Flow and energy based satisfiability tests for the Continuous Energy-Constrained Scheduling Problem with concave piecewise linear functions

Margaux Nattaf, Christian Artigues, Pierre Lopez

Flow and energy based satisfiability tests for the Continuous Energy-Constrained Scheduling Problem with concave piecewise linear functions

Margaux Nattaf[1,2⋆], Christian Artigues[1,3⋆⋆], and Pierre Lopez[1,3⋆⋆]

[1] CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
{nattaf,artigues,lopez}@laas.fr
[2] Univ. de Toulouse, UPS, F-31400 Toulouse France
[3] Univ. de Toulouse, LAAS, F-31400 Toulouse France

Abstract. A set of tasks consuming a continuous, cumulative and renewable resource with limited capacity, has to be scheduled. The consumption profile of the task is a decision variable of the problem that can vary continuously over time and must lie within a minimum and a maximum value once the task is started. The task is completed as soon as the integration over its time window of a non-decreasing and continuous efficiency function of the consumption profile has reached a predefined amount of energy. The goal is to find a feasible schedule, which is an NP-complete problem. For the case where functions are concave and piecewise linear, we present two satisfiability tests. The first one is the adaptation to concave functions of the variant of the energetic reasoning based satisfiability test previously proposed in [11] for linear functions. The second test is a flow-based linear program. Finally, the complementarity of the tests is assessed via their integration in a hybrid branch-and-bound.

Keywords: continuous scheduling, continuous resources, concave piecewise linear functions, energy constraints, energetic reasoning

1 Introduction

This paper deals with a scheduling problem involving a set of tasks and a continuously-divisible renewable resource of limited capacity shared by the tasks. Each task must be processed between a release date and a due date. During its time window, each task must receive a given total amount of resource units that we will refer to as a required energy amount. We consider the case where the resource amount (intensity) that a task requires during its processing is not fixed. More precisely, the resource requirement is a continuous function of time that must be determined. Once the activity is started the resource amount must lie within an interval until the total required energy has been received by the task.

⋆ Student
⋆⋆ Supervisor

Furthermore we consider that the total energy received by the task is not equal to the total amount of the resource used by it. Instead we have efficiency functions, which translates the required resource amounts into energy. Consequently, the duration of the activity is not fixed neither but is determined by the resource requirement function as the activity is finished once the necessary energy has been received.

As typical examples, we cite energy-consuming production scheduling problems. In [1], a foundry application is presented where a metal is melted in induction furnaces. Due to the complexity of the problem, efficiency functions were not considered in the paper. In a continuous time setting but still without considering the efficiency functions, constraint propagation algorithms based on the energetic reasoning concept were proposed in [2]. An extension of this work to linear efficiency functions were considered in [10, 11].

In this paper, we perform an analysis of the structural properties of the problem for more realistic concave piecewise linear functions [7, 9]. We show first that the resource demand profile of a task can be restricted to a piecewise constant function with break points at the start and ends of tasks. From these theoretical properties, we are able to compute the minimal resource consumption of a task inside an interval in $O(1)$ and we prove that the set of relevant intervals of polynomial size that was shown sufficient for energetic reasoning with linear functions is also sufficient in our case. We also define a new satisfiability test which rely on a flow-based linear program and we show that no test subsumes the other. A hybrid branch-and-bound algorithm integrating the two tests and a mixed integer linear program is evaluated on a set of instances.

2 Problem definition

In this paper, we consider the Continuous Energy-Constrained Scheduling Problem (CECSP) defined as follows. A set of tasks $A = \{1, \ldots, n\}$, consuming a continuous, cumulative and renewable resource of capacity B, has to be scheduled. At each time t, a task $i \in A$ consumes a variable amount of the resource, $b_i(t)$. The objective is to find, for each task $i \in A$, its start time st_i, its end time et_i and its resource allocation function $b_i(t)$. These quantities have to satisfy the following constraints.

