A New Characterization of Relevant Intervals for Energetic Reasoning

Alban Derrien and Thierry Petit

TASC (Mines Nantes, LINA, CNRS, INRIA),
4, Rue Alfred Kastler, FR-44307 Nantes Cedex 3, France.
{alban.derrien, thierry.petit}@mines-nantes.fr

Abstract. Energetic Reasoning (ER) is a powerful filtering algorithm for the Cumulative constraint. Unfortunately, ER is generally too costly to be used in practice. One reason of its bad behavior is that many intervals are considered as relevant, although most of them should be ignored. In the literature, heuristic approaches have been developed in order to reduce the number of intervals to consider, leading to a loss of filtering. In this paper, we provide a sharp characterization that allows to reduce the number of intervals by a factor seven without loss of filtering.

1 Introduction

Due to its relevance in many industrial contexts, the NP-Hard Cumulative Scheduling Problem (CuSP) has been widely studied in Constraint Programming (CP). This problem is defined on a set of activities $\mathcal{A}$ consuming a resource of capacity C. Each activity $a \in \mathcal{A}$ is defined by four variables: its starting time s_a, its processing time p_a, its ending time e_a and its height h_a, which represents the amount of resource consumed by the activity when it is processed. We use the notation $a = \{s_a, p_a, e_a, h_a\}$. Usually, variables p_a and h_a are fixed integers, as well as C. In this paper, we make such assumptions. A solution to a CuSP is a schedule that satisfies the following constraints:

$$\forall a \in \mathcal{A} : s_a + p_a = e_a \quad \wedge \quad \forall t \in \mathbb{N} : \sum_{t \in [s_a, e_a[, a \in \mathcal{A}} h_a \leq C$$

In CP, this problem is generally represented by the global constraint *Cumulative* [1]. The Energetic Reasoning of Baptiste et al. (ER) is one of the most powerful filtering algorithms for *Cumulative* [2]. This algorithm uses a characterization of relevant intervals, that is, intervals that are sufficient to check in order to ensure that all the undergoing rules used for filtering domains are satisfied. Unfortunately, ER is often too costly to be used in practice. First, its time complexity is $O(n^3)$. Moreover, the hidden constant in that time complexity is huge, as many intervals are characterized to be relevant although most of them should be ignored. In the literature, only heuristic approaches have been proposed for reducing the number of checked intervals [3].

This article provides a sharper characterization of relevant intervals. We reduce the number of intervals by a factor seven without loss of reasoning. From this theoretical work, we improve the ER checker and we introduce a new ER propagator. Compared with state-of-the-art ER techniques for *Cumulative*, our experiments show a significant reduction in the running time of both the ER checker and the ER propagator.

2 Background

Given a variable x, let $\underline{x}$ be the minimum value in its domain and $\overline{x}$ the maximum value. The principle of ER is to compare the available energy within a given time interval (length of that interval $\times$ capacity) with the energy necessarily taken by activities that should partially or totally overlap this interval. The minimum energy for an activity can be found either when the activity is left shifted or right shifted.

We define the part of a left shifted activity a in intersection with an interval $[t_1, t_2[$ as $LS(a, t_1, t_2) = \max(0, \min(\underline{e_a}, t_2) - \max(\underline{s_a}, t_1))$. Similarly, for the right shifted intersection we define $RS(a, t_1, t_2) = \max(0, \min(\overline{e_a}, t_2) - \max(\overline{s_a}, t_1))$. Then the *minimal intersection* of activity a with an interval $[t_1, t_2[$ is:

$$MI(a, t_1, t_2) = \min(LS(a, t_1, t_2), RS(a, t_1, t_2))$$

Proposition 1 (ER checker [5]). *If the condition*

$$\forall t_1, t_2 \in \mathbb{N}^2, t_1 < t_2 \qquad C \times (t_2 - t_1) \geq \sum_{i \in \mathcal{A}} h_i \times MI(i, t_1, t_2) \tag{1}$$

is violated then the problem represented by Cumulative *is unfeasible.*

One issue is then to find the smallest sufficient set of intervals $[t_1, t_2[$ that should be checked to detect the unfeasibility.

