Computing lower bounds for the cumulative scheduling problem

Jacques Carlier[1], Antoine Jouglet[1], and Abderrahim Sahli[2]

[1] Heudiasyc UMR CNRS 7253, Sorbonne Universités, Université de Technologie de Compiègne, Compiègne, France
jacques.carlier@hds.utc.fr, antoine.jouglet@hds.utc.fr
[2] LIGM UMR CNRS 8049, Université Paris-Est-Marne-La-Vallée, Paris, France
abderrahim.sahli@esiee.fr

Keywords: Cumulative Scheduling Problem, Energetic Reasoning, Lower Bounds.

1 Introduction

In this paper, we consider the Cumulative Scheduling Problem (Carlier 1987). An instance of this problem is composed of a set of n tasks $J = \{1, \ldots, n\}$. These tasks have to be scheduled without preemption by a resource of a given capacity C. Each task $i \in J$ cannot be scheduled before its release date r_i, has a duration p_i, is characterized by a tail q_i and needs c_i units of the resource to be processed. A schedule consists in assigning a starting time $s_i \geq r_i$ to each task i in such a way that the capacity of the resource is never exceeded : $\forall t, \displaystyle\sum_{i \in \{j \in J | s_j \leq t < s_j + p_j\}} c_i \leq C$.

In this paper, we propose some algorithms to compute lower bounds for the optimisation version of the cumulative scheduling problem (CuSP Optimisation). In CuSP Optimisation we have to find a schedule which minimizes the makespan $C_{max} = \max_{i \in J} \{s_i + p_i + q_i\}$. Let C^*_{max} be the optimal value of a given instance of CuSP Optimisation. The special case of CuSP Optimisation where $\forall i \in J, c_i = 1$ corresponds to the m-parallel machine scheduling problem $Pm|r_i, q_i|C_{max}$. Several lower bounds of C^*_{max} have been described for $Pm|r_i, q_i|C_{max}$ (Horn 1974, Labetoulle *et. al.* 1984, Carlier and Pinson 1998, Haouari 2003). In (Carlier, Pinson, Sahli and Jouglet submitted), we provided caracterizations of some lower bounds for CuSP Optimisation to analyse their structural differences. It leaded to the elaboration of new algorithms for Energetic Reasoning (ER) (Baptiste *et. al.* 2001) and we discussed the transformation of the destructive energetic bound (the ER based checker for CuSP Decision(C_{max})) into constructive energetic lower bounds of C^*_{max}. In the remainder let $LB_0(J) = \max_{i \in J} \{r_i + p_i + q_i\}$ be a trivial lower bound which can be easily computed in $O(n)$ time. The first constructive energetic lower bound, named $LB_2^{ER}(J)$, relies on particular tasks for which there is at least an interval of the time horizon in which they are necessarily scheduled because of their release dates and tails. Such tasks are called *crossing tasks*. The concept of crossing tasks is related to *core times*. The second constructive energetic lower bound, named $LB_3^{ER}(J)$ relies on ER. Both $LB_2^{ER}(J)$ and $LB_3^{ER}(J)$ were theoretically characterized in (Carlier J., Pinson E., Sahli A. and Jouglet A. submitted).

Section 2 is devoted to the introduction of the energetic approach initially proposed for the decision version of CuSP and its reformulation in the context of CuSP Optimisation. Section 3 explains the notion of crossing tasks which is the main concept used in $LB_2^{ER}(J)$ and which has also an important role in $LB_3^{ER}(J)$. We then describe an algorithm in $O(n \log n)$ time for $LB_2^{ER}(J)$. In Section 4, we describe an algorithm in $O(n^2)$ time and an algorithm in $O(\alpha(n) n \log n \log(\max_{i \in J} p_i))$ for $LB_3^{ER}(J)$, where $\alpha(n)$ is the inverse function of Ackermann.