First, each task i has to be executed during its time window $[r_i, d_i]$, i.e.

$$r_i \leq st_i \leq et_i \leq d_i \tag{1}$$

Then, if a task i is in process at time t, then $b_i(t)$ has to lie between a minimum and maximum requirement, b_i^{min} and b_i^{max} respectively, and has to be equal to zero otherwise, i.e.

$$b_i^{min} \leq b_i(t) \leq b_i^{max} \qquad \forall t \in [st_i, et_i] \tag{2}$$

$$b_i(t) = 0 \qquad \forall t \notin [st_i, et_i] \tag{3}$$

Note that the case where $b_i^{min} = 0$ corresponds to the preemptive case.

Furthermore, during its processing, a task receives an energy quantity from the resource. Thus, each task consumes a part of the same resource but the energy type received from the resource might be different for each task. In this sense, each task has is own conversion function, also called power processing rate function, f_i and a task is finished when it has received a required amount of energy W_i [1], i.e.

$$\int_{st_i}^{et_i} f_i(b_i(t))dt = W_i \tag{4}$$

Thus, function f_i has to be integrable. Hence, in this paper, we only consider continuous functions.

The last constraint is the resource capacity constraint. At each time t, the resource consumed by all tasks can not exceed the resource capacity, i.e.

$$\sum_{i \in A} b_i(t) \leq B \quad \forall t \tag{5}$$

In this paper, we consider the case where functions f_i are non-decreasing, concave, piecewise linear and for which $f_i(0) = 0$. This problem is NP-complete [11].

Theorem 1. *Let $\mathcal{I}$ be a feasible instance of CECSP, with non-decreasing, concave piecewise linear functions f_i such that $f_i(0) = 0,\ \forall i \in A$. A solution such that, for all $i \in A$, $b_i(t)$ is piecewise constant, exists. Furthermore, $\forall i \in A$ the only breakpoints of $b_i(t)$ can be restricted to the start and end times of the tasks.*

In order to prove Th.1, we start by proving that, for any interval $[t_1, t_2]$ where $b_i(t)$ is not constant, we can find a constant b_{iq} for which executing i at b_{iq} during $[t_1, t_2]$ gives more energy to i and consumes as much resource. This constant is equal to the mean value of $b_i(t)$ over $[t_1, t_2]$, i.e. $b_{iq} = \int_{t_1}^{t_2} b_i(t)dt/(t_2 - t_1)$.

Lemma 1. [1] *Let $b_{iq} = \frac{\int_{t_1}^{t_2} b_i(t)dt}{t_2 - t_1}$. Then, we have :*

$$\int_{t_1}^{t_2} b_{iq}dt = \int_{t_1}^{t_2} b_i(t)dt \tag{6}$$

$$\int_{t_1}^{t_2} f_i(b_{iq})dt \geq \int_{t_1}^{t_2} f_i(b_i(t))dt \tag{7}$$

Proof. Equation (6) is trivially verified by replacing b_{iq} by its value. To prove that equation (7) is satisfied, we use the following theorem, due to Jensen [8].

Theorem 2 (Jensen). *Let $\alpha(t)$ and $g(t)$ be two integrable functions on $[t_1, t_2]$ such that $\alpha(t) \geq 0,\ \forall t \in [t_1, t_2]$. We have:*

$$\phi\left(\frac{\int_{t_1}^{t_2} \alpha(t)g(t)dt}{\int_{t_1}^{t_2} \alpha(t)dt}\right) \geq \frac{\int_{t_1}^{t_2} \alpha(t)\phi(g(t))dt}{\int_{t_1}^{t_2} \alpha(t)dt} \tag{8}$$

where ϕ is a continuous concave funtion in $[\min_{t\in[t_1,t_2]} g(t), \max_{t\in[t_1,t_2]} g(t)]$.

[1] see illustration in Appendix

Replacing $\phi(t)$ by $f_i(t)$, $g(t)$ by $b_i(t)$ and $\alpha(t)$ by the constant function equal to 1, we obtain the desired result. □

We are now able to prove Th.1. Actually, we prove that if a solution S exists, then another solution S' can be created from S with the property that each function $b'_i(t)$ is piecewise constant.