Proposition 2 (Baptiste et al. characterization). *In order to ensure that the condition of Proposition 1 holds, it is sufficient to consider all pairs of activities (i, j) and check intervals $[t_1, t_2[$ from the set $O_B = \bigcup_{(i,j) \in \mathcal{A}^2} O_B(i, j)$, with:*

$$O_B(i, j) = \begin{cases} (t_1, t_2), t_1 \in O_1(i) < t_2 \in O_2(j) \\ (t_1, t_2), t_1 \in O_1(i) < t_2 \in O_{t_1}(j) \\ (t_1, t_2), t_2 \in O_2(j) > t_1 \in O_{t_2}(i) \end{cases}$$

and $O_1(i) = \{\underline{s_i}, \overline{s_i}, \underline{e_i}\}$, $O_2(i) = \{\overline{s_i}, \underline{e_i}, \overline{e_i}\}$, $O_t(i) = \{\underline{s_i} + \overline{e_i} - t\}$.

Proposition 1 can also be used to adjust bounds of starting and ending time variables. We examine if scheduling an activity a at its minimum schedule does not lead to a failure of condition (1). We first define the available energy for a over interval $[t_1, t_2[$ as the capacity of the interval minus the minimum intersection of all other activities:

$$\text{Avail}(a, t_1, t_2) = C \times (t_2 - t_1) - \sum_{i \in \mathcal{A} \setminus \{a\}} h_i \times MI(i, t_1, t_2)$$

Proposition 3. *For any activity a if there exists an interval $[t_1, t_2[$ such that $Avail(a, t_1, t_2) < h_a \times LS(a, t_1, t_2)$ then the left shift placement of a is not valid and the activity can not start before $t_2 - \frac{1}{h_a} \times Avail(a, t_1, t_2)$.*

Proposition 4. *For any activity a there exists an interval $[t_1, t_2[$ such that $Avail(a, t_1, t_2) < h_a \times RS(a, t_1, t_2)$ then the right shift placement of activity a is not valid and a can not end after $t_1 + \frac{1}{h_a} \times Avail(a, t_1, t_2)$.*

Definition 1 (Complete ER propagation). *The* Complete ER Propagation *is obtained when no activity can be adjusted using Proposition 3 or 4.*

The characterization of Proposition 2 is proved to be sufficient in [2] (Proposition 19) for the ER checker. Two open questions remain. The first one is related to the checker: The set of relevant intervals O_B is proved to be sufficient but could it be reduced? The second one is related to the propagator: Is O_B also sufficient to perform a complete ER propagation? In the next section, we demonstrate that one can respond affirmatively to those two questions.

3 The Energetic Reasoning checker revisited

Baptiste et al. showned that $f_1 : (t_1, t_2) \to C \times (t_2 - t_1) - \sum_{i \in \mathcal{A}} h_i \times MI(i, t_1, t_2)$ is continuous and piecewise linear, and that any piece can be bounded by points defined in their characterization. As extrema of a continuous and piecewise linear function can only be found on bounds of the pieces their characterization is sufficient. Out of the scope of Constraint Programming, Schwindt proposed in [9] a study of f_1 limited to local minima in order to compute a lower bound of the makespan. We propose a study adapted to the computation of relevant intervals for the Energetic Reasoning checker.

Lemma 1. *f_1 is locally minimum in (t_1, t_2) only if there exist two activities i and j such that the two following conditions are satisfied.*

$$\frac{\partial^- MI(i, t_1, t_2)}{\partial t_1} > \frac{\partial^+ MI(i, t_1, t_2)}{\partial t_1} \tag{2}$$

$$\frac{\partial^- MI(j, t_1, t_2)}{\partial t_2} > \frac{\partial^+ MI(j, t_1, t_2)}{\partial t_2} \tag{3}$$

Proof. By contradiction, let (t_1, t_2) such that for all activities in $\mathcal{A}$ condition (2) is not satisfied. Then $\sum_{i \in \mathcal{A}} h_i \times MI(i, t_1, t_2)$ has its left derivative lower than or equals to it's right derivative and f_1 has its left derivative greater than or equal to its right. By the second derivative test, minimal value of a function can only be found at points where its left derivative is lower than its right derivative. (t_1, t_2) can not be a local minimum. Proof is similar for condition (3). This proves the lemma. □

The set of intervals O_B characterizes for any couple of activity (i, j) a total number of 15 intervals. This number can be reduced thanks to Lemma 1: We can deduce necessary conditions for determining the subset of intervals that are really relevant. We first characterize the condition for which the end of an interval may be relevant.

