

TIMO BERTHOLD STEFAN HEINZ JENS SCHULZ**

An Approximative Criterion for the Potential of Energetic Reasoning*

* Supported by the DFG Research Center MATHEON *Mathematics for key technologies* in Berlin.
** Technische Universität Berlin, Institut für Mathematik, Straße des 17. Juni 136, 10623 Berlin, Germany

An Approximative Criterion for the Potential of Energetic Reasoning^{*}

Timo Berthold¹, Stefan Heinz¹, and Jens Schulz²

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 not commonly 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 an easy-to-check necessary condition for energetic reasoning to detect infeasibilities.

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. These results show that using this 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 constraints are satisfied. Computing such a schedule is known to be strongly \mathcal{NP} -hard [1].

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, see [2].

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. [2] 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. [3]. In the same paper, unit-size intervals are considered as a special case, which leads to the time-tabling algorithm [4]. Recently, Kooli et.al. [5] used integer programming techniques in order to improve the energetic reasoning algorithm. This approach extends the method presented by Hidri et.al. [6], 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 [7] 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 [7] 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 program:

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

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.,

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

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 [2]. This procedure is also known as *Left-Shift/Right-Shift* technique in case of cumulative scheduling with non-interruptible jobs.

Due to the precedence constraints and 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 $j \in \mathcal{J}$. Since the propagation algorithm is used during branch-and-bound search, we usually refer to lower bounds (corresponding to est_j) and upper bounds (corresponding to lct_j) of the start time variable S_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

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

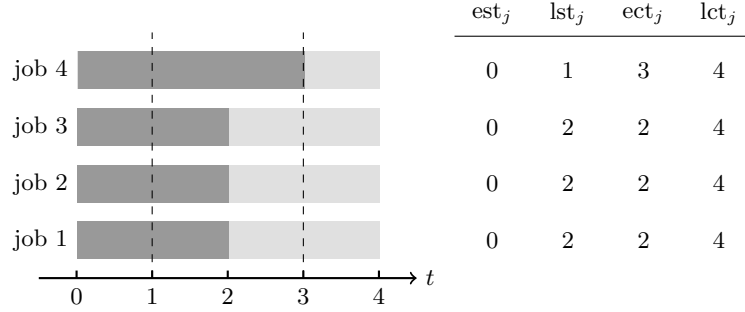


Fig. 1. Problem setup of Example 1.

Hence, $e_j(a, b)$ is the non-negative minimum of (i) the available energy in the interval $[a, b]$, i.e., $(b - a) \cdot r_j$, (ii) the energy of job j , i.e., $p_j \cdot r_j$, (iii) the left shifted energy, i.e., $(ect_j - a) \cdot r_j$ and (iv) the right shifted energy, i.e., $(b - lst_j) \cdot r_j$. Throughout the paper, we assume that intervals $[a, b]$ are non-empty, i.e., that $a < b$. With respect to $e_j(a, b)$, a problem is infeasible if more energy is required than available. For $a < b$ and a resource capacity C we can deduce:

Corollary 1 ([2]). *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 has a processing time of 3 instead. Figure 1 illustrates this setup. For the interval $[1, 3)$, the available energy is $(3 - 1) \cdot 2 = 4$. The first three jobs contribute one unit each, whereas the fourth job adds two units to the required energy. This sums up to $E(1, 3) = 5$. This shows that these jobs cannot be scheduled.

In order to detect infeasibility, $O(n^2)$ time-intervals need to be considered [2]. These intervals correspond to the start and completion times of the jobs and are precisely determined in the following way:

$$\begin{aligned}
 O_1 &:= \bigcup_j (\{est_j\} \cup \{est_j + p_j\} \cup \{lct_j - p_j\}), \\
 O_2 &:= \bigcup_j (\{lct_j\} \cup \{est_j + p_j\} \cup \{lct_j - p_j\}) \text{ and} \\
 O(t) &:= \bigcup_j \{est_j + lct_j - t\}.
 \end{aligned}$$

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

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

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

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 est_j of S_j can be updated according to Theorem 1 which was proved in Baptiste et.al. [2].