Proof (Th. 1). Let S be a feasible solution of $\mathcal{I}$ and let $(t_q)_{\{q=1..Q\}}$ be the increasing series of distinct start time and end time values ($Q \leq 2n$). For ease of notation, we define intermediary functions $\tilde{b}_i(t),\ \forall i \in A$ as follows:

$$\tilde{b}_i(t) = \begin{cases} b_{i0} & \text{if } t \in [t_0, t_1] \\ \quad\vdots & \\ b_{i(Q-1)} & \text{if } t \in [t_{Q-1}, t_Q] \end{cases} \qquad \text{with } b_{iq} = \frac{\int_{t_q}^{t_{q+1}} b_i(t)dt}{t_{q+1}-t_q}$$

Then, S' is constructed in the following way:

$$\begin{array}{l} st'_i = st_i \\ et'_i = \min(\tau | \int_{st_i}^{\tau} f_i(\tilde{b}_i(t))dt = W_i) \end{array} \qquad b'_i(t) = \begin{cases} \tilde{b}_i(t) & \text{if } t \in [st_i, et'_i] \\ 0 & \text{otherwise} \end{cases}$$

It is easy to see that S' satisfies the time window constraints (1), as, by Lemma 1, $et'_i \leq et_i$. Furthermore, S' verifies the energy constraints (4) since it is defined in this way. Then, S' also verifies the resource capacity constraints (5). Indeed, as S is a feasible solution, we have $\forall q \in \{1, \ldots, Q\}$ and $\forall t \in [t_q, t_{q+1}]$: $\sum_{i \in A} b_i(t) \leq B \Rightarrow \sum_{i \in A} \int_{t_q}^{t_{q+1}} b_i(t)dt \leq B(t_{q+1} - t_q)$.

And, then : $\sum_{i \in A} b'_i(t) \leq \sum_{i \in A} \tilde{b}_i(t) = \sum_{i \in A} b_{iq} = \sum_{i \in A} \frac{\int_{t_q}^{t_{q+1}} b_i(t)dt}{t_{q+1}-t_q} \leq B$. Finally, we can show that S' satisfies the resource requirement constraints (2)-(3) in a similar way. □

An interesting remark can be made about Th.1. Actually, in order to find a solution to CECSP, we only have to find, for each task, its start time st_i, its end time et_i and the quantity of resource allocated to i between two consecutive start/end times b_{iq}.

3 Energetic reasoning

In this section, we present an extension of the energetic reasoning for the CECSP with non-decreasing, linear function f_i [11] to the CECSP with non-decreasing, concave and piecewise linear f_i.

In order to present this satisfiability test, we define two quantities: the minimum resource consumption (resp. energy requirement) of a task i over an interval $[t_1, t_2]$, $\underline{b}(i, t_1, t_2)$ (resp. $\underline{w}(i, t_1, t_2)$). Then, the energetic reasoning consists in testing whether the available area within $[t_1, t_2]$ ($B \times |[t_1, t_2]|$) is large enough to contain the minimum resource quantity needed by all the tasks in this interval.

Theorem 3 ([6]). *Let $\mathcal{I}$ be an instance of CECSP. If it exists (t_1, t_2) such that $B(t_2 - t_1) - \sum_{i \in A} \underline{b}(i, t_1, t_2) < 0$ then $\mathcal{I}$ is infeasible.*

Proof. Left to the reader.

To compute the minimum required resource consumption, we, first, have to compute $\underline{w}(i, t_1, t_2)$. Given an interval $[t_1, t_2]$, the minimum consumption always corresponds to a configuration where task i is either left-shifted or right-shifted or both[1]. Therefore, $\underline{w}(i, t_1, t_2)$ can be computed exactly in the same way as in [11]. Hence, it is not described in this paper.

Thus, let J be the interval over which task i must receive an energy $\underline{w}(i, t_1, t_2)$, i.e. $J = [t_1, t_2] \cap [r_i, d_i]$. First, we claim that processing a task i at b_i^{min}, has the best efficiency ratio, i.e. $\max_{x \in [b_i^{min}, b_i^{max}]} f_i(x)/x = f_i(b_i^{min})/b_i^{min}$ (see Proposition 1). We have two cases to consider:

- J is sufficiently large to schedule the task at b_i^{min}, i.e. $|J| \geq \frac{\underline{w}(i,t_1,t_2)}{f_i(b_i^{min})}$, and then $\underline{b}(i, t_1, t_2) = b_i^{min} \frac{\underline{w}(i,t_1,t_2)}{f_i(b_i^{min})}$
- J is not large enough to schedule the task at b_i^{min} and finding $\underline{b}(i, t_1, t_2)$ is equivalent to solving:

$$\begin{aligned} &\text{minimize} && \int_J b_i(t)dt \\ &\text{subject to} && \int_J f_i(b_i(t))dt \geq \underline{w}(i, t_1, t_2) \end{aligned}$$

