just consider the adjustments of lower bounds. Let

$$e_j^{\text{left}}(a,b) := r_j \cdot \max\{0, \min\{p_j, b-a, \min\{b, \text{ect}_j\} - \max\{a, \text{est}_j\}\}\}$$

be the required energy in the interval $[a,b[$ of job j if it is *left-shifted*, i.e., it starts as early as possible. If $[a,b[$ intersects with $[\text{est}_j, \text{ect}_j[$ and the required energy $E(a,b) - e_j(a,b) + e_j^{\text{left}}(a,b)$ exceeds the available energy in $[a,b[$, then j cannot start at its earliest start time and the lower bound can be updated according to Theorem 1.

Theorem 1 ([1]). *Let $[a,b[, a < b$ and $j \in \mathcal{J}$ with $[\text{est}_j, \text{ect}_j[\cap [a,b[\neq \emptyset$. If $E(a,b) - e_j(a,b) + e_j^{\text{left}}(a,b) > (b-a) \cdot C$ holds, then the earliest start time of job j can be updated to*

$$\text{est}_j = a + \left\lceil \frac{1}{r_j} \left(E(a,b) - e_j(a,b) - (b-a) \cdot (C - r_j)\right) \right\rceil.$$

In case of feasibility tests, we are able to restrict the set of intervals that need to be considered. Whether such restrictions can also be made for variable bound adjustments is an open problem. The currently fastest energetic reasoning propagation algorithm runs in $O(n^3)$.

3 Restricted Energetic Reasoning

Energetic reasoning only finds variable bound adjustments when the bounds of the start time variables are tight, since it compares the available energy to the requested energy for some intervals. If the bounds are loose and small intervals are considered, a job may contribute almost no energy to that interval or in case of large intervals not enough energy is required in order to derive any adjustments. This is a clear drawback as we are faced with a very time-consuming algorithm. In order to come up with a practical competitive propagation algorithm, we identify intervals that seem promising to detect infeasibilities and variable bound adjustments.

3.1 Estimation of relevant intervals

Let us consider one resource with capacity C, and cumulative demands r_j for each job j. The total energy requirement of job j is given by $e_j = p_j \cdot r_j$. We measure the *relative energy consumption* $\tilde{e}_j := \frac{e_j}{\text{lct}_j - \text{est}_j}$.

We define the *relative energy histogram* $\tilde{E} : \mathbb{N} \to \mathbb{R}$ and the *relative energy* $\tilde{E}(a,b)$ of an interval $[a,b[$ by:

$$\tilde{E}(t) := \sum_{j:\text{est}_j \leq t < \text{lct}_j} \frac{e_j}{\text{lct}_j - est_j} \qquad \text{and} \qquad \tilde{E}(a,b) := \sum_{t=a}^{b-1} \tilde{E}(t).$$

This histogram approximates the required energy $E(a,b)$ computed by energetic reasoning for each point in time, as we prove in Theorem 2.

Theorem 2. *Given an arbitrary non-empty interval* $[a,b[$. *Then*

$$\alpha \cdot E(a,b) \le \tilde{E}(a,b)$$

with $\alpha > 1/3$.

Proof. We show the approximation factor α for each job separately. By linearity of summation, the theorem follows.

First, we restrict the study to the case where $\mathrm{est}_j \le a < b \le \mathrm{lct}_j$. Let

$$\tilde{e}_j(a,b) = \frac{p_j \cdot r_j}{\mathrm{lct}_j - est_j} \cdot (\min\{\mathrm{lct}_j, b\} - \max\{\mathrm{est}_j, a\}).$$

If the energy is underestimated in $[a,b[$, then it follows that $\mathrm{est}_j < a$ or $\mathrm{lct}_j > b$, since otherwise $\tilde{e}_j(a,b) = e_j(a,b)$. Assume $\mathrm{est}_j \le a < \mathrm{lct}_j < b$. By definition, $e_j(a,b) = e_j(a,\mathrm{lct}_j)$ and $\tilde{e}_j(a,b) = \tilde{e}_j(a,\mathrm{lct}_j)$. Applying a symmetrical argument to $a < \mathrm{est}_j < b \le \mathrm{lct}_j$, we can restrict the setting to $\mathrm{est}_j \le a < b \le \mathrm{lct}_j$, such that $\tilde{e}_j(a,b) = p_j r_r (b-a)/(\mathrm{lct}_j - \mathrm{est}_j)$. Note that in this case the energy gets underestimated, i.e., $0 < \tilde{e}_j(a,b) < e_j(a,b)$.

