

The Energetic Reasoning Checker Revisited

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 by the checker of ER, although most of them should be ignored. In this paper, we provide a sharp characterization that allows to reduce the number of intervals by a factor seven. Our experiments show that associating this checker with a Time-Table filtering algorithm leads to promising results.

1 Introduction

Due to its relevance in many industrial contexts, the Cumulative Scheduling Problem (CuSP) has been widely studied in Constraint Programming (CP). The CuSP is NP-Hard. It is defined by a set of activities \mathcal{A} and 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 resources 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. In this paper, we make such an assumption. A solution to a CuSP is a schedule that satisfies the following constraints:

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

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 algorithm for Cumulative [2]. This algorithm uses a characterization of intervals of interest, 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 of interest although most of them should be ignored. In the literature, only heuristics approaches have been proposed for reducing the number of checked intervals [3].

This article provides a sharper characterization of intervals of interest. Our experiments show a significant reduction in the running time of the ER checker. We point out that associating this energetic checker with a Time-Table filtering algorithm (based on the cumulative profile of mandatory parts [7]) leads to promising results. Further, our approach should permit to answer affirmatively to a theoretical open question with respect to energetic filtering algorithms: Are the intervals of interest for the ER checker sufficient in order to perform a complete ER filtering for the activities variables?

2 Background

Given a variable x , we denote \underline{x} the minimum value in its domain and \overline{x} the maximum value. The principle of ER is to compare the available area within a given interval (length of that interval \times capacity) with the quantity of resource necessarily taken by activities that should partially or totally overlap this interval. The ER checker is defined using $MI^a(t_1, t_2)$, the *minimum intersection* of activity a with an interval $[t_1, t_2]$. $MI^a(t_1, t_2) = \max(0, \min(p_a, t_2 - t_1, \underline{e}_a - t_1, t_2 - \overline{s}_a))$.

Proposition 1 (ER checker [4]). *If the condition*

$$\forall t_1, t_2 \in \mathbb{N}^2, t_1 < t_2 \quad C \times (t_2 - t_1) \geq \sum_{a \in \mathcal{A}} h_a \times MI^a(t_1, t_2) \quad (1)$$

is violated then the problem represented by Cumulative is unfeasible.

The issue is then to find the smallest set of intervals $[t_1, t_2], t_2 > t_1$ that should be checked to detect the unfeasibility. From this condition a bound adjustment rule can be defined. Value \underline{s}_a can be updated if scheduling activity a on \underline{s}_a would cause an overload as described above. To check such potential overloads, we need to compute how much additional energy a would require during $[t_1, t_2]$ if a is scheduled on \underline{s}_a .

Baptiste et al. characterization In order to ensure that condition (1) holds, it is enough to check intervals of the form: $[t_1, t_2], t_1 \in O_1 < t_2 \in O_2, [t_1, t_2], t_1 \in O_1 < t_2 \in O(t_1), [t_1, t_2], t_2 \in O_2 > t_1 \in O(t_2)$, where $O_1 = \{\underline{s}_a, \forall a \in \mathcal{A}\} \cup \{\overline{s}_a, \forall a \in \mathcal{A}\} \cup \{\underline{e}_a, \forall a \in \mathcal{A}\}$, $O_2 = \{\overline{e}_a, \forall a \in \mathcal{A}\} \cup \{\overline{s}_a, \forall a \in \mathcal{A}\} \cup \{\underline{e}_a, \forall a \in \mathcal{A}\}$, $O(t) = \{\underline{s}_a + \overline{e}_a - t, \forall a \in \mathcal{A}\}$. The resulting subset of intervals forms the *intervals of interest* of the ER. There is for each pair of activities $9+3+3=15$ intervals of interest. The characterization is proved to be sufficient in [2] (proposition 19), by implicitly analyzing the slack function $Slack(t_1, t_2) = C \times (t_2 - t_1) - \sum_{a \in \mathcal{A}} h_a \times MI^a(t_1, t_2)$. As $MI^a(t_1, t_2)$ can be computed in constant time and given the definitions of O_1, O_2 and O_t , we obtain a naive checker in $O(n^3)$, by computing $MI^a(t_1, t_2)$ for all a, t_1, t_2 . However, the slack function is continuous piecewise linear, a local extrema can only be found at flexion points, i.e., when the slope is changing, which has been proved to be only in such intervals. This leads to a checker in $O(n^2)$ [2]. Two open questions remain.

1. The new set of intervals of interest is proved to be sufficient but could it be reduced?
2. Is the new set of intervals also sufficient to ensure a complete filtering of the time bound adjustment rule based on the checker?