Lemma 2. *For any activity j and any interval starting time t_1 there exists at most one interval $[t_1, t_2[$ such that $\frac{\partial^- MI(j,t_1,t_2)}{\partial t_2} > \frac{\partial^+ MI(j,t_1,t_2)}{\partial t_2}$:*

1. *if* $t_1 \leq \underline{s_j}$ *then only* $[t_1, \overline{e_j}[$ *has to be considered*
2. *if* $t_1 > \underline{s_j} \wedge t_1 \geq \underline{e_j}$ *then no interval has to be considered*
3. *if* $t_1 > \underline{s_j} \wedge t_1 < \underline{e_j} \wedge t_1 < \overline{s_j}$ *then only* $[t_1, \underline{s_j} + \overline{e_j} - t_1[$ *has to be considered*
4. *if* $t_1 > \underline{s_j} \wedge t_1 < \underline{e_j} \wedge t_1 \geq \overline{s_j}$ *then only* $[t_1, \underline{e_j}[$ *has to be considered*

Proof. Let us study the variation of the function $f_2^j : t_2 \rightarrow MI(j, t_1, t_2)$ when t_2 varies. As an example that illustrates the case of the first item, Figure 1 is a representation of the evolution of the minimal intersection of an activity with the following data: $j = \{s_j \in [2,4],\ p_j = 4,\ e_j \in [6,8], h_j\}$. We can distinguish three cases.

- If $t_2 \le \overline{s_j}$ then
  $MI(j, t_1, t_2) = 0$.
- If $\overline{s_j} \le t_2 \le \overline{e_j}$ then
  $MI(j, t_1, t_2) = t_2 - \overline{s_j}$.
- And finally if $\overline{e_j} \le t_2$ then
  $MI(j, t_1, t_2) = p_j$.

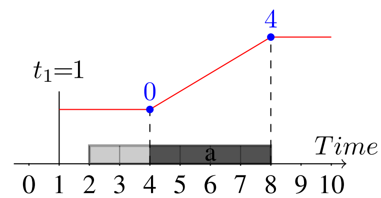

Fig. 1: a graphical exemple

The only interval for which $\frac{\partial^- MI(j,t_1,t_2)}{\partial t_2} > \frac{\partial^+ MI(j,t_1,t_2)}{\partial t_2}$ is then $[t_1, \overline{e_j}[$; $[1, 8[$ in the example. Similar case-based proofs apply for other items [4]. □

Lemma 3. *f_1 is locally minimum in (t_1, t_2) only if there exist two activities i and j such that $(t_1, t_2) \in O_C(i, j)$ with*

$$O_C(i,j) = \begin{cases} [\underline{s_i},\ \overline{e_j}[& \text{if } \underline{s_i} \le \underline{s_j} \wedge \overline{e_j} \ge \overline{e_i} \\ [\underline{s_i},\ \underline{s_j}+\overline{e_j}-\underline{s_i}[& \text{if } \underline{s_i} > \underline{s_j} \wedge \underline{s_i} < \underline{e_j} \wedge \underline{s_i} < \overline{s_j} \wedge \underline{s_j}+\overline{e_j}-\underline{s_i} \ge \overline{e_i} \\ [\underline{s_i},\ \underline{e_j}] & \text{if } \underline{s_i} > \underline{s_j} \wedge \underline{s_i} < \underline{e_j} \wedge \underline{s_i} \ge \overline{s_j} \wedge \underline{e_j} \ge \overline{e_i} \\ [\overline{s_i}, \overline{e_j}] & \text{if } \overline{s_i} \le \underline{s_j} \wedge \overline{e_j} < \overline{e_i} \wedge \overline{e_j} > \overline{s_i} \wedge \overline{e_j} \le \underline{e_j} \\ [\overline{s_i},\ \underline{s_j}+\overline{e_j}-\overline{s_i}] & \text{if } \overline{s_i} > \underline{s_j} \wedge \overline{s_i} < \underline{e_j} \wedge \overline{s_i} < \overline{s_j} \wedge \\ & \quad \overline{s_i} < \underline{s_j}+\overline{e_j}-\overline{s_i} \le \underline{e_i} \wedge \underline{s_j}+\overline{e_j}-\overline{s_i} < \overline{e_i} \\ [\overline{s_i},\ \underline{e_j}] & \text{if } \overline{s_i} > \underline{s_j} \wedge \overline{s_i} < \underline{e_j} \wedge \overline{s_i} \ge \overline{s_j} \wedge \\ & \quad \underline{e_j} < \overline{e_i} \wedge \underline{e_j} > \overline{s_i} \wedge \underline{e_j} \le \underline{e_i} \\ [\underline{s_i}+\overline{e_i}-\overline{e_j},\ \overline{e_j}] & \text{if } \overline{e_j} < \overline{e_i} \wedge \overline{e_j} > \overline{s_i} \wedge \overline{e_j} > \underline{e_i} \wedge \underline{s_i}+\overline{e_i}-\overline{e_j} \le \overline{s_j} \\ [\underline{s_i}+\overline{e_i}-\underline{e_j},\ \underline{e_j}] & \text{if } \underline{e_j} < \overline{e_i} \wedge \underline{e_j} > \overline{s_i} \wedge \underline{e_j} > \underline{e_i} \wedge \\ & \quad \overline{s_j} \le \underline{s_i}+\overline{e_i}-\underline{e_j} < \underline{e_j} \wedge \underline{s_j} < \underline{s_i}+\overline{e_i}-\underline{e_j} \end{cases}$$

