



Scheduling under energy constraints

Christian Artigues, Pierre Lopez, Alain Haït

► To cite this version:

Christian Artigues, Pierre Lopez, Alain Haït. Scheduling under energy constraints. International Conference on Industrial Engineering and Systems Management (IESM 2009), May 2009, Montréal, Canada. pp.CD-Rom Session 5D. hal-00367614

HAL Id: hal-00367614

<https://hal.science/hal-00367614>

Submitted on 11 Mar 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Scheduling under energy constraints [★]

Christian ARTIGUES^a, Pierre LOPEZ^a, Alain HAÏT^b

^a*CNRS; LAAS; 7 avenue du Colonel Roche, F-31077 Toulouse, France
Université de Toulouse; UPS, INSA, INP, ISAE; LAAS; F-31077 Toulouse, France*

^b*Université de Toulouse; Institut Supérieur de l'Aéronautique et de l'Espace; 10 avenue E. Belin B.P. 54032;
F-31055 Toulouse, France.*

Abstract

In this paper we present a scheduling problem dealing with energy constraints (typically electrical energy). Mainly we propose an extension of specific resource constraint propagation techniques (known as “energy reasoning”) to efficiently prune the search space and then to facilitate its resolution. We also present dominance rules and a branching scheme to solve the problem via tree search. Finally, computational results are provided.

Key words: Production scheduling, energy constraints, constraint propagation, energy reasoning

1 Introduction

Since the last two decades, hard combinatorial problems, mainly in scheduling, have been the target of many approaches combining Operations Research and Artificial Intelligence techniques. These approaches are generally focussed on constraint satisfaction as a general paradigm for representing and solving efficiently such problems. At the heart of these approaches, a panel of consistency enforcing techniques is used to dramatically prune the search space.

Therefore propagation techniques dedicated to resource and time constrained scheduling problems, viewed as special instances of Constraint Satisfaction Problems (CSP), have been developed to speed up the search for a feasible schedule or to detect early an inconsistency. For instance the energy reasoning [3] has enabled the joint integration of both resource and time constraints in order to prevent the combinatorics of solving conflicts between activities in competition for limited resources.

Furthermore, it is still of interest to search for propagating novel types of constraints according to real world problems. The new environmental constraints, but also the increase of the energy cost, should prompt us to consider as a crucial and promising issue to look into the problem of the power consumption optimization in production scheduling. Real-time (processor) scheduling theory has often addressed energy constraints. However, the consideration of energy constraints in production scheduling, and particularly focussed on constraint propagation techniques, has been relatively unexplored.

[★] This paper was not presented at any other revue. Corresponding author C. Artigues. Tel. +33-561-337-907. Fax +33-561-336-936.

Email addresses: christian.artigues@laas.fr (Christian ARTIGUES), pierre.lopez@laas.fr (Pierre LOPEZ), alain.hait@isae.fr (Alain HAÏT).

The objective of this paper is to present a formal description for the propagation of energy constraints induced by machine power consumptions. The idea is to apply the generic energy reasoning in scheduling and to extend it for the consideration of electric power consumptions. This paper is then organized as follows: Section 2 presents the problem under study. A new problem, the energy scheduling problem (EnSP), is introduced. Section 3 presents a short review of related work. Section 4 reminds the reader of the bases of the energy reasoning in scheduling. Section 5 proposes the above-mentioned extensions of this reasoning to process the energy constraints. In Section 6 we present dominance rules in order to reduce the search space. In Section 7, a branching scheme is proposed to solve the problem via tree search. Computational experiments are described in Section 8.

2 The scheduling problem under energy constraints

In this section we introduce the energy scheduling problem (EnSP). We first present the related cumulative scheduling problem (CuSP). Then we present the EnSP. Finally we show how we can model an industrial parallel machine scheduling problem with electrical energy constraints as an association of an EnSP and a CuSP.

2.1 The cumulative scheduling problem

A CuSP is a sub-problem of the RCPSP (resource-constrained project scheduling problem), in which precedence constraint are relaxed and a single resource is considered at a time [1]. The problem is also called the multiprocessor task problem and is denoted $P|r_i, d_i; size_i|$ — in the well-known three field scheduling notation [2]. An instance of the CuSP can be defined as follows: a set of n activities $A = \{1, 2, \dots, n\}$ is to be processed without interruption on a given resource of *capacity* B . To each activity i are associated its resource requirement b_i , its release date r_i , its deadline d_i , and its processing time p_i (note that capacity and resource requirements are assumed to be constant over the planning horizon). The CuSP can be stated as follows. Activity i start time (st_i) and completion time ($ft_i = st_i + p_i$) have to belong to the time window $[r_i, d_i]$. Activities can be simultaneously processed according to the satisfaction of the cumulative constraint: $\sum_{i \in A} b_{it} \leq B$, for every time point t , where $b_{it} = b_i$ if $st_i \leq t < ft_i$ and $b_{it} = 0$ otherwise.