Schwindt's characterization Schwindt has proposed a finer characterization of the minimal value of the slack function (Theorems 3.7 and 3.8 in [8], written in German): The slack function is locally minimal in interval $[t_1, t_2]$ only if its left derivative is negative and right derivative positive, for both t_1 and t_2 . As the slack function correspond to the negation of the sum of minimal intersections, it must exists an activity i (resp j) such that its minimal intersection has a left derivative greater than its right derivative on t_1 (resp. t_2). This leads to Theorem 1.

Theorem 1. *The slack function is locally minimum in interval $[t_1, t_2]$ only if it exists two activities i, j such that the two following conditions are satisfied.*

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

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

Schwindt analyzes the variation of minimum intersection. He provides a first characterization of the 8 possible intervals of interest for any pair of activities. This answers to the first open question: The number of intervals in Baptiste et al. characterization can be reduced. We propose in next section a different analyze of the minimum intersection, which leads to a sharper characterization of activity pairs.

3 A New Characterization

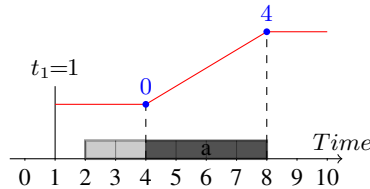
For symmetry reasons we focus on condition (3). We wish to characterize for a starting time t_1 and an activity a the positive inflection points of function $t_2 \rightarrow MI^a(t_1, t_2)$: Values t_2 at which the left derivative of is greater than the right derivative.

Theorem 2. *For a starting time t_1 and an activity a the function $t_2 \rightarrow MI^a(t_1, t_2)$ has at most one positive inflection point.*

Proof. We prove theorem 2 by studying the four positions of t_1 w.r.t. a . We prove that the function $t_2 \rightarrow MI^a(t_1, t_2)$ is continuous piecewise linear, composed of at most three parts. The two inflection points correspond to the start of consumption (SoC_a) and the end of consumption (EoC_a). We also prove that EoC_a is the only inflection point with a left derivative greater than the right derivative. Graphically we show an example of each cases with an activity $a = \{s_a \in [2, 4], p_a = 4, e_a \in [6, 8], h_a\}$.

Case 1.

On the first case, the minimum starting time of the activity a is greater than or equal to the profile starting time: $t_1 \leq \underline{s}_a$.



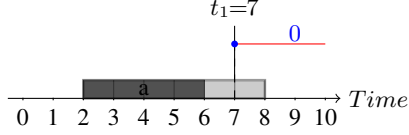
By definition we have $MI^a(t_1, t_2) = \max(0, \min(p_a, t_2 - t_1, \underline{e}_a - t_1, t_2 - \underline{s}_a))$. We can deduce three different situations:

1. if $t_2 \leq \underline{s}_a$ then $MI^a(t_1, t_2) = 0$.
2. or $\underline{s}_a \leq t_2 \leq \underline{e}_a$ then $MI^a(t_1, t_2) = t_2 - \underline{s}_a$.
3. and finally $\underline{e}_a \leq t_2$ then $MI^a(t_1, t_2) = p_a$.

$p_a - (\underline{e}_a - t_2)$ equals 0 when $t_2 = \underline{s}_a$ and p_a when $t_2 = \underline{e}_a$. This proves that when $t_1 \leq \underline{s}_a$ then consumption function is continue and piecewise linear, composed of three pieces with only one positive inflection point: EoC_a .

Case 2.

On the second case, the minimum ending time of the activity a is smaller than or equal to the profile starting time: $t_1 \geq \underline{e}_a$.

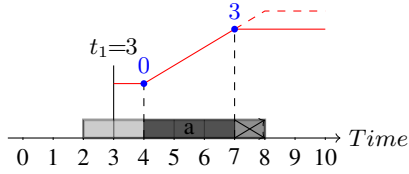


In this case, we have $MI^a(t_1, t_2) = 0$ for any interval. Then the function is trivially continue and piecewise linear with zero inflection point.

Case 3.

On the third case, the profile starting time is greater than the minimum start time, but smaller than the minimum end time and maximum start time:

$t_1 > \underline{s}_a$ and $t_1 < \underline{e}_a$ and $t_1 < \overline{s}_a$



Let $\Delta = t_1 - \underline{s}_a$. We distinct three cases for the value of t_2 :

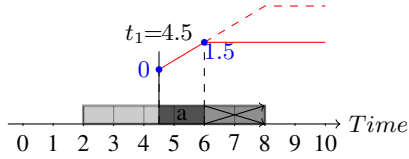
1. if $t_2 \leq \overline{s}_a$ then $MI^a(t_1, t_2) = 0$.
2. or $\overline{s}_a \leq t_2 \leq \overline{e}_a - \Delta$ then $MI^a(t_1, t_2) = t_2 - \overline{s}_a$.
3. and finally $t_2 \geq \overline{e}_a - \Delta$ then $MI^a(t_1, t_2) = p_a - \Delta$.