2 The energetic reasoning in CuSP Optimisation

A lot of works of the literature considers the decision version of the CuSP in which all tasks have to be completed before a given value of C_{max}. Being given a value C_{max}, we denote this problem by CuSP Decision(C_{max}). In CuSP Decision(C_{max}), tails q_i are replaced by deadlines $d_i(C_{max}) = C_{max} - q_i$. Therefore, each task i has to processed in interval $[r_i, d_i(C_{max})]$. It can lead to unfeasible instances. The *Energetic Reasoning* (ER) (Erschler and Lopez 1990) (Erschler 1991) (Baptiste *et. al.* 1999) is a very well known technique to solve CuSP Decision(C_{max}) allowing feasibility tests and time-bound adjustments. Given a time interval $[\alpha, \delta]$, ER is based on the computation of the minimal part, named *energy*, of the tasks that must be processed in any feasible schedule between times α and δ. The minimal energy required by task i over $[\alpha, \delta]$ is obtained from positions of i that overlap as less as possible with the interval. The difference between the length of a given interval multiplied by C and the sum of the tasks energies is called the *slack* of the interval. If we can find an interval with a negative slack, then the instance is unfeasible. While the slack has to be non-negative on any interval, it is sufficient to test at most $O(n^2)$ particular intervals (Baptiste *et. al.* 1999). It permitted to exhibit a checker which runs in $O(n^2)$ time (Baptiste *et. al.* 2001). Derrien and Petit (Derrien and Petit 2014) have later reduced the number of intervals which has to be considered. Ouellet and Quimper (Ouellet and Quimper 2018) described an $O(n \log^2 n)$ algorithm. Recently, we provided a $O(\alpha(n) n \log n)$ algorithm for the checker (Carlier, Sahli, Jouglet and Pinson submitted), where $\alpha(n)$ is the inverse function of Ackermann. We also provided an $O(n^2)$ algorithm for time-bound adjustments (Carlier *et. al.* 2020).

In the context of CuSP Optimisation, we use ER in algorithms in which the value of C_{max} dynamically changes during the execution. Thus, the deadline $d_i(C_{max}) = C_{max} - q_i$ of task i is also modified. Actually, it is simpler to manipulate directly tails q_i which are constant. Therefore, we propose a reformulation of ER with tails which manipulates directly C_{max}. Instead of considering intervals, we now equivalently manipulate triplets $(\alpha, \gamma, C_{max})$ which corresponds to intervals $[\alpha, \delta = C_{max} - \gamma]$ in CuSP Decision(C_{max}).

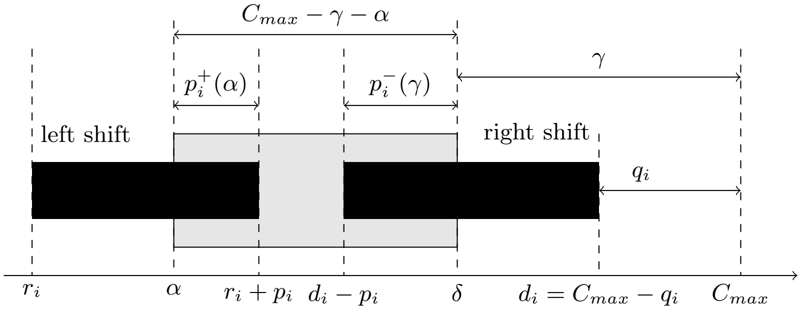

Fig. 1. Intersection energy.

For given values of C_{max}, $\alpha \in \{0, \dots, C_{max}\}$ and $\gamma \in \{0, \dots, C_{max} - \alpha\}$, we define:

- $\delta = C_{max} - \gamma$
- $\forall i \in J$ $p_i^+(\alpha) = \min(\max(0, r_i + p_i - \alpha), p_i)$, $p_i^-(\gamma) = \min(\max(0, q_i + p_i - \gamma), p_i)$ and $W_i(C_{max}, \alpha, \gamma) = c_i \min(p_i^+(\alpha), p_i^-(\gamma), C_{max} - \alpha - \gamma)$ is the energy of task i.
- The total required energy by tasks is $W(C_{max}, \alpha, \gamma) = \sum_i W_i(C_{max}, \alpha, \gamma)$. The slack, which is the difference between the maximum energy available over $[\alpha, C_{max} - \gamma]$ and the total required energy by tasks, is $S(C_{max}, \alpha, \gamma) = C(C_{max} - \gamma - \alpha) - W(C_{max}, \alpha, \gamma)$.