2.2 The energy scheduling problem

The energy scheduling problem (EnSP) takes as input a set of n activities $A = \{1, 2, \dots, n\}$ having to be processed without interruption using an energy resource of *capacity* (available power) B . Instead of being defined through its duration p_i and resource demand b_i , each activity is defined through its required energy W_i , and its minimum and maximum resource requirements $b_i^{\min}$ and $b_i^{\max}$ such that the number of allocated resource units (provided power) has to remain between $b_i^{\min}$ and $b_i^{\max}$. Note here that for practical motivations, we consider that changes in the power allocated to an activity only occur at discrete time periods. The EnSP amounts to find a start time $st_i \geq r_i$, a completion time $ft_i \leq d_i$ and a power allocation b_{it} such that $b_i^{\min} \leq b_{it} \leq b_i^{\max}$ for $t \in [st_i, ft_i - 1]$ and $b_{it} = 0$ otherwise. The global power limitation constraint is written $\sum_{i \in A} b_{it} \leq B$ for any time period t . We consider both b_{it} and $p_i = ft_i - st_i$ as discrete variables. Last, an energy requirement constraint $W_i \leq \sum_{t=st_i}^{ft_i-1} b_{it}$, holds for each activity i , i.e. the energy brought to i must be at least W_i . We remark that enforcing equality would yield to possibly infeasible solutions in the case where the remaining energy to be brought to an activity at a given time period is strictly lower than $b_i^{\min}$. Consequently, in accordance with practical cases, we consider the energy brought to an activity can be larger than the required one.

Consider a problem instance with $B = 5$ and 3 activities such that $W_1 = 12$, $b_1^{\min} = 1$, $b_1^{\max} = 5$, $r_1 = 0$, $d_1 = 6$, $W_2 = 12$, $b_2^{\min} = 2$, $b_2^{\max} = 5$, $r_2 = 2$, $d_2 = 6$, $W_3 = 6$, $b_3^{\min} = b_3^{\max} = 2$, $r_3 = 2$, $d_3 = 5$.

Figure 1 displays a feasible solution for the problem. One can observe that there is no solution for which all the activities have a rectangular shape.

2.3 Industrial case modeling example

As a typical application, consider a tubing plant in the Montréal area, presented in [4,7], and more precisely, the foundry where metal is melted in induction furnaces and then cast in individual billets. The

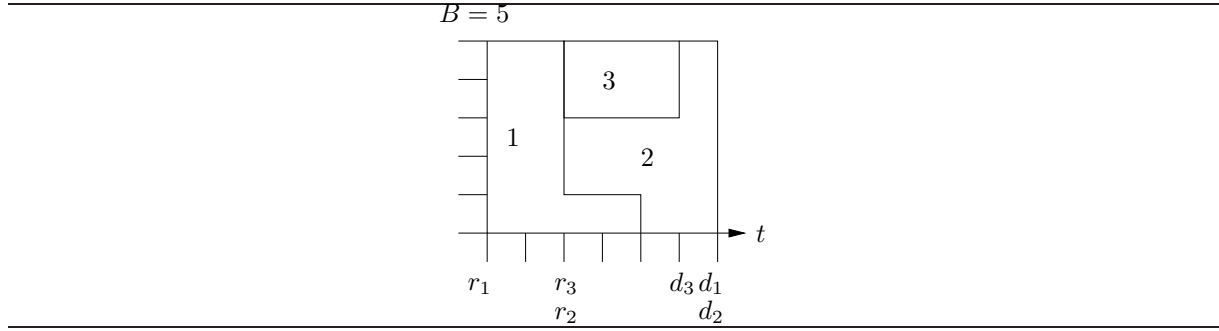


Fig. 1. Solution of an EnSP

melting operation has a variable duration that depends on the power given to the furnace (constrained by physical and operational considerations).

The foundry has five production lines. From a scheduling view-point, this facility can easily be recognized as a parallel machine problem. The electric power is a cumulative resource which is a generalization of a parallel-machine type resource.

A standard parallel machine problem can be modeled as a CuSP where activities require only one resource unit. In the considered context, electric power overheads must be closely managed. And knowing that the power consumption depends mainly on the activity of the foundry, we have to integrate these energy features in the production planning and scheduling of the melting facility.

In the application context, each operation has to be processed inside its time window by one furnace to be selected among a set of m equivalent furnaces. Once this assignment is done, operations assigned to the same furnace must be sequenced. On the other hand, each operation needs a predefined electrical energy amount E_i . The power of each furnace can be adjusted at each time period according to a minimal and maximal value $P_{\min}$ and $P_{\max}$. The power used by all furnaces at a given time period cannot exceed a global threshold P .

Hence the problem corresponds to the conjunction of a CuSP (for the parallel machine side) with $B = m$ and $b_i = 1$ for all operations i , and an EnSP (for the electric power side) with $W_i = E_i$, $B = P$, $b_i^{\min} = P_{\min}$, $b_i^{\max} = P_{\max}$. In a constraint programming framework, both subproblems can be considered separately as global constraints to perform feasibility tests and time-bound adjustments through energy reasoning. In Section 3, we show through a short literature review that feasibility tests and time-bound adjustments for the EnSP cannot be directly found in previous work. In Section 4, we recall the (standard) energy reasoning for the CuSP. In Section 5, we propose an extension of the energy reasoning for the EnSP.