As $t_2 - \overline{s}_a$ equals 0 when $t_2 = \overline{s}_a$ and $p_a - \Delta$ when $t_2 = \overline{e}_a - \Delta$ we have proved that the consumption is continue and piecewise linear, composed of three pieces with only one positive inflection point: EoC_a .

Case 4.

On the fourth case, the profile starting time intersect the mandatory part a :

$t_1 > \underline{s}_a$ and $t_1 < \underline{e}_a$ and $t_1 \geq \overline{s}_a$



In this case we have two distinct cases for the value of t_2 :

1. if $t_2 \leq \underline{e}_a$ then $MI^a(t_1, t_2) = t_2 - t_1$
2. otherwise $MI^a(t_1, t_2) = p_a - \Delta$

As $t_2 - t_1 = p_a - \Delta$ when $t_2 = \underline{e}_a$ then the consumption function is continue and piecewise linear, composed of two pieces, with only one positive inflection point : EoC_a .

Conclusion We have shown that for any starting value t_1 , and any activity a the function $t_2 \rightarrow MI^a(t_1, t_2)$ is linear and piecewise continue, with, in each case, at most one point at which the left derivative is greater than the right derivate: EoC_a . This proves the Theorem. \square

Table 1: Intervals of interest for a pair of activities (i, j)

conditions	interval	
$\underline{s}_i \leq \underline{s}_j \wedge \overline{e}_j \geq \overline{e}_i$	$[\underline{s}_i, \overline{e}_j]$	A
$\underline{s}_i \geq \underline{s}_j \wedge \underline{s}_i \leq \underline{e}_j \wedge \underline{s}_i \leq \overline{s}_j \wedge \underline{s}_j + \overline{e}_j - \underline{s}_i \geq \overline{e}_i$	$[\underline{s}_i, \underline{s}_j + \overline{e}_j - \underline{s}_i]$	B
$\underline{s}_i \geq \underline{s}_j \wedge \underline{s}_i \leq \underline{e}_j \wedge \underline{e}_j \geq \overline{e}_i$	$[\underline{s}_i, \underline{e}_j]$	C
$\underline{s}_i \leq \underline{s}_j \leq \overline{e}_j \leq \underline{e}_i$	$[\underline{s}_i, \overline{e}_j]$	D
$\underline{s}_i \geq \underline{s}_j \wedge \underline{s}_i \leq \underline{e}_j \wedge \underline{s}_i \leq \overline{s}_j \wedge \underline{s}_j + \overline{e}_j \leq \overline{s}_i + \underline{e}_i \wedge \underline{s}_j + \overline{e}_j \geq 2 \times \overline{s}_i$	$[\underline{s}_i, \underline{s}_j + \overline{e}_j - \underline{s}_i]$	E
$\underline{s}_j \leq \underline{s}_i \leq \underline{e}_j \leq \underline{e}_i$	$[\underline{s}_i, \underline{e}_j]$	F
$\overline{e}_j \leq \overline{e}_i \wedge \overline{e}_j \geq \overline{s}_i \wedge \overline{e}_j \geq \underline{e}_i \wedge \underline{s}_i + \overline{e}_i \leq \overline{s}_j + \underline{e}_j$	$[\underline{s}_i + \overline{e}_i - \overline{e}_j, \overline{e}_j]$	G
$\underline{e}_j \leq \overline{e}_i \wedge \underline{e}_j \geq \overline{s}_i \wedge \underline{e}_j \geq \underline{e}_i \wedge \underline{s}_i + \overline{e}_i \leq \underline{s}_j + \overline{e}_j \wedge \underline{s}_i + \overline{e}_i \leq 2 \times \underline{e}_j$	$[\underline{s}_i + \overline{e}_i - \underline{e}_j, \underline{e}_j]$	H

We have characterized for an activity a the possible value at which condition (3) holds. By symmetry, we can deduce the value of the starting time at which condition (2) holds. Table 1 summarizes the intervals of interest of a pair of activities (i, j) .