There exists a schedule with makespan C_{max} only if $\forall(\alpha, \gamma)$ with $\alpha \in \{0, \dots, C_{max} - 1\}$ and $\gamma \in \{0, \dots, C_{max} - \alpha - 1\}$, we have $S(C_{max}, \alpha, \gamma) \geq 0$. In fact, by adapting results of

(Baptiste *et. al.* 1999, Derrien and Petit 2014), there are only $O(n^2)$ couples (α, γ) values to consider for a given value of C_{max}.

3 $LB_2^{ER}(J)$: a constructive lower bound based on the notion of crossing-tasks

Given a makespan C_{max}, a task i is called a C_{max}-*crossing-task* if and only if there exists an interval of time in which task i is necessarily scheduled, *i.e.* if $C_{max} - q_i - p_i < r_i + p_i$. If a task i is necessarily scheduled during interval $[t, t+1)$, i is called a (C_{max}, t)-crossing-task $(t \in \{C_{max} - q_i - p_i, \ldots, r_i + p_i - 1\})$.

We provide an algorithm to compute the lower bound $LB_2^{ER}(J)$ which corresponds to the smallest value of $C_{max} \geq LB_0(J)$ for which for any time $t \in \{0, \ldots, C_{max}\}$, the sum of capacities required by (C_{max}, t)-crossing tasks in J is lower than or equal to C. Let $\chi(C_{max})$ be the set of C_{max}-crossing-tasks and let $\chi(C_{max}, t)$ be the set of (C_{max}, t)-crossing tasks. Thus, note that $LB_2^{ER}(J)$ corresponds to the smallest value $C_{max} \geq \max_{i \in J} (r_i + p_i + q_i)$ for which for any time $t \in \{0, \ldots, C_{max}\}$ we have $\sum_{i \in \chi(C_{max}, t)} c_i \leq C$.

Note that a C_{max}-crossing-task i becomes crossing at time $C_{max} - q_i - p_i$ while it is not crossing anymore from time $r_i + p_i$. Let T be the list of dates in $\{r_i + p_i | i \in J\}$. Our algorithm iterates over the different $t \in T$ in the non-increasing order. At each iteration of the algorithm, we maintain an AVL-tree CT in such a way that it contains all (C_{max}, t)-crossing-task. We also maintain a variable C_{CT} in such a way it corresponds to the sum of the capacities of the crossing-tasks over $[t-1, t)$. At each iteration, we will verify that $C_{CT}(C_{max}, t-1) \leq C$. It allows to ensure that at the end of the algorithm C_{max} has been adjusted to the smallest value $C_{max} \geq LB_0(J)$ for which for any time $t \in \{0, \ldots C_{max}\}$ we have $C_{CT}(C_{max}, t) \leq C$. To maintain these properties, we use a forward linked list allowing the tasks i which are not crossing at time $t - 1 \geq C_{max} - q_i - p_i$ to be known : these tasks have to be removed from CT. We also use another forward list allows the tasks i which are crossing at time $t - 1 \geq r_i + p_i$ to be known and which have therefore to be inserted in CT. When $C_{CT} > C$, we adjust C_{max} in such a way that $C_{CT} \leq C$. This algorithm runs in $O(n \log n)$ time. It is analogous to the sweep algorithm of (Beldiceanu and Carlsson 2002) verifying the cumulative constraint. It uses additional data structures for adjusting C_{max}.

4 LB_3^{ER}: a constructive lower bound based on the energies

We also provide two algorithms to compute $LB_3^{ER}(J)$ which corresponds to the smallest value of $C_{max} \geq LB_2^{ER}(J)$ for which for any $\forall(\alpha, \gamma)$ with $\alpha \in \{0, \ldots, C_{max} - 1\}$ and $\gamma \in \{0, \ldots, C_{max} - \alpha - 1\}$, we have $S(C_{max}, \alpha, \gamma) \geq 0$.