Theorem 1 (Baptiste et.al. [2]). *Let $[a, b)$ with $a < b$ 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*

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

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 compares the available energy to the requested energy for certain intervals. Therefore, it is more likely to detect variable bound adjustments if the bounds are tight, i.e., the interval $[\text{est}_j, \text{lst}_j]$, from which to choose S_j , is small. 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* $\tilde{E} : \mathbb{N} \rightarrow \mathbb{R}$ and the *relative energy* $\tilde{E}(a, b)$ of an interval $[a, b)$ by:

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

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. *Let an arbitrary non-empty interval $[a, b)$ be given. Then*

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

with $\alpha > 1/3$.

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

First, we show that we can restrict the study to the case where $\text{est}_j \leq a < b \leq \text{lct}_j$. Therefore, let

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

If the energy is underestimated in $[a, b)$, then it follows that $\text{est}_j < a$ or $\text{lct}_j > b$, since otherwise $\tilde{e}_j(a, b) = e_j(a, b)$. Assume $\text{est}_j \leq a < \text{lct}_j < b$. Then, $e_j(a, b) = e_j(a, \text{lct}_j)$ and $\tilde{e}_j(a, b) = \tilde{e}_j(a, \text{lct}_j)$ holds. Applying a symmetrical argument to $a < \text{est}_j < b \leq \text{lct}_j$, we can restrict the setting to $\text{est}_j \leq a < b \leq \text{lct}_j$, such that $\tilde{e}_j(a, b) = p_j \cdot r_j \cdot (b - a) / (\text{lct}_j - \text{est}_j)$. 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 the fact that $\tilde{e}_j(a, b) < e_j(a, b)$.

Case 2. Assume the following two properties:

- (i) $1 \leq \min\{\text{est}_j - a, b - \text{lct}_j\} < \min\{b - a, p_j\}$
- (ii) $e_j(a, b) = \min\{\text{est}_j - a, b - \text{lct}_j\} \cdot r_j$.

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

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

Minimizing α' with respect to $1 \leq \min\{\text{ect}_j - a, b - \text{lst}_j\}$ 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 $b - a < \min\{p_j, \text{ect}_j - a, b - \text{lst}_j\}$. Thus, $e_j(a, b) = (b - a) \cdot r_j$. That means, the job is completely executed in $[a, b)$, i.e., $[a, b) \subseteq [\text{lst}_j, \text{ect}_j)$. This yields the condition $\text{ect}_j \geq \text{lst}_j + (b - a)$, which is equivalent to $2p_j - (b - a) \geq \text{lct}_j - \text{est}_j$. Thus,

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

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., $[\text{lst}_j, \text{ect}_j)$, overlaps this interval or if a job is associated with a large interval $[\text{est}_j, \text{lct}_j)$ 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 one of the following conditions holds*

- (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

Algorithm 1: Restricted energetic reasoning propagation algorithm for lower bounds.

Input: Resource capacity C , set \mathcal{J} of jobs with earliest start times est_j , and a scaling factor α .

Output: Earliest start times est'_j for each job j or an infeasibility is detected.