3 Related work

In Section 4, we present the standard energy reasoning consisting in feasibility tests and adjustment rules for the CuSP [3]. Many variants or extensions of the CuSP have been considered, for which feasibility tests and adjustment rules have been issued. However, we show hereafter that none of these variants correspond to the EnSP.

Clearly the CuSP cannot be used to model the EnSP since activities are not necessarily of rectangular shape (see Section 4). In fact, the EnSP can be defined as a relaxation of the (continuous) CuSP. Indeed, we obtain the CuSP by setting $b_i^{\min} = b_i^{\max} = b_i$.

However in [1], other relaxations of the CuSP are considered. The *fully elastic relaxation* corresponds to a particular EnSP where $b_i^{\min} = 0$ and $b_i^{\max} = B$. Hence although the feasibility tests and adjustment rules proposed for the fully elastic CuSP hold for the EnSP, they may not capture all the structure of the EnSP since the fully elastic CuSP is itself a relaxation of the EnSP.

The *partially elastic relaxation* restricts elasticity by enforcing regularity constraints of the changes involving nominal b_i . Namely, we have $b_i^{\min} = 0$ and $b_i^{\max} = B$ as for the fully elastic case but for any

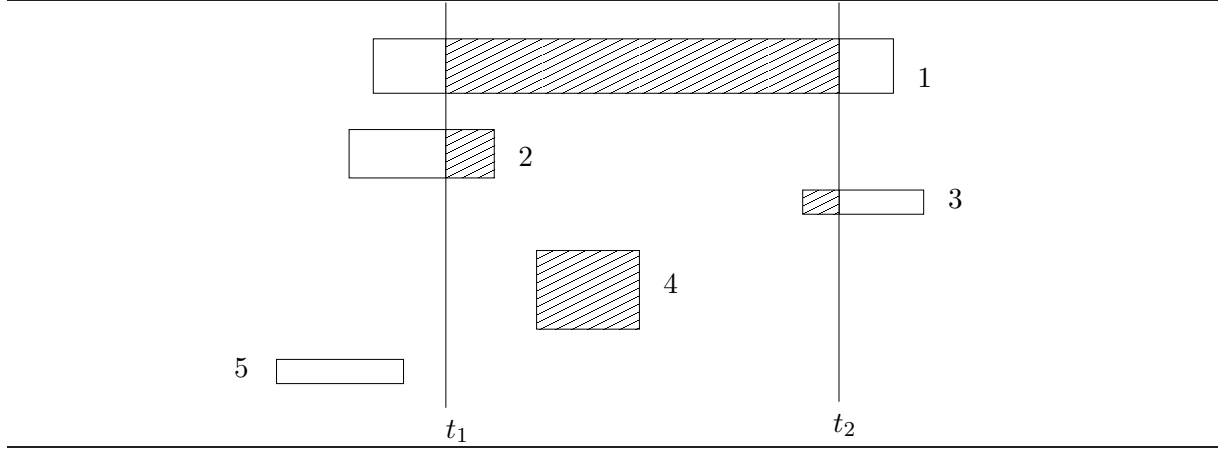


Fig. 2. Consumption of five activities

interval $[r_i, t]$ it must hold $\sum_{\tau=r_i}^t b_{i\tau} \leq b_i \cdot (t - r_i)$. We do not have such regularity constraints in the EnSP, so the partially elastic CuSP and the EnSP are not comparable in terms of complexity.

Another related extension of the CuSP has been proposed in [5], aiming at considering an activity as a sequence of consecutive subtasks such that the resource consumption of each subtask is given by a function of the subtask duration. In our case the consumption of an activity at a time period t is a decision variable.

In the discrete time-resource tradeoff model [6], the duration of each activity is not predetermined, but changes as a discrete non-increasing function of the amount of renewable resources assigned to it. However, the activities still have a rectangular shape.

4 The classical energy reasoning for the CuSP

In the energy reasoning for scheduling, the idea is to propose a smart way for simultaneously considering time and resource constraints in a unique reasoning. In that context, the *energy* is generically defined as the product of a time duration by a resource quantity. As an illustration, we can say that the problem of scheduling n activities of duration $p_i, i=1..n$ in an amount $b_i, i=1..n$ using a given resource available in a constant amount B over a time horizon Δ of duration p_Δ is isomorphic to the placement problem of n rectangles of surface area $p_i \cdot b_i, i = 1, \dots, n$, in a rectangle of surface area $B \cdot p_\Delta$.

To present the energy reasoning, one must consider a working time interval, an available energy and a total consumed energy over this interval.

Let $[t_1, t_2]$ be a reference time interval. Bounds of the interval are arbitrarily chosen but they also can be fixed to particular times. Over $[t_1, t_2]$ and for a resource of capacity B , the available energy is defined as $B \cdot (t_2 - t_1)$.