Our first algorithm uses twice an adjustment procedure of C_{max}. Indeed, the r_i and q_i play a symmetrical role. Therefore, for each given $\alpha \in \{r_i, r_i + p_i, C_{max} - q_i - p_i | i \in J\}$, we check the couples (α, γ) with $\gamma \in \{C_{max} - r_i - p_i, q_i + p_i, q_i, \alpha + q_i - r_i | i \in J\}$ such that $\gamma < C_{max} - \alpha$. Next, we build the instance in which the r_i and the q_i values are interchanged and we apply the same procedure. It ensures that all relevant couples (α, γ) identified by (Baptiste *et. al.* 1999, Derrien and Petit 2014) are considered during the algorithm. The adjustment procedure iteratively considers the different pertinent values of α in an outer loop while. For each value of α it then considers the pertinent values of γ in decreasing order, allowing the right bound of the associated interval to increase iteratively while we maintain the value of the required energies of the task in this interval. Each time it is detected that the slack is negative on the current interval, the value of C_{max} is adjusted and our data structures are updated to continue the consideration of the other intervals. The whole algorithm runs in $O(n^2)$ time and uses only simple data structures (arrays and forward linked lists).

Our second algorithm relies on the direct use of our checker described in (Carlier J., Sahli A., Jouglet A. and Pinson E. submitted) to do a dichotomic search on $LB_3^{ER}(J)$. The complexity is theoretically attractive : $O(\alpha(n)n\log n\log(\max_{i\in J}p_i))$, where $\alpha(n)$ is the inverse function of Ackermann.

A drawback is that we don't compute the energetic balance of each classical interval which should be useful for computing adjustments. Moreover, the checker uses very complex data structures which makes it very hard to implement.

Acknowledgements

These works are partially financed by the project 2018-0062H of the *Gaspard Monge Program for Optimization, operations research, and their interactions with data science.*

References

Baptiste P., Le Pape C. and Nuijten W., 1999, "Satisfiability tests and time-bound adjustments for cumulative scheduling problems", *Annals of Operations Research*, 92, pp. 305-333.

Baptiste P., Le Pape C., Nuijten W., 2001, "Constraint-based scheduling : applying constraint programming to scheduling problems", *International Series in Operations Research and Management Science*, vol 39, Kluwer.

Beldiceanu N. and Carlsson M., 2002, "A new multi-resource cumulative constraint with negative heights", *CP*, 2470, pp. 63-79.

Carlier J., 1987, "Scheduling jobs withe release dates and tails on identical machines to minimize the makespan", *European Journal of Operational Research*, 29, pp. 298-306.

Carlier J., Pinson E., 1998, "Jackson's pseudo preemptive schedule for the $Pm/r_i, p_i/C_{max}$ scheduling problem", *Annals of Operations Research*, 83(0), pp. 41-58 (1998)

Carlier J., Pinson E., Sahli A. and Jouglet A., to appear, "An $O(n^2)$ algorithm for time-bound adjustments for the cumulative scheduling problem" , *European Journal of Operational Research.*

Carlier J., Pinson E., Sahli A. and Jouglet A., submitted, "Comparison of three classical lower bounds for the cumulative scheduling problem".

Carlier J., Sahli A., Jouglet A. and Pinson E., submitted, "A nearly $o(n\log n)$ checker algorithm for the cumulative scheduling problem".

Derrien A., Petit T., 2014, "A new characterization of relevant intervals for energetic reasoning", In *International conference on principles and practice of constraint programming*, pp. 289-297.

Erschler J., Lopez P., 1990, "Energy-based approach for task scheduling under time and resources constraints", In *Proceedings of the 2^{nd} international workshop on project management and scheduling*, pp. 115-121.

Erschler J., Lopez P., Thuriot, 1991, "Raisonnement temporel sous contraintes de ressource et problèmes d'ordonnancement", *Revue d'Intelligence Artificielle*, 5(3), pp. 7-32.

Haouari M., Gharbi A., 2003, "An improved max-flow-based lower bound for minimizing maximum lateness on identical parallel machines", *Operations Research Letters*, 31(1), pp. 49-52.

Horn W., 1974, "Some simple scheduling algorithms", *Naval Research Logistics Quaterly*, 21, pp. 177-185.

Labetoulle J., Lawler E., Lenstra J., Rinnooy Kan ., 1984, "Preemptive scheduling of uniform machines subject to release dates", *Progress in combinatorial optimization* (Waterloo Ont, 1982), Academic Press, pp. 245-261.

Ouellet .Y, Quimper C.G., 2018, "A $O(n\log^2 n)$ checker and $O(n^2\log n)$ filtering algorithm for the energetic reasoning", In, *International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pp. 477-494.