Proof. Suppose $\nexists(i,j)$ such that $(t_1,t_2) \in O_C(i,j)$ then by Lemma 2 and its symmetric both condition $\frac{\partial^- MI(j,t_1,t_2)}{\partial t_2} > \frac{\partial^+ MI(j,t_1,t_2)}{\partial t_2}$ and $\frac{\partial^- MI(j,t_1,t_2)}{\partial t_1} > \frac{\partial^+ MI(j,t_1,t_2)}{\partial t_1}$ can not be satisfied; by Lemma 1 f_1 can not be minimal. This proves the Lemma. □

Theorem 1. *In order to ensure ER checker property holds (condition (1)), it is enough to check intervals of the form $O_C(\mathcal{A}) = \bigcup_{(i,j)\in\mathcal{A}^2} O_C(i,j)$.*

Proof. Suppose $\exists[t_1,t_2[$ such that $\sum_{i\in\mathcal{A}} h_i \times MI(i,t_1,t_2) - C \times (t_2 - t_1) < 0$. By Lemma 3, $\exists[t_1^*, t_2^*[\in O_C(\mathcal{A})$ such that $\sum_{i\in\mathcal{A}} h_i \times MI(i,t_1^*,t_2^*) - C \times (t_2^* - t_1^*) \le \sum_{i\in\mathcal{A}} h_i \times MI(i,t_1,t_2) - C \times (t_2 - t_1)$. f_1 is negative in (t_1^*, t_2^*), thus checking $[t_1^*, t_2^*[$ leads to a failure. The characterization is sufficient. □

This precise characterization reduces the number of relevant intervals for any pair of activities. Our characterization leads to 2 intervals for any pair of activities, as no more than two conditions can be simultaneously valid. We have thus reduced the number of intervals by a factor 7 compared with Baptiste et al. characterization. Moreover, no intervals start by $\underline{e_i}$ or end by $\overline{s_j}$.

4 Characterization of intervals for the propagator

Similarly to the checker, we aim to find minimal values of the induced function $f_3^a : (t_1, t_2) \rightarrow \text{Avail}(a, t_1, t_2) - h_a \times \mathit{LS}(a, t_1, t_2)$. If f_3^a takes a negative value, the lower bound of activity a can be adjusted (thanks to Proposition 3).