We denote by $w(i, t_1, t_2)$ the consumption of activity i (i.e. how long i uses the resource) over $[t_1, t_2]$. Two cases must be distinguished:

- (1) $[st_i, ft_i] \cap [t_1, t_2] = \emptyset \Rightarrow w(i, t_1, t_2) = 0$;
- (2) $[st_i, ft_i] \cap [t_1, t_2] \neq \emptyset \Rightarrow w(i, t_1, t_2) = b_i \cdot (\min(ft_i, t_2) - \max(st_i, t_1))$.

In Figure 2, striped areas represent the consumption of each activity from 1 to 5 between t_1 and t_2 .

One is usually especially interested in computing the lower and upper bounds of the consumption: for the consumption of activity i over interval $[t_1, t_2]$, we might derive from above equations the minimum and the maximum consumptions. The relevant notion for our purpose is obviously the minimum consumption, also called the *mandatory* (or *compulsory*) *consumption*: when trying to check whether i before j is feasible, we intend to take into account that another activity k will *necessarily* consume the resource, between st_i and ft_j , for *at least* some time T . Therefore we will not consider anymore the maximum consumption in the remainder of the paper.

The mandatory consumption of an activity i is denoted by $\underline{w}(i, t_1, t_2)$. To compute it, the activity has to be shifted to its left and right utmost positions on its time window $[r_i, d_i]$, retaining the minimum value of all intersections between such positions and the reference interval. One then gets:

- the left-shifted consumption:

$$w_L(i, t_1, t_2) = b_i \cdot \max\{0, \min(p_i, t_2 - t_1, r_i + p_i - t_1)\}$$

- the right-shifted consumption:

$$w_R(i, t_1, t_2) = b_i \cdot \max\{0, \min(p_i, t_2 - t_1, t_2 - d_i + p_i)\}$$

The mandatory consumption of activity i is then:

$$\underline{w}(i, t_1, t_2) = \min\{w_L(i, t_1, t_2), w_R(i, t_1, t_2)\} = b_i \cdot \max\{0, \min(p_i, t_2 - t_1, r_i + p_i - t_1, t_2 - d_i + p_i)\}$$

From this definition, it yields a satisfiability test (global inconsistency rule) which includes total mandatory consumption over the set of activities A :

Property 1 *CuSP feasibility test*

If $\exists [t_1, t_2]$ s.t. $\sum_{i \in A} \underline{w}(i, t_1, t_2) > B \cdot (t_2 - t_1)$, then no feasible solution exists for the CuSP.

In [1], the set of relevant intervals $[t_1, t_2]$ is characterized and an $O(n^2)$ algorithm is provided to perform the feasibility tests over all these intervals.

From this satisfiability test, we can now propose local consistency rules to derive time-window adjustments for a specified task. Let $SL(i, t_1, t_2) = B \cdot (t_2 - t_1) - \sum_{j \in A \setminus \{i\}} \underline{w}(j, t_1, t_2)$ be the maximum available energy (*i.e.* the slack) for processing i on $[t_1, t_2]$.

Property 2 *CuSP time-bound adjustments*

Release date adjustment. If an activity i verifies: $\exists [t_1, t_2]$ s.t. $w_L(i, t_1, t_2) > SL(i, t_1, t_2)$, then a valid lower bound of the completion time of i can be deduced and then impacts its release date as follows:
 $r_i \leftarrow \max\{r_i, \lceil t_2 - SL(i, t_1, t_2) / b_i \rceil\}$.

Deadline adjustment. Symetrically, if an activity i verifies: $\exists [t_1, t_2]$ s.t. $w_R(i, t_1, t_2) > SL(i, t_1, t_2)$, then a valid upper bound of the start time of i can be deduced and then impacts its deadline as follows:
 $d_i \leftarrow \min\{d_i, \lfloor t_1 + SL(i, t_1, t_2) / b_i \rfloor\}$.

In [1], an $O(n^3)$ algorithm is provided to perform all the time-bound adjustments over the relevant intervals.

5 Energy reasoning for the EnSP

A first basic feasibility rule is to check whether there is enough time in each activity time window to bring the energy it requires when the maximum power is allocated to the activity.