In the latter case, thanks to lemma 1, we know that it exists a solution where $b_i(t)$ is constant and this constant is equal to $f_i^{-1}(W_i/|J|)$. Since f_i is a non-decreasing concave piecewise linear function this value can easily be computed. Indeed, let P_i be the number of pieces of function f_i and let $\gamma_p^i, \forall p \in \{1, \ldots, P_i\}$ be the break points of f_i. Thus, $f_i(b) = a_{ip} * b + c_{ip},\ \forall b \in [\gamma_p^i, \gamma_{p+1}^i]$ and $f_i^{-1}(W_i/|J|) = \max_p(\frac{\underline{w}(i,t_1,t_2) - c_{ip}|J|}{a_{ip}|J|})$.

Therefore, the expression of $\underline{b}(i, t_1, t_2)$ is the following:

$$\underline{b}(i, t_1, t_2) = \max(b_i^{min} \frac{\underline{w}(i, t_1, t_2)}{f_i(b_i^{min})}, \max_p(\frac{1}{a_{ip}}(\underline{w}(i, t_1, t_2) - |J|c_{ip}))) \tag{9}$$

if $b_i^{min} \neq 0$, else $\underline{b}(i, t_1, t_2) = \max(0, \max_p(\frac{1}{a_{ip}}(\underline{w}(i, t_1, t_2) - |J|c_{ip})))$. Note that, in this case, the energetic reasoning can be very poor.

Now, we have to prove that $\max_{x \in [b_i^{min}, b_i^{max}]} f_i(x)/x = f_i(b_i^{min})/b_i^{min}$. To do so, we use the following proposition:

Proposition 1. *Let g be a function defined on an interval J and $\forall t \in J \setminus \{\delta\}$, $\phi_\delta(t) = \frac{g(t)-g(\delta)}{t-\delta}$. Then, g is concave if and only if $\phi_\delta(t)$ is decreasing, $\forall \delta \in J$.*

In particular, when $\delta = 0$ and since $f_i(0) = 0$, $\frac{f_i(t)}{t}$ is decreasing. Thus, $\max_{x \in [b_i^{min}, b_i^{max}]} f_i(x)/x = f_i(b_i^{min})/b_i^{min}$.

Indeed, this result also implies that the relevant intervals, i.e. the intervals on which we have to perform the test, described in [11] adapting from [5] are exactly the ones for our problem. Furthermore, the time window adjustments, also described in [11] adapting the ones in [3], are still valid in our case.

[1] see illustration in Appendix

4 Flow-based checker

In this section, we describe a linear program which helps to detect infeasibility. Indeed, if the program is infeasible, then there is no solution for the CECSP.

To describe this program, let $(t_q)_{q\in\mathcal{Q}}$ be the increasing series of distinct domain bounds of the start and end times values and let s_i^{max} (resp. e_i^{min}) be the latest start (resp. earliest end) time of i. Then, for each interval $[t_q, t_{q+1}]$ and for each task, we define two variables b_{iq} and w_{iq}, which stand for the quantity of resource (resp. energy) used (resp. received) by task i in this interval.

$$\max \quad \sum_{i\in A}\sum_{e\in\mathcal{E}} w_{iq} \tag{10}$$

$$\sum_{i\in A} b_{iq} \leq B(t_{q+1} - t_q) \qquad \forall q \in \mathcal{Q} \tag{11}$$

$$b_{iq} \geq b_i^{min}(t_{q+1} - t_q) \qquad \forall i \in A\ ; \forall q \in \mathcal{Q} | s_i^{max} \leq t_q \leq e_i^{min} \tag{12}$$

$$b_{iq} \leq b_i^{max}(t_{q+1} - t_q) \qquad \forall i \in A\ ; \forall q \in \mathcal{Q} \tag{13}$$

$$b_{iq} = 0 \qquad \forall i \in A\ ; \forall q \in \mathcal{Q} | t_q \notin [r_i, d_i] \tag{14}$$