Case 1. Consider the case $e_j(a,b) = p_j \cdot r_j$. That means the job is fully contained in $[a,b[$. This is a contradiction to $\tilde{e}_j(a,b) < e_j(a,b)$.

Case 2. Assume the following two properties:

(i) $1 \le \min\{\mathrm{ect}_j - a, b - \mathrm{lst}_j\} < \min\{b-a, p_j\}$
(ii) $e_j(a,b) = \min\{\mathrm{ect}_j - a, b - \mathrm{lst}_j\} \cdot r_j$.

Thus, $\alpha' := \tilde{e}_j(a,b)/e_j(a,b)$ yields:

$$\alpha' = \frac{p_j(b-a)}{(\mathrm{lct}_j - \mathrm{est}_j) \cdot \min\{\mathrm{ect}_j - a, b - \mathrm{lst}_j\}} > \frac{\max\{p_j, b-a\}}{\mathrm{lct}_j - \mathrm{est}_j}.$$

Minimizing α' with respect to $1 \le \min\{\mathrm{ect}_j - a, b - \mathrm{lst}_j\}$ yields $b-a = k$ and $p_j := k+1$ for some $k \in \mathbb{N}$, such that $\alpha' = \max\{k+1, k\}/(3k) > 1/3$.

Case 3. Finally, consider the case $b - a < \min\{p_j, \mathrm{ect}_j - a, b - \mathrm{lst}_j\}$. Thus, $e_j(a,b) = (b-a)r_j$. That means, the job is completely executed in $[a,b[$, i.e., $[a,b[\subseteq [\mathrm{lst}_j, \mathrm{ect}_j[$. This yields the condition $\mathrm{ect}_j \ge \mathrm{lst}_j + (b-a)$, which is equivalent to $2p_j - (b-a) \ge \mathrm{lct}_j - \mathrm{est}_j$. Thus,

$$\tilde{e}_j(a,b) = \frac{p_j \cdot r_j}{\mathrm{lct}_j - \mathrm{est}_j}(b-a) \ge \frac{p_j}{2p_j - (b-a)} r_j \cdot (b-a) = \underbrace{\frac{1}{2 - \frac{b-a}{p_j}}}_{=:\alpha''} e_j(a,b).$$

We obtain $\alpha := \min\{\alpha', \alpha''\} > 1/3$. □

The proof shows that an underestimation of $E(a, b)$ happens if the *core* of a job, i.e., $[\mathrm{lst}_j, \mathrm{ect}_j[$, overlaps that interval or if a job is associated with a large interval $[\mathrm{est}_j, \mathrm{lct}_j[$ and intersects just slightly with $[a, b[$. The following corollary states the necessary condition that we use in our propagation algorithm.

Corollary 2. *Energetic reasoning cannot detect any infeasibility, if either of the following conditions hold*

(i) for all $[a, b[, a < b$, $\tilde{E}(a, b) \leq \frac{1}{3}(b - a)\, C$
(ii) for all t*:* $\tilde{E}(t) \leq \frac{1}{3}\, C$.

The histogram $\tilde{E}$ can be computed in $O(n \log n)$ by first sorting the earliest start times and latest completion times of all jobs and then creating the histogram chronologically from earliest event to latest event. Since there are $O(n)$ event points (the start and completion times of the jobs) only $O(n)$ changes in the histogram need to be stored.