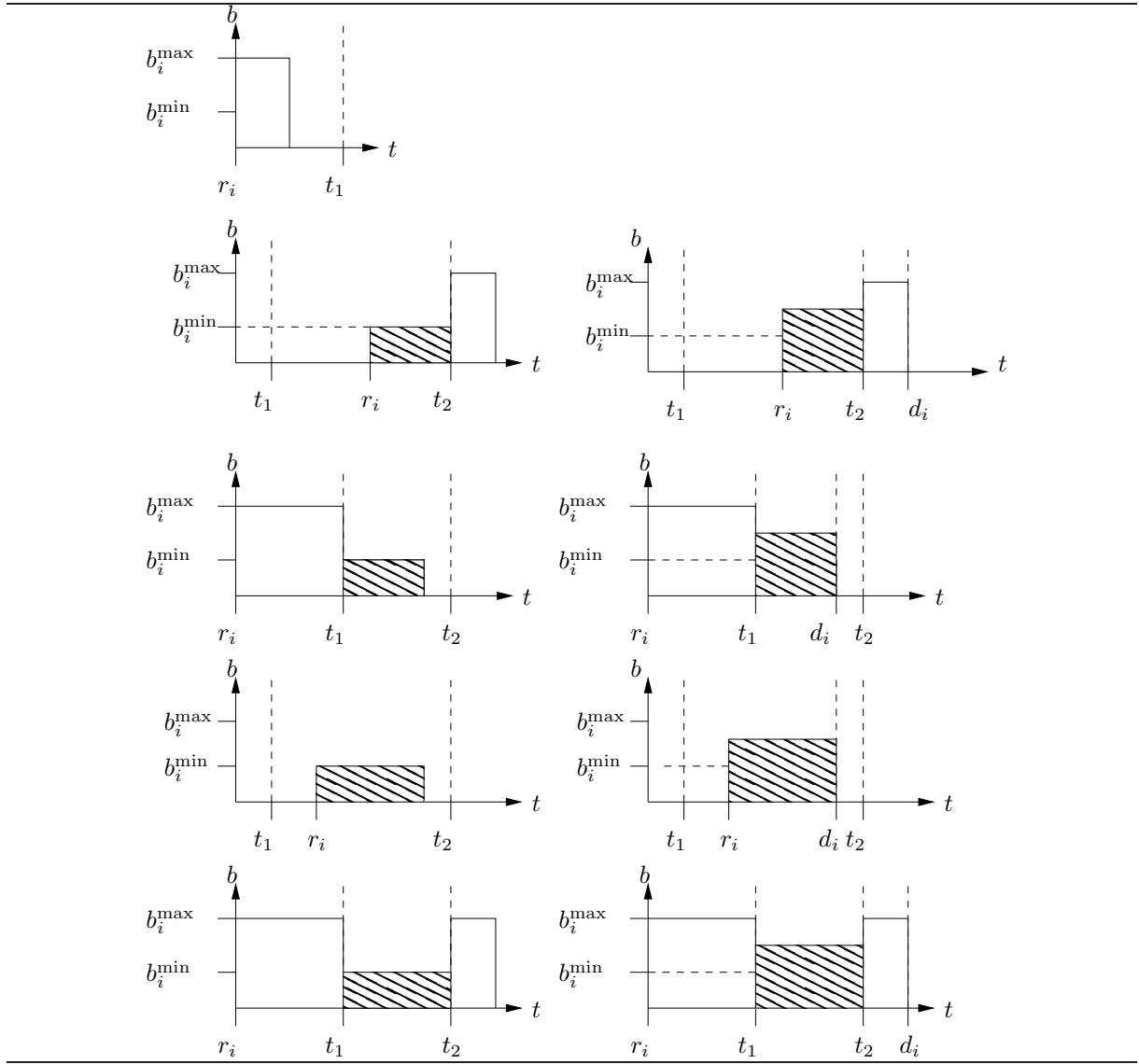
Namely, this basic feasibility test can be written as follows:

Property 3 *EnSP basic feasibility test*

If, for an activity i , $b_i^{\max} \cdot (d_i - r_i) < W_i$, the EnSP is infeasible.

In what follows we consider this condition is fulfilled for each activity. To extend the energy reasoning, the basic question to answer is, given an interval $[t_1, t_2]$, what is the mandatory consumption $\underline{w}(i, t_1, t_2)$ of each activity i ?

Obviously if $r_i \geq t_2$ or $d_i \leq t_1$, $\underline{w}(i, t_1, t_2) = 0$. Let us consider now that $r_i < t_2$ and $d_i > t_1$. As for the standard energy reasoning, the mandatory consumption of each activity i in $[t_1, t_2]$ is attained either


 Fig. 3. Different cases for left-shifted consumption $w_L(i, t_1, t_2)$

when the activity starts at its release date or when it ends by its due date. When $r_i < t_2$, the relevant cases are displayed in Figure 3.

To compute $\underline{w}(i, t_1, t_2)$ we need to compute the maximum energy $w^-(i, t_1)$ consumed by i before t_1 , as well as the maximum energy $w^+(i, t_2)$ consumed by i after t_2 . We have

$$w^-(i, t_1) = \min \{W_i, \max(0, b_i^{\max} \cdot (t_1 - r_i))\}$$

$$w^+(i, t_2) = \min \{W_i, \max(0, b_i^{\max} \cdot (d_i - t_2))\}$$

It follows that the minimal energy consumption of i inside $[t_1, t_2]$ verifies $\underline{w}(i, t_1, t_2) \geq v$ where

$$v = \min \{W_i - w_i^-(i, t_1), W_i - w_i^+(i, t_2), b_i^{\min} \cdot (t_2 - t_1)\}$$

or equivalently:

$$v = \min \{W_i - \min(W_i, b_i^{\max} \cdot \max(0, t_1 - r_i, d_i - t_2)), b_i^{\min} \cdot (t_2 - t_1)\}$$

Because of the minimal resource requirement $b_i^{\min}$, we cannot have $\underline{w}(i, t_1, t_2) < b_i^{\min}$ if $\underline{w}(i, t_1, t_2) > 0$. Furthermore the required work W_i has to be performed inside the time window $[r_i, d_i]$. So in the case

where it is necessary to consume $b_i^{\min} \cdot (t_2 - t_1)$ inside the interval, we have to check whether consuming the maximal energy outside the interval is sufficient to bring the required energy W_i . The case where $b_i^{\min} \cdot (t_2 - t_1)$ is not a sufficient energy amount because of time window tightness is displayed at the right bottom of Figure 3. Hence we set:

$$\underline{w}(i, t_1, t_2) = 0 \text{ if } v = 0 \text{ and } \underline{w}(i, t_1, t_2) = \max(b_i^{\min}, v, W_i - w_i^-(i, t_1) - w_i^+(i, t_2)) \text{ otherwise.}$$

This yields the following feasibility test.