Lemma 4. *f_3^a is locally minimum in (t_1, t_2) only if one of the four conditions is satisfied:*

$$\exists (i,j), \frac{\partial^- \mathit{MI}(i,t_1,t_2)}{\partial t_1} > \frac{\partial^+ \mathit{MI}(i,t_1,t_2)}{\partial t_1} \wedge \frac{\partial^- \mathit{MI}(j,t_1,t_2)}{\partial t_2} > \frac{\partial^+ \mathit{MI}(j,t_1,t_2)}{\partial t_2} \quad (4)$$

$$\exists i, \frac{\partial^- \mathit{MI}(i,t_1,t_2)}{\partial t_1} > \frac{\partial^+ \mathit{MI}(i,t_1,t_2)}{\partial t_1} \wedge \frac{\partial^- \mathit{LS}(a,t_1,t_2)}{\partial t_2} > \frac{\partial^+ \mathit{LS}(a,t_1,t_2)}{\partial t_2} \quad (5)$$

$$\exists j, \frac{\partial^- \mathit{LS}(a,t_1,t_2)}{\partial t_1} > \frac{\partial^+ \mathit{LS}(a,t_1,t_2)}{\partial t_1} \wedge \frac{\partial^- \mathit{MI}(j,t_1,t_2)}{\partial t_2} > \frac{\partial^+ \mathit{MI}(j,t_1,t_2)}{\partial t_2} \quad (6)$$

$$\frac{\partial^- \mathit{LS}(a,t_1,t_2)}{\partial t_1} > \frac{\partial^+ \mathit{LS}(a,t_1,t_2)}{\partial t_1} \wedge \frac{\partial^- \mathit{LS}(a,t_1,t_2)}{\partial t_2} > \frac{\partial^+ \mathit{LS}(a,t_1,t_2)}{\partial t_2} \quad (7)$$

Proof. Similar to proof of Lemma 1. □

We can build from Lemma 4 the set of relevant intervals for a couple of activities from the four conditions. Intervals satisfying condition (4) have already been defined: $O_C(\mathcal{A} \backslash a)$. From conditions (5), (6) and (7) we can similarly build the set L^a studying the conditions from the left shift placement function $f_4^a : (t_1, t_2) \rightarrow \mathit{LS}(a, t_1, t_2)$.

Lemma 5. *For any activity a and any interval starting time t_1 there exists at most one interval $[t_1, t_2[$ such that $\frac{\partial^- \mathit{LS}(a,t_1,t_2)}{\partial t_2} > \frac{\partial^+ \mathit{LS}(a,t_1,t_2)}{\partial t_2}$:*

- *If $t_1 < \underline{e_a}$ then only $[t_1, \underline{e_a}[$ has to be considered.*
- *If $t_1 \geq \underline{e_a}$ then no intervals have to be considered.*

Proof. We consider 3 different cases :

1. $t_1 < \underline{s_a}$:
   Then $\mathit{LS}(a, t_1, t_2) = \max(0, \min(\underline{e_a}, t_2) - \underline{s_a})$
   (a) if $t_2 \leq \underline{s_a}$ then $\mathit{LS}(a, t_1, t_2) = 0$.
   (b) if $\underline{s_a} \leq t_2 \leq \underline{e_a}$ then $\mathit{LS}(a, t_1, t_2) = t_2 - \underline{s_a}$.
   (c) if $\underline{e_a} \leq t_2$ then $\mathit{LS}(a, t_1, t_2) = p_a$.
   The only interval for which $\frac{\partial^- \mathit{LS}(a,t_1,t_2)}{\partial t_2} > \frac{\partial^+ \mathit{LS}(a,t_1,t_2)}{\partial t_2}$ is then $[t_1, \underline{e_a}[$.
2. $\underline{s_a} \leq t_1 < \underline{e_a}$:
   Then $\mathit{LS}(a, t_1, t_2) = \max(0, \min(\underline{e_a}, t_2) - t_1)$
   (a) if $t_2 \leq \underline{e_a}$ then $\mathit{LS}(a, t_1, t_2) = t_2 - t_1$.
   (b) if $\underline{e_a} \leq t_2$ then $\mathit{LS}(a, t_1, t_2) = \underline{e_a} - t_1$.
   The only interval for which $\frac{\partial^- \mathit{LS}(a,t_1,t_2)}{\partial t_2} > \frac{\partial^+ \mathit{LS}(a,t_1,t_2)}{\partial t_2}$ is then $[t_1, \underline{e_a}[$.
3. $\underline{e_a} \leq t_1$: Then $\mathit{LS}(a, t_1, t_2) = 0$ and no interval satisfies the condition.