3.2 Restricted Energetic Reasoning Propagation Algorithm

We now present a restricted version of energetic reasoning which is based on the results of the previous section. Due to Theorem 2, only intervals $[a, b[$ containing points in time t with $\tilde{E}(t) > 1/3\, C$ need to be checked. Note that the cardinality of this set may still be cubic in the number of jobs. We introduce an approach, in which we only execute the energetic reasoning algorithm on interval $[t_1, t_2[$ if

$$\forall\, t \in [t_1, t_2[: \qquad \tilde{E}(t) > \alpha \cdot C$$

holds. For given $\tilde{E}$, this condition can be checked in $O(n)$. If it holds, we check each pair $(a, b) \in O_1 \times O_2$ with $[a, b[\subseteq [t_1, t_2[$ in order to detect infeasibility or to find variable bound adjustments.

The procedure is captured in Algorithm 1. Here only the propagation of lower bounds is shown, upper bound adjustments work analogously.

As mentioned before, the relative energy histogram $\tilde{E}(t)$ can be computed in $O(n \log n)$ and needs $O(n)$ space. The sets O_1 and O_2 also need $O(n)$ space and are sorted in $O(n \log n)$. Loops 5 and 10 together consider at most all $O(n^2)$ intervals $O_1 \times O_2$. Loop 13 will consider at most $O(n)$ jobs. Since $E(a, b)$ is precomputed in line 11 and all other calculations can be done in constant time, we are able to bound the total running time.

Corollary 3. *Algorithm 1 can be implemented in* $O(n^3)$.

Algorithm 1: Restricted Energetic Reasoning propagation algorithm for lower bounds.

Input: Resource capacity C, set $\mathcal{J}$ of jobs, and a scaling factor α.
Output: Earliest start times est'_j for each job j or an infeasibility is detected.

Create relative energy histogram $\tilde{E}$.
Compute and sort sets O_1 and O_2.
if $\tilde{E}(t) \leq \alpha \cdot C$ *for all* t **then**
  stop.
forall *event points* t *in increasing order* **do**
  if $\tilde{E}(t) \leq \alpha \cdot C$ **then**
    continue.
  $t_1 := t$.
  Let t_2 be the first event point after t with $\tilde{E}(\tau) \leq \alpha \cdot C$.
  forall $(a, b) \in O_1 \times O_2 :\ [a, b[\subseteq [t_1, t_2[$ **do**
    if $E(a, b) > (b - a) \cdot C$ **then**
      stop: infeasible.
    forall *jobs* j *with* $[a, b[\cap[\mathrm{est}_j, \mathrm{ect}_j[\neq \emptyset$ **do**
      if $E(a, b) - e_j(a, b) + e_j^{\mathrm{left}}(a, b) > (b - a) \cdot C$ **then**
        $\mathrm{rest} := E(a, b) - e_j(a, b) - (b - a) \cdot (C - r_j)$.
        $\mathrm{est}'_j := a + \lceil \mathrm{rest}\, / r_j \rceil$.
      if $\mathrm{est}'_j > \mathrm{lst}_j$ **then**
        stop: infeasible.
  $t := t_2$.

Asymptotically, it has the same running time as pure energetic reasoning, but the constants are much smaller. Compared to the pure energetic reasoning algorithm we only consider large intervals if the relative energy consumption is huge over a long period. The savings in running time and further influences on the solving process will be discussed in the following Section.

4 Computational results

We performed our computational results on the RCPSP test sets $j30$ and $j60$ from the PSPLIB [8]. Each test set contains 480 instances with 30 or 60 jobs per instance. We use the default settings of our implementation of the cumulative constraint as presented in [2] with SCIP version 1.2.1.5 and integrated CPLEX release version 12.10 as underlying LP solver. The only scheduling specific propagation algorithm used is energetic reasoning and its parametric variants using the necessary condition from Corollary (2). A time limit of one hour was enforced for each instance. All computations were obtained on Dual QuadCore Xeon X5550 2.67 GHz computers (in 64 bit mode), running Linux, and 24 GB of main memory.