- 1 Create relative energy histogram \tilde{E} .
- 2 Compute and sort event points and sets O_1 and O_2 .
- 3 **forall** jobs j **do**
- 4 $\text{est}'_j := \text{est}_j$.
- 5 **forall** event points t in increasing order **do**
- 6 **if** $\tilde{E}(t) \leq \alpha \cdot C$ **then**
- 7 continue .
- 8 $t_1 := t$.
- 9 Let t_2 be the first event point after t with $\tilde{E}(t_2) \leq \alpha \cdot C$.
- 10 **forall** $(a, b) \in O_1 \times O_2 : [a, b] \subseteq [t_1, t_2]$ **do**
- 11 **if** $E(a, b) > (b - a) \cdot C$ **then**
- 12 stop: infeasible .
- 13 **forall** jobs j with $[a, b] \cap [\text{est}_j, \text{ect}_j] \neq \emptyset$ **do**
- 14 **if** $E(a, b) - e_j(a, b) + e_j^{\text{left}}(a, b) > (b - a) \cdot C$ **then**
- 15 $\text{leftover} := E(a, b) - e_j(a, b) - (b - a) \cdot (C - r_j)$.
- 16 $\text{est}'_j := \max\{\text{est}'_j, a + \lceil \text{leftover} / r_j \rceil\}$.
- 17 **if** $\text{est}'_j > \text{lst}_j$ **then**
- 18 stop: infeasible .
- 19 $t := t_2$.

jobs. We introduce an approach, in which we only execute the energetic reasoning algorithm on interval $[t_1, t_2]$ if

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

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 runs over at most $O(n)$ jobs. The computed value for $E(a, b)$ in line 11 can be used in the remain-

ing inner loops and all other calculations can be done in constant time, such that we are able to bound the total running time.

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

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 experiments on the RCPSP test sets J30 and J60 from the PSPLIB [7]. Each test set contains 480 instances with 30 and 60 jobs per instance, respectively. The implementation was done in SCIP version 1.2.1.5, which integrates CPLEX release version 12.1.0 as underlying LP solver. We used the implementation of the cumulative constraint presented in [8].

The only scheduling specific propagation algorithm used was 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), 24 GB of main memory, running a Linux system.

Parameter settings. According to Theorem 2, it suffices to consider only $\alpha > 1/3$. Choosing a value close to $1/3$, however, results in checking the vast majority of the intervals, similar to energetic reasoning. To evaluate the impact of different values of α , we ran the algorithm with $\alpha \in \{0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2\}$. For comparison, we further show results for $\alpha = 0.0$ which refers to pure energetic reasoning.

Evaluation of all instances. Table 1 shows aggregated computational results for all instances from the test sets J30 and J60 that were solved by at least one algorithm. These were 473 and 397 for J30 and J60, respectively. In both cases the pure energetic reasoning algorithm, which corresponds to $\alpha = 0.0$, serves as reference solver. For J30, pure energetic reasoning failed to compute a proven optimal solution on eight out of the 473 instances. This is shown in the column “outs” while column “solved” displays the number of instances solved to proven optimality.

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.

α	J30						J60					
	solved	outs	better	worse	bobj	wobj	solved	outs	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

Choosing $\alpha = 0.8$ or 0.9 solved all 473 instances, whereas a smaller or larger value decreased the number of solved instances. Column “better” tells how many instances were solved more than 10% faster than the reference solver (pure energetic reasoning). Accordingly, “worse” expresses how often a solver was more than 10% slower than the reference setting $\alpha = 0.0$. Here, choosing $\alpha = 0.9$ performed best. Since some instances timed out, we show how often better (“bobj”) or worse (“wobj”) primal bounds were found. Using the weak propagation factor $\alpha = 1.2$ yielded worst results. In this case, 30 instances could not be solved, 110 were more than 10% slower, and 13 instances had a worse primal bound compared to the reference solver.

For the test set J60, the results are similar, except that pure energetic reasoning performs best w.r.t. the number of solved instances. In contrast, for all settings with α between 0.6 and 1.1, the number of “better” instances is greater than the number of “worse” instances.

Evaluation of all optimal solved instances. Since many instances are either trivial or could not be solved, Table 2 presents the results only for those instances that

- (i) could be solved to optimality by all solvers,
- (ii) at least one solver needed more than one search node, and
- (iii) at least one solver needed more than one second of computational running time.

That means, we exclude all extremely easy instances and those which at least one of the solver was not able to solve. There are 112 and 32 instances remaining for the test sets J30 and J60, respectively.