Combination of cases 1, 2 and 3 proves the lemma. □

We now precisely characterize relevant intervals for the left shift placement of activity a, from the conditions 5 , 6 and 7 : $L^a = \bigcup_{i\in\mathcal{A}\setminus a} L_1^a(i) \bigcup_{j\in\mathcal{A}\setminus a} L_2^a(j) \bigcup L_3^a$. From Lemma 5 and the symmetric of Lemma 2, we can characterize for any i the interval that satisfy condition (5).

$$L_1^a(i) = \begin{cases} [\underline{s_i},\ \underline{e_a}[& \text{if} \quad \underline{s_i} < \overline{s_a} \ \wedge\ \overline{e_i} < \underline{e_a} \\ [\underline{s_i} + \overline{e_i} - \underline{e_a},\ \underline{e_a}[& \text{if} \quad \underline{s_i} + \overline{e_i} - \underline{e_a} < \underline{e_a} \ \wedge\ \underline{s_i} + \overline{e_i} - \underline{e_a} < \overline{s_a} \ \wedge \\ & \qquad \underline{e_a} < \overline{e_i} \ \wedge\ \underline{e_a} > \overline{s_i} \ \wedge\ \underline{e_a} > \underline{e_i} \\ [\overline{s_i},\ \underline{e_a}[& \text{if} \quad \overline{s_i} < \overline{s_a} \ \wedge\ \underline{e_a} < \overline{e_i} \ \wedge\ \underline{e_a} < \overline{s_i} \ \wedge\ \underline{e_a} \leq \underline{e_i} \end{cases}$$

From the symmetric of Lemma 5 and Lemma 2 we can characterize for any j the interval that satisfy condition (6).

$$L_2^a(j) = \begin{cases} [\underline{s_a},\ \overline{e_j}[& \text{if} \quad \underline{s_a} \leq \underline{s_j} \ \wedge\ \overline{e_j} < \overline{e_a} \\ [\underline{s_a},\ \underline{s_j} + \overline{e_j} - \underline{s_a}[& \text{if} \quad \underline{s_a} > \underline{s_j} \ \wedge\ \underline{s_a} < \underline{e_j} \ \wedge\ \underline{s_a} < \overline{s_j} \ \wedge \\ & \qquad \underline{s_j} + \overline{e_j} - \underline{e_a} > \underline{s_a} \ \wedge\ \underline{s_j} + \overline{e_j} - \underline{e_a} < \overline{e_a} \\ [\underline{s_a},\ \underline{e_j}[& \text{if} \quad \underline{s_a} > \underline{s_j} \ \wedge\ \underline{s_a} < \underline{e_j} \ \wedge\ \underline{s_a} \geq \overline{s_j} \ \wedge\ \overline{e_j} < \overline{e_a} \end{cases}$$

From Lemma 5 and its symmetric we can build the interval that satisfy condition (7).

$$L_3^a = \{\ [\underline{s_a},\ \underline{e_a}[\}$$

Lemma 6. *f_3^a is locally minimum only in $(t_1, t_2) \in O_L^a$ with $O_L^a = O_C(\mathcal{A}\setminus a) \cup L^a$.*

Proof. Same proof as Lemma 3. □

The same reasoning leads to the characterization of relevant intervals for the right shift placement $R^a = \bigcup_{j\in\mathcal{A}\setminus a} R_1^a(j) \bigcup_{i\in\mathcal{A}\setminus a} R_2^a(i) \bigcup R_3^a$.The precise characterization is symmetrical to the left shift placement characterization.

The number of relevant intervals for any activity a is then $|O_C(\mathcal{A}\setminus a) \cup L^a \cup R^a|$. By construction, $|O_C(\mathcal{A}\setminus a)| = 2(n-1)^2$ and $|L^a| = |R^a| = 2.n + 1$. Compared with Baptiste et al. characterization, our characterization reduces by a factor 7 the number of relevant intervals.

Theorem 2. *In order to ensure a complete ER propagation (Definition 1) it is sufficient to check intervals $[t_1, t_2[$ in $O_P = O_C(\mathcal{A}) \bigcup_{a\in\mathcal{A}} L^a \bigcup_{a\in\mathcal{A}} R^a$.*

Proof. Same proof as Theorem 1. □

We can thus respond affirmatively to the second open question:

Property 1. Baptiste et al. characterization of relevant intervals O_B is sufficient to ensure a complete ER propagation.