Parameter settings. According to Theorem 2, it suffices to check for $\alpha > 1/3$, whereas α close to $1/3$ would end up in checking most of the intervals as energetic reasoning does. Due to the outcome we show results for $\alpha \in \{0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2\}$. Starting with an initial setting '0.0', i.e., the pure energetic reasoning, we evaluate the different parameter settings of α, yielding settings from '0.5' to '1.2'.

Evaluation of all instances. Table 1 shows aggregated computational results for all instances from the test sets $j30$ and $j60$ that are solved by at least one algorithm. The pure energetic reasoning algorithm, which corresponds to a factor $\alpha = 0.0$ serves as reference solver, '0.0'. For $j30$, 473 instances can be solved by at least one solver, pure energetic reasoning has eight timeouts on these, see column 'limit'. Choosing $\alpha = 0.8$ or 0.9 solves all 473 instances, whereas a smaller or larger value decreases the number of solved instances. Column 'better' tells how many instances are solved more than 10% faster than by the reference solver. Respectively, 'worse' expresses how often a solver is more than 10% slower than the reference setting '0.0'. Here, $\alpha = 0.9$ performs best. Since some instances timed out, we show how often better ('bobj') or worse ('wobj') primal bounds are found. Using a weak propagation ($\alpha = 1.2$) yields worst results since 30 instances are not solved, 110 are more than 10% slower and for 13 instances a worse primal bound is found.

For test set $j60$, best results are gained for $\alpha = 0.9$. Though, there is one more timeout than by the reference solver, in twice as many cases the restricted version is at least 10% faster than pure energetic reasoning.

Evaluation of all optimal solved instances. Since many instances are unsolved or trivial, Table 2 presents the results only on those instances that (i) could be solved to optimality by all solvers, (ii) at least one solver needed more that one search node, and (iii) at least one solver needed more that one second of computation time. That means, we exclude all easy instances and those which none of the solver was able to solve. In case of the test sets $j30$ and $j60$ we are left with 112 and 32 instances, respectively.

Columns 'better' and 'worse' reveal that values $\alpha = 0.9$ and 1.0 are dominating. Column 'shnodes' and 'shtime' state the shifted geomet-

Table 1. Overview for 473 instances from $j30$ and 397 instances from $j60$. Only those instances are considered that are solved by at least one solver.

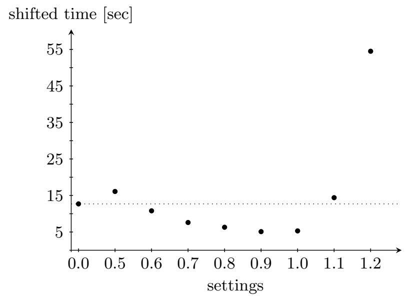
	j30						j60					
α	solved	limit	better	worse	bobj	wobj	solved	limit	better	worse	bobj	wobj
0.0	465	8	–	–	–	–	393	4	–	–	–	–
0.5	467	6	69	77	3	1	387	10	43	45	1	6
0.6	471	2	81	64	3	0	388	9	45	43	1	5
0.7	472	1	89	53	3	0	388	9	49	37	1	4
0.8	473	0	94	51	3	0	391	6	55	34	1	3
0.9	473	0	105	40	3	0	392	5	59	30	1	4
1.0	472	1	102	40	3	1	392	5	57	31	1	3
1.1	465	8	72	84	3	4	375	22	45	39	1	19
1.2	443	30	52	110	3	13	358	39	38	56	1	35

Table 2. Overview on those instances (i) which are solved with all settings, (ii) where at least one solver needed more than one search node, and (iii) at least one solver needed more than one second. This results in 112 instances for the $j30$ test set and 32 for the $j60$ test set.