$$w_{iq} \leq a_{ip} b_{iq} + c_{ip}(t_{q+1} - t_q) \qquad \forall i \in A\ ; \forall q \in \mathcal{Q}\ ; \forall p \in \{1..P_i\} \tag{15}$$

$$\sum_{q\in\mathcal{Q}} w_{iq} = W_i \qquad \forall i \in A \tag{16}$$

The linear program is based on the following observation: in all intervals $[t_q, t_{q+1}] \subseteq [s_i^{max}, e_i^{min}]$, task i has to be scheduled, at least, at b_i^{min}. These constraints are expressed by (12). Constraints (11) model the resource capacity constraints. Constraints (13) impose that the maximum resource requirement constraints are satisfied. Constraints (14) set the resource consumption of task i to be equal to 0 in $[t_q, t_{q+1}]$ if $[t_q, t_{q+1}] \not\subseteq [r_i, d_i]$. Constraints (15) combined with the objective function, ensure the resource conversion. Finally, constraints (16) state that the tasks received the required energy.

In some cases, this linear program is stronger than the energetic reasoning. For example, consider the following instance, with $n = 3,\ B = 3,\ r_i = \{0, 4, 0\},\ d_i = \{2, 6, 6\},\ W_i = \{4, 4, 10\},\ b_i^{min} = \{2, 2, 1\},\ b_i^{max} = \{2, 2, 2\}$ and $f_i(b) = b,\ \forall i = \{1, 2, 3\}$.

The relevant intervals for the energetic reasoning are: $[0, 2],\ [0, 4],\ [0, 6],\ [2, 4],\ [2, 6],\ [4, 6]$. Figure 1 illustrates the energetic reasoning on some of these intervals. It can be seen that no infeasibility is detected, since, for each intervals, the sum of minimum consumption never exceeds the interval capacity.

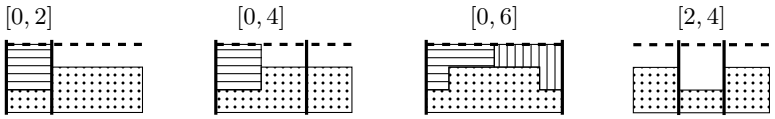


Fig. 1: Energetic reasoning on four of the six relevant intervals

Consider now the flow-based LP defined previously. We have six relevant time points: $t_0 = 0,\ t_1 = 1,\ t_2 = 2,\ t_3 = 4,\ t_4 = 5,\ t_5 = 6$ and b_{iq} corresponds to the amount of resource allocated to task i in time slot $[t_q, t_{q+1}]$. Note that we do not need variable w_{iq} since $f_i(b) = b$ is the identity function. Writing the LP, we obtain:

$$b_{10} = b_{11} = 2,\ b_{1q} = 0 \quad q = 2..4 \quad (17)$$

$$b_{23} = b_{24} = 2,\ b_{2q} = 0 \quad q = 0..2 \quad (18)$$

$$b_{3q} \leq 2 \quad q = 0, 4 \quad (19)$$

$$1 \leq b_{3q} \leq 2 \quad q = 1, 3 \quad (20)$$

$$b_{3q} + 2 \leq 3 \quad q \neq 2 \quad (21)$$

$$2 \leq b_{32} \leq 4 \quad (22)$$

$$b_{30} + b_{31} + b_{32} + b_{33} + b_{34} = 10 \quad (23)$$

$$b_{32} \leq 6 \quad (24)$$

As constraints (21) bound b_{3q} from above by 1, for $q = 0, 1, 3, 4$, constraint (23) implies that $b_{32} = 6$ which contradicts constraint (22). Thus, the LP detects the infeasibility of the instance whereas the energetic reasoning fails to prove it.

5 Hybrid branch and bound

In this section, we describe the hybrid branch-and-bound algorithm we used to solve the CECSP. This algorithm is an adaptation of the one in [11] for linear functions.