Proof. By Theorem 2, O_P is sufficient and $O_P \subset O_B$. □

5 Algorithms and Experiments

5.1 Checker

Baptiste et al. proposed an $O(n^2)$ checker algorithm based on their characterization. Their algorithm loops over set $O_1 = \bigcup_{a \in \mathcal{A}} \{\underline{s_a}, \overline{s_a}, \underline{e_a}\}$ to compute all relevant intervals starting by a value in O_1. We have shown that $\underline{e_a}$ is not relevant as a starting value. We propose a version of the algorithm adapted to our characterization, reducing the relevant starting values. We replace O_1 by $O'_1 = \bigcup_{a \in \mathcal{A}} \{\underline{s_a}, \overline{s_a}\}$ and apply the same algorithm.

5.2 Propagator

The same adaptation could be made to Baptiste et al's propagator using the reduced set O'_B, removing $\underline{e_a}$ from $O_1(a)$ and $\overline{s_a}$ from $O_2(a)$. This adaptation is simple but it deals with a superset of the relevant intervals obtained with our sharp characterization. Therefore, we propose a new ER algorithm. As the characterization given in Theorem 2, the algorithm is in 3 parts. First, we apply Baptiste et al's algorithm reduced to the set of relevant intervals $O_C(\mathcal{A})$ (lines 1 to 9). Then, for all activities we check its left and right shifted placements with sets L^a (lines 11 to 15) and R^a (lines 16 to 20).

Algorithm 1: ERpropagator()

---

1 **foreach** $(t_1, t_2) \in O_C(\mathcal{A})$ **do**
2     $W := \sum_{a \in \mathcal{A}} h_a \times MI(a, t_1, t_2)$;
3   **if** $W > C \times (t_2 - t_1)$ **then** fail;
4   **else foreach** $a \in \mathcal{A}$ **do**
5         $avail := C \times (t_2 - t_1) - W + h_a \times MI(a, t_1, t_2)$;
6   **if** $avail < h_a.LS(a, t_1, t_2)$ **then**
7             $\underline{s_a} := \max(\underline{s_a},\ t_2 - \frac{1}{h_a} \times avail)$;
8   **if** $avail < h_a.RS(a, t_1, t_2)$ **then**
9             $\overline{e_a} := \min(\overline{e_a},\ t_1 + \frac{1}{h_a} \times avail)$;
10 **foreach** $a \in \mathcal{A}$ **do**
11   **foreach** $(t_1, t_2) \in L^a$ **do**
12         $avail := C \times (t_2 - t_1) - \sum_{i \in \mathcal{A} \setminus a} h_a \times MI(i, t_1, t_2)$;
13   **if** $avail < h_a.MI(a, t_1, t_2)$ **then** fail;
14   **else if** $avail < h_a.LS(a, t_1, t_2)$ **then**
15             $\underline{s_a} := \max(\underline{s_a},\ t_2 - \frac{1}{h_a} \times avail)$;
16   **foreach** $(t_1, t_2) \in R^a$ **do**
17         $avail := C \times (t_2 - t_1) - \sum_{i \in \mathcal{A} \setminus a} h_a \times MI(i, t_1, t_2)$;
18   **if** $avail < h_a.MI(a, t_1, t_2)$ **then** fail;
19   **else if** $avail < h_a.RS(a, t_1, t_2)$ **then**
20             $\overline{e_a} := \min(\overline{e_a},\ t_1 + \frac{1}{h_a} \times avail)$;

5.3 Experiments

Experiments were run on a 2.9 GHz Intel Core i7, in Choco [10] version 3 (release 13.03). In order to check the gain obtained with the new characterization we have considered 100 random instances and the instances from the PSPLIB [7]. Random instances have either 10 or 20 activities. Their processing times were chosen within $[1, 10]$, their heights within $[1, 5]$. We used the *first fail* [6] search strategy (the current default strategy of Choco) and compared our algorithms with the corresponding state of the art algorithms [2], both combined with the Time-Table (TT) filtering algorithm of Letort et al. [8]. The number of nodes is identical for all proved instances, as expected. Table 1 shows a running time improvement of 20 to 36% using the new checker (measured in $\mu s/node$). Table 2 shows a time improvement of 49 to 72% using the new propagator.

Instances	New checker ($\mu s/node$)	Baptiste et al ($\mu s/node$)	Gain in %
Random10	16	25	36
Random20	44	56	21
PspLib 30	451	619	27
PspLib 120	1 339	1 683	20

Table 1: Comparison of average running of ER checkers.