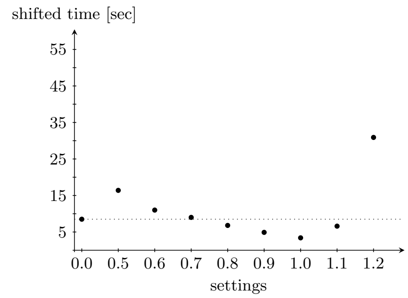
	j30				j60			
α	better	worse	shnodes	shtime	better	worse	shnodes	shtime
0.0	–	–	314	12.7	–	–	171	8.5
0.5	49	54	1664	16.1	8	17	1098	16.4
0.6	54	45	1676	10.8	10	15	1005	11.0
0.7	59	33	1679	7.6	11	13	1102	9.0
0.8	60	35	1883	6.3	14	11	1237	6.8
0.9	66	28	2174	5.1	17	9	1240	4.9
1.0	66	24	2895	5.3	16	9	1274	3.4
1.1	47	56	10336	14.4	15	9	2271	6.6
1.2	34	74	52194	54.4	10	20	18050	30.9

ric mean[3] of all nodes and of the running time, respectively. Table 2 shows that the pure energetic reasoning needs by far the fewest number of nodes. For each instance of the parametric algorithm the number of nodes increases by at least a factor of 5. The more we relax the value of α from 0.5 to 1.2, the more nodes are needed. Besides that, a weak propagation ($\alpha = 1.2$) behaves worst in all columns. The best running times are gained with values 0.9 and 1.0 for α. In these cases the restricted energetic reasoning was more than twice as fast as the pure energetic reasoning algorithm. This is illustrated in Figure 2, which shows how the shifted geometric running times vary with different values of α.

[3] The shifted geometric mean of values $t_1, \dots, t_n$ is defined as $\left(\prod(t_i + s)\right)^{1/n} - s$ with shift s. We use a shift $s = 10$ for time and $s = 100$ for nodes in order to decrease the strong influence of the very easy instances in the mean values.



(a) Running times of test set $j30$.



(b) Running times of test set $j60$.

Fig. 2. Comparison of running times from Table 2.

5 Conclusions

We presented a necessary condition for energetic reasoning to detect infeasibilities or to derive variable bound adjustments. This result was incorporated into a parametrical adjustable version of energetic reasoning. By checking this condition, we only apply this powerful but expensive algorithm, when the estimated energy is above a certain threshold α.

Computational results revealed that choosing α close to 1.0 can speed up the search by a factor of two though the number of nodes drastically increases.

References

1. P. Baptiste, C. Le Pape, and W. Nuijten, *Constraint-based scheduling: applying constraint programming to scheduling problems*, International Series in Operations Research & Management Science, 39, Kluwer Academic Publishers, Boston, MA, 2001.
2. T. Berthold, S. Heinz, M. E. Lübbecke, R. H. Möhring, and J. Schulz, *A constraint integer programming approach for resource-constrained project scheduling*, in Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, A. Lodi, M. Milano, and P. Toth, eds., vol. 6140 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2010, pp. 313–317.
3. J. Blazewicz, J. K. Lenstra, and A. H. G. R. Kan, *Scheduling subject to resource constraints: classification and complexity*, Discrete Applied Mathematics, 5 (1983), pp. 11 – 24.
4. U. Dorndorf, T. Phan-Huy, and E. Pesch, *A survey of interval capacity consistency tests for time- and resource-constrained scheduling*, J. Weglarz, Kluwer Academic, Boston, MA, 1999, ch. 10, pp. 213–238.
5. L. Hidri, A. Gharbi, and M. Haouari, *Energetic reasoning revisited: application to parallel machine scheduling*, Journal of Scheduling, 11 (2008), pp. 239–252.

6. R. Klein and A. Scholl, *Computing lower bounds by destructive improvement: An application to resource-constrained project scheduling*, Eur. J. Oper. Res., 112 (1999), pp. 322–346.
7. A. Kooli, M. Haouari, L. Hidri, and E. Néron, *Ip-based energetic reasoning for the resource constrained project scheduling problem*, Electronic Notes in Discrete Mathematics, 36 (2010), pp. 359 – 366. ISCO 2010 - International Symposium on Combinatorial Optimization.
8. PSPLib, *Project Scheduling Problem LIBrary*. http://129.187.106.231/psplib/. Last accessed 2010/Feb/01.