First, a branch-and-bound algorithm is used to reduce the size of possible start and end intervals (until their size is less than a given $\epsilon > 0$) and, then, a start/end event-based MILP is used in order to find exact task start and end times and to determine the quantity of resource allocated to i between two consecutive events. The event-based MILP used for our case is the same used for linear functions except that constraints:

$$W_{ie} \leq a_i B_{ie} + c_i(t_{e+1} - t_e) \quad \forall i \in A\ ; \forall e \in \mathcal{E}$$

are replaced by constraints:

$$W_{ie} \leq a_{ip} B_{ie} + c_{ip}(t_{e+1} - t_e) \quad \forall i \in A\ ;\ \forall p \in P_i;\ \forall e \in \mathcal{E}$$

The branching procedure is inspired by the work of Carlier *et al.* [4]. At the beginning, a task can start (resp. end) at any time $st_i \in [r_i, s_i^{max}]$ (resp $et_i \in [e_i^{min}, d_i]$). The idea is, at each node, to reduce the size of one of these intervals. For example, suppose that we choose to reduce the start time interval of i, then we create two nodes: one with constraint $st_i \in [r_i, (r_i + s_i^{max})/2]$ and one with constraint $st_i \in [(r_i + s_i^{max})/2, s_i^{max}]$. The variable on which we will branch are choosed with the following heuristic: we choose the smallest interval among all $[r_i, s_i^{max}]$ and $[e_i^{min}, d_i]$.

At each node, we apply one or both of the satisfiability tests described above and, if the test does not fail, we perform the time-window adjustments found in [11]. We continue this procedure using a depth-first strategy until all intervals are smaller than an ϵ. When it happens, the remaining solution space is searched via the event-based MILP.

6 Experiments

The experiments are conducted on an Intel Core i7-4770 processor with 4 cores and 8 gigabytes of RAM under the 64-bits Ubuntu 12.04 operating system. The hybrid branch-and-bound algorithm is coded in C++ and uses CPLEX 12.6 with 1 threads at each leaf.

First, the instances have been generated with identity power processing rate functions, i.e. $f_i(b) = b,\ \forall i \in A$. We generated 129 instances of $n = 30$ tasks according to the following framework. The resource availability B is set to 10 and three parameters regulate the dispersion of the data: $scale$, mf and df. Then, r_i, b_i^{min}, b_i^{max} are randomly generated in their corresponding interval: $b_i^{max} \in [1, B]$, $b_i^{min} \in [0, b_i^{max}/2]$, $r_i \in [0, (df * n * B * scale)/2]$; W_i is set to $\max(1, b_i^{min} + \delta)$, where δ is generated with a binomial distribution $\mathcal{B}(scale * B, 0.5)$, and d_i is set to $r_i + W_i/b_i^{max} + \sigma$, where σ is uniformly generated in $[1, mf * scale * B]$.

Then, we transform them in order to obtain instances with power processing rate functions in the following way. We randomly generated the parameters of the function: $P_i \in [1, b_i^{max} - b_i^{min}]$, the number of pieces of f_i, then the break points γ_p^i of f_i such that $\gamma_{p-1}^i < \gamma_p^i$, and finally, for each piece, a_{ip} within interval $[1, 10]$ and c_{ip} is calculate to preserve continuity.

Table 1 presents the results of the hybrid branch-and-bound algorithm with parameter $\epsilon = 2.5$ with a total time limit of 7200 seconds. Row 1 corresponds to the results with both satisfiability tests, row 2 with only energetic reasoning and row 3 with only the flow-based checker. In the table, "solved" mean either proved infeasible or that a solution is found.

	%solved	#nodes	%ER cut	%flow cut
both	96.1	25.5	13.1	12.5
flow	95.3	26	X	24.71
ER	86.8	17.33	13.1	X

Table 1: Results of the comparison of the two satisfiability test.

We can see that the flow-based satisfiability test solves more instances than the energetic reasoning. However, it does not dominate the energetic reasoning since performing both tests is better.

7 Conclusion

We demonstrated that the resource usage changes can be restricted to start and end times of tasks. This allows us to present two polynomial satisfiability tests for the CECSP with concave piecewise linear power processing rate functions and we compare them in a solution method. We also showed the interest of both methods since none of them dominates the other one.

In the continuity of this work, it will be interesting to establish the instance properties that make one test stronger than the other. Studying more general functions is also a challenging research direction.