Property 4 *EnSP feasibility test*

If $\exists [t_1, t_2]$ s.t. $\sum_{i \in A} \underline{w}(i, t_1, t_2) > B \cdot (t_2 - t_1)$, then no feasible solution exists for the EnSP.

As for the CuSP, let $SL(i, t_1, t_2) = B \cdot (t_2 - t_1) - \sum_{j \in A \setminus \{i\}} \underline{w}(j, t_1, t_2)$ denote the maximum available energy (i.e. the slack) for processing i on $[t_1, t_2]$. We obtain bound adjustments considering the two extreme cases for an activity i .

Consider $w_L(i, t_1, t_2)$ the minimal energy consumption of i in $[t_1, t_2]$ when i is left shifted (i.e. $st_i = r_i$). We have $w_L(i, t_1, t_2) \geq x$ where

$$x = \min\{W_i - w_i^-(i, t_1), b_i^{\min} \cdot (t_2 - t_1)\}$$

or equivalently :

$$x = \min\{W_i - \min(W_i, b_i^{\max} \cdot \max(0, t_1 - r_i)), b_i^{\min} \cdot (t_2 - t_1)\}$$

and, we have:

$$w_L(i, t_1, t_2) = 0 \text{ if } x = 0 \text{ and } w_L(i, t_1, t_2) = \max(b_i^{\min}, x, W_i - w_i^-(i, t_1) - w_i^+(i, t_2)) \text{ otherwise.}$$

Symmetrically, consider $w_R(i, t_1, t_2)$ the minimal energy consumption of i in $[t_1, t_2]$ when i is right shifted (i.e. $ft_i = d_i$). We have $w_R(i, t_1, t_2) \geq y$ where

$$y = \min\{W_i - w_i^+(i, t_2), b_i^{\min} \cdot (t_2 - t_1)\}$$

or equivalently :

$$y = \min\{W_i - \min(W_i, b_i^{\max} \cdot \max(0, d_i - t_2)), b_i^{\min} \cdot (t_2 - t_1)\}$$

and, we have $w_R(i, t_1, t_2) = 0$ if $y = 0$ and $w_R(i, t_1, t_2) = \max(b_i^{\min}, y, W_i - w_i^-(i, t_1) - w_i^+(i, t_2))$ otherwise.

We obtain the following time-bound adjustments:

Property 5 *EnSP time-bound adjustments*

Release date adjustment. If an activity i verifies: $\exists [t_1, t_2]$ s.t. $w_L(i, t_1, t_2) > SL(i, t_1, t_2)$, then the release date can be updated as follows:

$$r_i \leftarrow \max\{r_i, \lceil t_2 - SL(i, t_1, t_2) / b_i^{\min} \rceil\}.$$

Deadline adjustment. Symetrically, if an activity i verifies: $\exists [t_1, t_2]$ s.t. $w_R(i, t_1, t_2) > SL(i, t_1, t_2)$, then the deadline can be updated as follows:

$$d_i \leftarrow \min\{d_i, \lfloor t_1 + SL(i, t_1, t_2) / b_i^{\min} \rfloor\}.$$

As the EnSP admits the CuSP as special case, it is a priori difficult to enumerate the intervals to be considered. Indeed, from [1] we know that a part of the relevant intervals for the CuSP is such that $t_1 = r_j + p_j$ and/or $t_2 = d_j - p_j$ for some activity j . For the EnSP, except when $b_j^{\min} = b_j^{\max}$ (which

corresponds to the CuSP case), we have not a fixed activity duration but a set of possible durations from $\lceil W_j/b_j^{\max} \rceil$ to $\lceil W_j/b_j^{\min} \rceil$. For the sake of simplicity we restrict the considered intervals to $O_1 \times O_2$ where $O_1 = \{r_j | j \in A\}$ and $O_2 = \{d_j | j \in A\}$.