Columns “better” and “worse” (which have the same meaning as in Table 1) reveal that values $\alpha \in \{0.9, 1.0\}$ are dominating all other settings

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.

α	J30				J60			
	better	worse	shnodes	shtime [sec]	better	worse	shnodes	shtime [sec]
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

w.r.t. the running time, independently of the test set. This behavior can be seen in more detail in the columns “shnodes” and “shtime” which state the shifted geometric mean³ of all nodes and of the running time, respectively. These columns show that 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 , the more nodes are needed. Besides that, the weak propagation factor $\alpha = 1.2$ performs worst for all criteria. 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. Finally, the development of the running times in shifted geometric mean are illustrated in Figure 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.

³ The shifted geometric mean of values t_1, \dots, t_n is defined as $(\prod(t_i + s))^{1/n} - s$ 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.

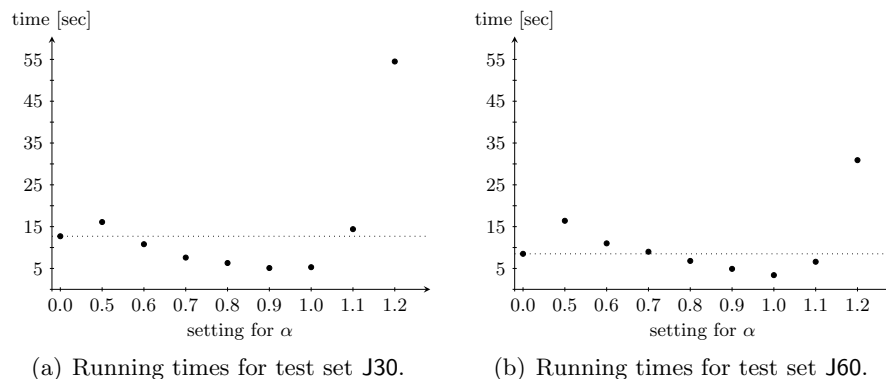


Fig. 2. Comparison of the running times shown in Table 2.

References

1. Blazewicz, J., Lenstra, J.K., Kan, A.H.G.R.: Scheduling subject to resource constraints: classification and complexity. *Discrete Applied Mathematics* **5**(1) (1983) 11–24
2. Baptiste, P., Le Pape, C., Nuijten, W.: Constraint-based scheduling: applying constraint programming to scheduling problems. *International Series in Operations Research & Management Science*, 39. Kluwer Academic Publishers, Boston, MA (2001)
3. Dorndorf, U., Phan-Huy, T., Pesch, E.: 10. In: A survey of interval capacity consistency tests for time- and resource-constrained scheduling. J. Weglarz, Kluwer Academic, Boston, MA (1999) 213–238
4. Klein, R., Scholl, A.: Computing lower bounds by destructive improvement: An application to resource-constrained project scheduling. *European Journal of Operational Research* **112**(2) (1999) 322–346
5. Kooli, A., Haouari, M., Hidri, L., Néron, E.: IP-based energetic reasoning for the resource constrained project scheduling problem. *Electronic Notes in Discrete Mathematics* **36** (2010) 359–366 ISCO 2010 - International Symposium on Combinatorial Optimization.
6. Hidri, L., Gharbi, A., Haouari, M.: Energetic reasoning revisited: application to parallel machine scheduling. *Journal of Scheduling* **11** (2008) 239–252
7. PSPLib: Project Scheduling Problem LIBrary. <http://129.187.106.231/psplib/> Last accessed 2010/Feb/01.
8. Berthold, T., Heinz, S., Lübbecke, M.E., Möhring, R.H., Schulz, J.: A constraint integer programming approach for resource-constrained project scheduling. In Lodi, A., Milano, M., Toth, P., eds.: *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. Volume 6140 of *Lecture Notes in Computer Science*. (2010) 313–317