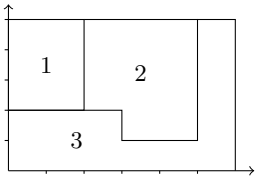
References

1. C. Artigues, P. Lopez, and A. Haït (2013). The energy scheduling problem: Industrial case study and constraint propagation techniques. *International Journal of Production Economics*, 143(1):13-23.
2. C. Artigues and P. Lopez. Energetic reasoning for energy-constrained scheduling with a continuous resource. *Journal of Scheduling*, DOI:10.1007/s10951-014-0404-y.
3. P. Baptiste, C. Le Pape and W. Nuijten (2001). *Constraint-based scheduling*, Kluwer Academic Publishers, Boston/Dordrecht/London.
4. J. Carlier and B. Latapie (1991). Une méthode arborescente pour résoudre les problèmes cumulatifs. *RAIRO. Recherche opérationnelle*, 25(3): 311–340.
5. A. Derrien and T. Petit (2014). A new characterization of relevant intervals for energetic reasoning. *Principles and Practice of Constraint Programming*, Lecture Notes in Computer Science, 8656, pp. 289–297.
6. J. Erschler and P. Lopez (1990). Energy-based approach for task scheduling under time and resources constraints. *2nd International Workshop on Project Management and Scheduling*, pp. 115-121, Compiègne, France.
7. M.N. Hung, C. Le Van and P. Michel (2005). Non-convex aggregative technology and optimal economic growth. ftp://www-bsg.univ-paris1.fr/pub/mse/cahiers2005/B05095.pdf.
8. J.L.W.V. Jensen (1906). Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30(1): 175–193.
9. J. Lewis. Algebra symposium: Optimizing fuel consumption. http://homepages.math.uic.edu/ jlewis/math165/asavgcost.pdf.
10. M. Nattaf, C. Artigues and P. Lopez (2014). A polynomial satisfiability test using energetic reasoning for energy constraint scheduling. *14th International Workshop on Project Management and Scheduling*, pp. 169–172, Munich, Germany.
11. M. Nattaf, C. Artigues and P. Lopez (2015). A hybrid exact method for a scheduling problem with a continuous resource and energy constraints, *Constraints*, DOI:10.1007/s10601-015-9192-z.

Appendix

Example of a CECSP instance and solution

In the example of Fig. 1, we can see that, the energy received by task 2 is equal to $(2 \times 3 + 1) + (2 \times 4 + 1) + (2 \times 4 + 1) = 25$ which is not equal to the amount of resource consumed (11 in this case).



$B = 5$

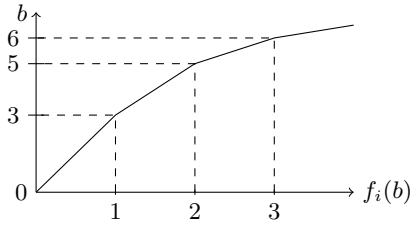
i	r_i	d_i	W_i	b_i^{min}	b_i^{max}	$f_i(b)$
1	0	2	6	3	3	b
2	1	5	25	3	4	$2b+1$
3	0	6	39	1	5	$3b$

Fig. 1: An example of instance and corresponding solution of CECSP.

Illustration of Lemma 1

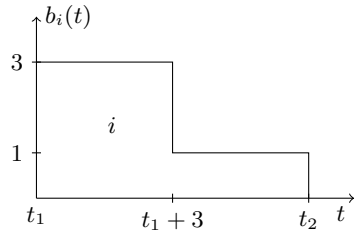
Figure 2 presents the value of functions $b_i(t)$ and $f_i(b_i(t))$ before and after applying Lemma 1. In figure 2b, $b_i(t)$ is equal to 3 between t_1 and $t_1 + 3$ and equal to 1 between $t_1 + 3$ and $t_2 = t_1 + 6$. Thus $\int_{t_1}^{t_2} b_i(t)dt = 12$ and $b_{iq} = 12/6 = 2$. Applying Lemma 1 set $b_i(t)$ to 2 between t_1 and t_2 (figure 2c) and we also have $\int_{t_1}^{t_2} b_i(t)dt = 12$.