Instances	Algorithm 1 ($\mu s/node$)	Baptiste et al ($\mu s/node$)	Gain in %
Random10	91	244	62
Random20	327	641	49
PspLib 30	4 372	8 809	50
PspLib 120	41 418	151 390	72

Table 2: Comparison of average running of ER propagators.

We also compared those combinations with the state-of-the-art filtering combination: TT + Time-Table Edge-Finding (TTEF) [11]. We tried to prove optimality. On the random10 instances, TT associated with our new ER propagator proved 63 out of 100 instances in the given time limit of five minutes. TT+TTEF was only able to prove 8 instances, mainly due to the fact that TTEF does not include an energetic checker whereas our ER propagator does; The combination TT+TTEF+ our ER Checker proved 72 instances. This shows the interest of an energetic checker as a standard feature of *Cumulative* in existing solvers. Regarding the ER propagator, a promising perspective of our work is to exploit the theoretical characterization to design a light version, with a lower time complexity than the current propagator but still filtering more values than TTEF.

6 Discussion and conclusion

We have proposed a new characterization of relevant intervals for the energetic reasoning. Our characterization reduces by a factor seven the number of relevant intervals for the checker and for filtering any activity. We answered to an open question: Baptiste et al. characterization is sufficient to ensure a complete bounds adjustment. Compared with state-of-the-art ER techniques for Cumulative, our experiments show a significant reduction in the running time of both the ER checker and the ER propagator. Our sharpened characterization opens the new possibility to analyze the impact, in terms of filtering, of each type of relevant interval. This may help to design heuristics for ignoring some intervals without decreasing too much the pruning power of ER.

References

1. Abder Aggoun and Nicolas Beldiceanu. Extending chip in order to solve complex scheduling and placement problems. *Math. Comput. Model.*, 17(7):57–73, April 1993.
2. Philippe Baptiste, Claude Le Pape, and Wim Nuijten. *Constraint-Based Scheduling: Applying Constraint Programming to Scheduling Problems*. International Series in Operations Research and Management Science. Kluwer, 2001.
3. Timo Berthold, Stefan Heinz, and Jens Schulz. An approximative criterion for the potential of energetic reasoning. In *Proceedings of the First international ICST conference on Theory and practice of algorithms in (computer) systems*, TAPAS'11, pages 229–239, Berlin, Heidelberg, 2011. Springer-Verlag.
4. Alban Derrien and Thierry Petit. The Energetic Reasoning Checker Revisited. In *CP Doctoral Program 2013*, pages 55–60, Uppsala, Sweden, September 2013.
5. Jacques Erschler, Pierre Lopez, and Catherine Thuriot. Scheduling under time and resource constraints. In *Proc. of Workshop on Manufacturing Scheduling, 11th IJCAI*, Detroit, USA, 1989.
6. Robert M. Haralick and Gordon L. Elliot. Increasing tree search efficiency for constraint satisfaction problems. *Artificial Intelligence*, 14(3):263–313, 1980.
7. Rainer Kolisch and Arno Sprecher. Psplib – a project scheduling problem library. *European Journal of Operational Research*, 96:205–216, 1996.
8. Arnaud Letort, Nicolas Beldiceanu, and Mats Carlsson. A scalable sweep algorithm for the cumulative constraint. In Michela Milano, editor, *Principles and Practice of Constraint Programming*, Lecture Notes in Computer Science, pages 439–454. Springer Berlin Heidelberg, 2012.
9. Christoph Schwindt. *Verfahren zur Lösung des ressourcenbeschränkten Projektdauerminimierungsproblems mit planungsabhängigen Zeitfenstern*. PhD thesis, Fakultät für wirtschaftswissenschaften der Universität Fridericiana zu Karlsruhe, 1998. in German.
10. CHOCO Team. Choco: an open source Java CP library. Research report 10-02-INFO, Ecole des Mines de Nantes, 2010.
11. Petr Vilím. Timetable edge finding filtering algorithm for discrete cumulative resources. In Tobias Achterberg and J. Beck, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, volume 6697 of *Lecture Notes in Computer Science*, pages 230–245. Springer Berlin / Heidelberg, 2011.