Our characterization is sharper than the one proposed by Schwindt: For instance, case *B* in Table 1 is sharper than the equivalent case *iii* in table 3.5 page 84 in [8]. We may notice that intervals of interest may only start at values of the form $\underline{s}_a, \overline{s}_a$ or end at values $\underline{e}_a, \overline{e}_a$. This leads to a lighter algorithm for the $O(n^2)$ checker proposed by Baptiste et al. [2]. We present the checker for interval with starting dates in $O_S = \{\underline{s}_a, \forall a \in \mathcal{A}\} \cup \{\overline{s}_a, \forall a \in \mathcal{A}\}$ as possible starting values (as CuSP is symmetric).

We define an event as a pair (time, activity). Let \mathcal{E}_M be the events for the last completion times, $(\overline{e}_a, a) \forall a \in \mathcal{A}$ and ordered in increasing order of time. Similarly, we define $\mathcal{E}_m = (\underline{e}_a, a) \forall a \in \mathcal{A}$, $\mathcal{S}_M = (\overline{e}_a, a) \forall a \in \mathcal{A}$ and $\mathcal{L} = (\underline{s}_a + \overline{e}_a, a) \forall a \in \mathcal{A}$.

```

1  foreach  $t_1 \in O_S$  do
2      slope =  $C - \sum_a MI^a(t_1, t_1 + 1)$ ;
3      Load = 0;  $t_2^{old} = t_1$ ;
4       $\mathcal{L}' = \{(t' - t_1, a) \mid (t', a) \in \mathcal{L}\}$ ;
5      foreach event( $t_2, a$ ) in  $\mathcal{S}_M, \mathcal{E}_m, \mathcal{E}_M, \mathcal{L}'$  do
6          Load += slope  $\times (t_2 - t_2^{old})$ ;
7          if Load < 0 then Fail;
8          if event is a SoC $_a$  then
9              | slope -=  $h_a$ ;
10         else if event is a EoC $_a$  then
11             | slope +=  $h_a$ ;
12         end
13          $t_2^{old} = t_2$ ;
14     end
15 end
```

// The slope represents the evolution of the Slack over time.
// Events are evaluated in increasing order of their time. Starting on $t_1 + 1$.

Algorithm 1: Energetic Reasoning Checker.

4 Experiments and Future Work

Experiments were run on a 2.9 GHz Intel Core i7, in Choco [9] version 3 (release 13.03). In order to check the gain obtained with the new characterization we have considered random instances and instances from the PSPLIB[6]. 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* [5] search strategy, and three checkers: Algorithm 1, Baptiste et al. checker and the basic $O(n^3)$ checker. The number of nodes is identical for all instances for the three checkers, as expected. Table 2 shows a running time improvement of 20 to 30% using Algorithm 1, in comparison with Baptiste et al. algorithm.

We also used our checker in combination of the Time-Table filtering algorithm [7], to compare that combination with Time-Table Edge-Finding filtering algorithm [10].

We fixed a solving time limit of five minutes. Surprisingly, when proving optimality on random instances with a single resource, using the checker proved optimality for 72 of the 100 generated instances, while TTEF was unable to do so. This proves that associating this new checker with a Time-Table approach could be a good default propagator in constraint solvers.

Instances	Algorithm 1 ($\mu s/node$)	Baptiste et al. ($\mu s/node$)	$O(n^3)$ ($\mu s/node$)
Random10	16.47	24.97	29.31
Random20	43.95	56.24	78.74
PspLib 30	450.67	618.77	1268.92
PspLib 120	1 339.24	1 683.26	11 288.54

Table 2: Average time per node.

	TT	TT + TTEF	TT + Algorithm 1
Random20	6	7	72

Table 3: #proved optimum over 100 instances.

As a future work, our results should be confirmed by using dedicated search heuristics, which was not the case in our preliminary experiments. Moreover, as a direct consequence of our new case-based characterization, we can answer affirmatively to the second theoretical open question: The interval of interests of Baptiste et al. is complete for the propagation algorithm. As the proof needs a better enlightenment and cannot easily fit in a short paper, we will present it in a new paper.

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. 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.
5. Robert M. Haralick and Gordon L. Elliot. Increasing tree search efficiency for constraint satisfaction problems. *Artificial Intelligence*, 14(3):263–313, 1980.
6. Rainer Kolisch and Arno Sprecher. PspLib – a project scheduling problem library. *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH*, 96:205–216, 1996.
7. 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.
8. Christoph Schwindt. *Verfahren zur Lösung des ressourcenbeschränkten Projektdauerminimierungsproblems mit planungsabhängigen Zeitfenstern*. PhD thesis, Fakultät für wirtschaftswissenschaften der Universität Fridericiana zu Karlsruhe, 1998. in German.
9. CHOCO Team. Choco: an open source Java CP library. Research report 10-02-INFO, Ecole des Mines de Nantes, 2010.
10. 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.