We can illustrate the adjustments performed on the Figure 1 example. Consider interval $[t_1, t_2]$ with $t_1 = 2$ and $t_2 = 5$. We have $\underline{w}(1, t_1, t_2) = w_L(1, t_1, t_2) = 2$, $\underline{w}(2, t_1, t_2) = w_R(3, t_1, t_2) = 7$ and $\underline{w}(3, t_1, t_2) = w_L(3, t_1, t_2) = w_R(3, t_1, t_2) = 6$. Note the configuration displayed in Figure 1 actually corresponds to the minimal consumption of the three activities in $[t_1, t_2]$. Consider the case where activity 1 is right shifted. We have $w_R(1, t_1, t_2) = 7$ (same configuration as the one displayed for activity 2). Since $\underline{w}(3, t_1, t_2) + \underline{w}(2, t_1, t_2) = 13$ the slack for activity 1 in $[t_1, t_2]$ is $SL(1, t_1, t_2) = 15 - 13 = 2$. Since $w_R(1, t_1, t_2) > SL(1, t_1, t_2)$, the deadline of activity 1 can be updated according to property 5 by setting $d_1 \leftarrow t_1 + 2/1 = 4$.

6 Dominance rules for the EnSP

The following dominance rules can be established for the EnSP.

Property 6 *Active schedules*

Active schedules are dominant for the EnSP.

Consider a solution S to the EnSP such that there is an activity i starting at time st_i and a time period $t < st_i$ such that there is a feasible solution S' setting $st_i = t$ without changing the schedule of other activities. The search space can be obviously reduced to the set of solutions for which no such property holds.

Property 7 *Power change*

Schedules for which, for any activity i , changes in the allocated power only occur on activity release dates or completion times, are dominant.

Consider a feasible solution S to an EnSP instance such that a power change occurs at a time t which does not verify the above property and such that t is maximal. Simple arguments can show that the power change can be postponed while keeping the solution feasible, which contradicts the maximality of t .

7 Branching scheme

A simple branching scheme based on time incrementation can be derived from the dominance rules presented in Section 6. Each node corresponds to a decision time point initially set to $t = \min_{i \in A} r_i$. For each activity the required energy W_i is progressively decreased and all activities are scheduled when $W_i = 0$ for all activities. At each node, associated with a decision time t , activities are partitioned into the following subsets. The *started* activities are such that the decision to start the activity has been taken at some ancestor node (at a time point $t' < t$) but no decision has been taken yet for the current decision point and $W_i > 0$. The *completed* activities are such that $ft_i \leq t$ and $W_i = 0$. The *available* activities are such that $r_i \leq t$ but no start decision has been taken yet for these activities. The *processed* activities are such that the decision to process the activity at time t with some resource amount b has already been taken and $W_i > 0$. The *unavailable* activities verify $r_i > t$ and $W_i > 0$. The *postponed* activities are those selected for being scheduled later (see branching scheme below).

At each node an activity either *started* or *available* is selected for being included in the *processed* set (or in the *postponed* set for the available activities). The activity i^* with the smallest due date is selected first and, in case of tie, the activity with the most remaining work (W_{i^*}) is selected. Let Q and E denote the set of *started* and *processed* activities, respectively. If $i^* \in Q$, $\bar{b}_{i^*} = B - \sum_{j \in Q \setminus \{i^*\}} b_j^{\min} - \sum_{j \in E} b_{jt}$ denotes the available power for i at time t . If $i^* \notin Q$, the available power for i^* is $\bar{b}_{i^*} = B - \sum_{j \in Q} b_j^{\min} - \sum_{j \in E} b_{jt}$.

If $\bar{b}_{i^*} > b_{i^*}^{\min}$, a part of i can be scheduled at time t . A child node is generated for $b \in [b_{i^*}^{\min}, \min(\bar{b}_{i^*}, b_{i^*}^{\max})]$ corresponding to an allocation of power $b_{i^*t} = b$ to i^* at time t . An additional child node, only for *available*

activities, corresponds to postponing activity i^* to a decision point $t' > t$ such that t' is either equal to the minimum between the smallest possible completion time of an activity of E and the smallest release date of *unavailable* activities, strictly greater than t . This time point is unknown at this step since set E is under construction, so the activities are just marked as *postponed* without any other update.

If no activity can be selected for being scheduled at t , we have different reasons. If all activities are in the *completed* set, the search succeeds. If all activities are either *processed*, *postponed*, *unavailable* or *available* but without enough resource capacity, the search must continue from the next decision time point set to the smallest release date or completion time of *processed* activities greater than t . At this time we may check whether the new decision point is still compatible with the due date of the *available* activities and if the schedule is active. Actually we just check whether the schedule is still semi-active by verifying that no *postponed* activity could actually start at t . If one due date cannot be satisfied or if the schedule is no more semi-active, a failure occurs and the node is pruned. Otherwise decision time point is updated. The processed activities are transferred either to the *completed* set or to the *started* set. The *postponed* activities are moved to the *available* set. The *unavailable* activities such that $r_i \leq t$ are moved to the *available* set. The activity selection process starts again and the process is iterated until an activity is selected for being processed or a failure occurs.