Using function $f_i(b)$ defined by figure 2a, $f_i(b_i(t))$ is equal to 6 between t_1 and $t_1 + 3$ and equal to 3 between $t_1 + 3$ and t_2 (see figure 2d). Thus $\int_{t_1}^{t_2} f_i(b_i(t))dt = 27$. Applying Lemma 1 set $b_i(t)$ to 2 between t_1 and t_2 and then, $f_i(b_i(t)) = 5$ on this interval (figure 2e). Hence, $\int_{t_1}^{t_2} f_i(b_i(t))dt = 30 \geq 27$.

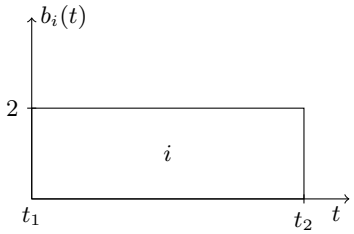


(a) function $f_i(b)$

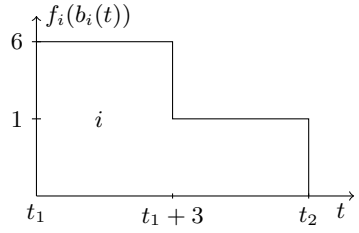
Fig. 2: Illustration of Lemma 1 (Part 1)



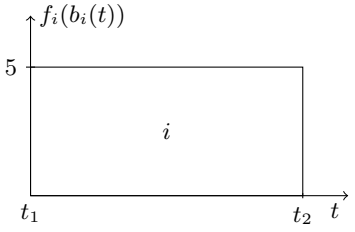
(b) $b_i(t)$ before applying Lemma 1



(c) $b_i(t)$ after applying Lemma 1



(d) $f_i(b_i(t))$ before applying Lemma 1



(e) $f_i(b_i(t))$ after applying Lemma 1

Fig. 2: Illustration of Lemma 1 (Part 2)

Possible configurations for minimal energy consumption

Figure 3 represents the different configurations leading to a minimal energy consumption ($\underline{w}(i, t_1, t_2)$) inside $[t_1, t_2]$. Configurations figs. 3a, 3d and 3g correspond to the case where i is left-shifted, configurations figs. 3c, 3f and 3i to the case where i is right-shifted and configurations figs. 3b, 3e and 3h to the case where i is both.

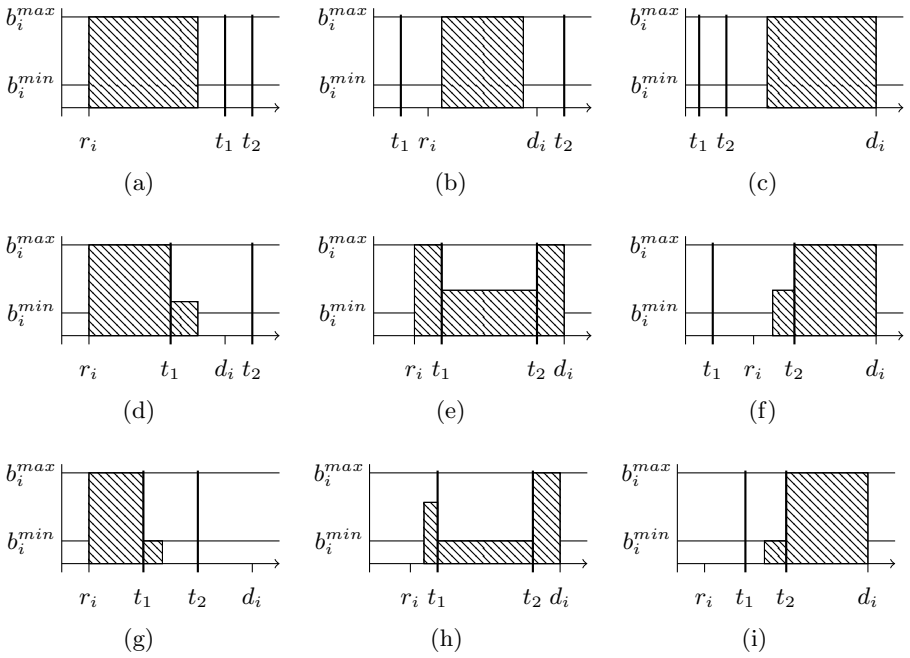


Fig. 3: Possible configurations for minimal consumptions