We illustrate the branching process on the Figure 1 problem instance. For the root node where $t = 0$, activity 1 is in the *available* set while activities 2 and 3 are in the *unavailable* set. We branch to the second node by selecting activity 1 for being scheduled at maximal power. Activity 1 is included in the *processed* set. At time $t = 0$ no other activity is available. Time t is set of the next decision point $t = 2$. Activities 2 and 3 are included in the *available* set and activity 1 is transferred to the *started* set. The third node selects activity 3 with power $b = 2$ as the activity with the smallest due date for being inserted in the *processed* set. Activities 1 and 2 can both be processed at time 2 and have the same due date but activity 2 has the most remaining work. So the fourth node selects activity 2 for being processed at time $t = 2$ with the maximal available power, taking account of processed and started activities, equal to $b = 2$. For the fifth node, activity 1 can now be processed at time $t = 2$ with its minimal power $b = 1$. No activity is available anymore at time $t = 2$, so we proceed to the next time point corresponding with the completion time of activity 1 at time $t = 4$ and activities 2 and 3 are now both in the *started* set while 1 is put into the *completed* set. For the seventh node, activity 3 is still selected with power $b = 2$ as it has the smallest due date. Then, the eighth node selects activity 2 with the maximal available power $b = 3$. Since all activities are in the *processed* set, the time point is increased to the completion time $t = 5$ of activity 3 and activity 2 is included in the *started* set. For the ninth node, activity 2 is selected with the maximal power $b = 5$ and completed at time $t = 6$. For this example no backtracking has been necessary.

8 Computational experiments

In this section we illustrate on a few randomly generated problem instances the interest of the proposed energy reasoning. In Table 1, we provide the results of two tree search methods, the first one with energy reasoning feasibility tests and time-bound adjustments applied at each node, and the second one without these tests and adjustments. Five instance EnSP1, ..., EnSP5 have been selected as illustrative ones. The first four instances have 25 activities while the last one comprises 30 activities. The solution found, the CPU time in seconds, and the number of nodes in the search tree are provided for each pair instance / method. The available power has been set to $B = 10$. The energy requirements and time windows have been randomly generated to obtain infeasible and feasible instances.

Table 1
Compared results on EnSP instances

Instance	n	B	results with energy reasoning			results without energy reasoning		
			Solution	CPU (s)	#Nodes	Solution	CPU (s)	#Nodes
EnSP1	25	10	Infeasible	0	1	NA	NA	$> 10^6$
EnSP2	25	10	Feasible	6	101970	Feasible	17	464933
EnSP3	25	10	Feasible	13	212382	Feasible	16	443491
EnSP4	25	10	Feasible	22	368264	Feasible	31	811607
EnSP5	30	10	Infeasible	27(2)	69350(69350)	Infeasible	4	173819

For the 25 activity instances, the results show the superiority of the approach incorporating energy reasoning, both for number of nodes and CPU time. For infeasible instance EnSP1, the energy reasoning proves infeasibility at the root node, while the branching scheme cannot reach this result after 10^6 nodes (“NA” stands for ‘Not Available’). However, for infeasible instance EnSP5, the branching scheme incorporating energy reasoning obtains a significantly lower number of nodes but the required CPU time is much more important. This indicates that despite the quality of the adjustments, the algorithm complexity ($O(n^3)$) is too big. Hence, in this case we may use energy reasoning only for a subset of nodes. Indeed, if we use it only at the root node (results displayed in parenthesis in Table 1), we still obtain 69350 nodes with a CPU time of 2 seconds, so incorporating energy reasoning becomes profitable again.

9 Conclusion

We presented the energy scheduling problem, an extension of the cumulative scheduling problem to represent energy requirements of activities. We showed this model is well-adapted to a parallel machine scheduling industrial context with electric power limitations. We proposed an extension of the standard energy reasoning scheme that was not covered by previous works on this subject. Finally we draw the scheme of a tree search method based on dominance rules. Computational experiments illustrate the quality of energy reasoning adjustments and the trade-off that has to be found for their efficient use inside tree search.

References

- [1] P. Baptiste, C. Le Pape, and W. Nuijten. Satisfiability tests and time-bound adjustments for cumulative scheduling problems. *Annals of Operations Research*, 92:305–333, 1999.
- [2] M. Drozdowski. Scheduling multiprocessor tasks – an overview. *European Journal of Operational Research*, 94:215–230, 2004.
- [3] J. Erschler and P. Lopez. Energy-based approach for task scheduling under time and resources constraints. In *2nd International Workshop on Project Management and Scheduling*, pages 115–121, Compiègne, France, 1990.
- [4] A. Haït, C. Artigues, M. Trépanier, and P. Baptiste. Ordonnancement sous contraintes d’énergie et de ressources humaines. In *11ème Congrès de la Société Française de Génie des Procédés*, Saint-Etienne, France, 2007.
- [5] E. Poder, N. Beldiceanu, and E. Sanlaville. Computing a lower approximation of the compulsory part of a task with varying duration and varying resource consumption. *European Journal of Operational Research*, 153(1):239–254, 2004.
- [6] M. Ranjbar, B. De Reyck, and F. Kianfar. A hybrid scatter search for the discrete time/resource trade-off problem in project scheduling. *European Journal of Operational Research*, 193(1):35–48, 2009.
- [7] M. Trépanier, P. Baptiste, A. Haït, and I.D. Arciniegas Alvarez. Modélisation des impacts du délestage énergétique sur la production. In *6ème Congrès International de Génie Industriel*, Besançon, France, 2005.