Polyhedral results and valid inequalities for the Continuous Energy-Constrained Scheduling Problem

Margaux Nattaf[a], Markó Horváth[b], Tamás Kis[b], Christian Artigues[a,*], Pierre Lopez[a]

[a] LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France
[b] Institute for Computer Science and Control, Hungarian Academy of Sciences, Hungary

Abstract

This paper addresses a scheduling problem involving a continuously-divisible and cumulative resource with limited capacity. During its processing, each task requests a part of this resource, which lies between a minimum and a maximum requirement. A task is finished when a certain amount of energy is received by it within its time window. This energy is received via the resource and an amount of resource is converted into an amount of energy with an increasing and pseudo-linear efficiency function. The goal is to minimize the resource consumption. The paper focuses on an event-based mixed integer linear program, providing several valid inequalities, which are used to improve the performance of the model. Furthermore, we give a minimal description of the polytope of all feasible assignments to the on/off binary variable for a single activity along with a dedicated separation algorithm. Computational experiments are reported in order to show the effectiveness of the results.

Keywords: continuous scheduling, continuous resources, linear efficiency functions, mixed-integer programming, valid inequalities, polyhedral combinatorics

1. Introduction

Most of the scheduling problems dealing with resource constraints assume a fixed duration and do not allow the resource usage to vary over time. However, several extensions of existing problems, such as the resource-constrained project scheduling problem or the cumulative scheduling problem, have been developed to tackle at least one

*Corresponding author
Email address: artigues@laas.fr (Christian Artigues)

of these issues. Among them, there are the multi-mode resource-constrained project scheduling problem [18] in which task duration and consumption depend on their execution mode. Another example of such extension is the project scheduling problem with variable-intensity activities [9] where the resource used by an activity can change during its execution and the duration of the activity depends on this consumption. The same idea is developed in [6] with the problem of scheduling activities using a work-content resource, in [4] in the context of malleable task scheduling and in [10] where a scheduling problem with flexible resource profiles and continuous time is considered. In this paper, we study a problem called the continuous energy-constrained scheduling problem (CECSP), a generalization of the cumulative scheduling problem that no longer assumes fixed duration and resource requirement. The principal difference between the CECSP and the previously cited problem are discussed after the problem description.

In the CECSP, we are given as input a set of non-preemptive tasks $\mathcal{A} = \{1, \ldots, n\}$ and a continuously-divisible cumulative resource of capacity B. For each task i, a release date r_i and a deadline d_i define an interval in which the task must be executed. At each time t during its execution, each task requests a quantity of resource $b_i(t)$ that has to be determined. This resource usage has to lie between a minimum requirement, b_i^{min}, and a maximum requirement, b_i^{max}.

The particularity of the CECSP is that a task no longer has a fixed duration but instead an energy requirement W_i needs to be fulfilled before the task deadline. This energy is computed from the task resource usage, using an efficiency function f_i[1]. We assume these functions to be increasing and pseudo-linear. An efficiency function f_i can be defined as follows:

$$f_i(b) = \begin{cases} 0 & \text{if } b = 0 \\ a_i * b + c_i & \text{if } b \in [b_i^{min}, b_i^{max}] \setminus \{0\} \end{cases}$$

with $a_i > 0$ and $-a_i * b_i^{min} \leq c_i$ to ensure that $f_i(b) \geq 0,\ \forall b \in [b_i^{min}, b_i^{max}]$.

Therefore, to solve the CECSP, we have to find, for each task $i \in \mathcal{A}$, its start time st_i, its end time et_i and its resource allocation function $b_i(t)$, $\forall t \in \mathcal{T} = [\min_{i \in \mathcal{A}} r_i, \max_{i \in \mathcal{A}} d_i]$. These quantities have to satisfy the following constraints:

$$r_i \leq st_i < et_i \leq d_i \qquad \forall i \in \mathcal{A} \qquad (1)$$

[1]Some authors call this function the power processing rate function [3, 8, 18].

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

$$b_i(t) = 0 \qquad \forall i \in \mathcal{A},\ \forall t \notin [st_i, et_i[\tag{3}$$

$$\int_{st_i}^{et_i} f_i(b_i(t))dt = W_i \qquad \forall i \in \mathcal{A} \tag{4}$$

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

The objective we are interested in is the minimization of the total resource consumption which can be expressed as:

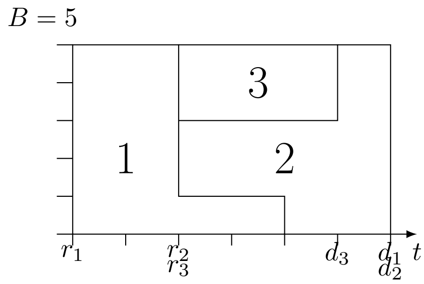
$$miniminze \sum_{i \in \mathcal{A}} \int_{st_i}^{et_i} b_i(t)dt$$

In [14], the authors show that the problem of finding a feasible solution is NP-complete.

Example 1 ([12])**.** Consider an instance with $n = 3$ and $B = 5$. The other data are displayed in Table 1a, and a feasible solution is depicted in Fig. 1b.

i	r_i	d_i	W_i	b_i^{min}	b_i^{max}	$f_i(b)$
1	0	6	28	1	5	$2b+1$
2	2	6	32	2	5	$b+5$
3	2	5	6	2	2	b

(a) an instance of CECSP



(b) the corresponding solution

Figure 1: An example of an instance and the corresponding solution for the CECSP

This solution is feasible since each task lies in its time window, all the constraints of maximum and minimum requirements are satisfied and the total resource usage at each time does not exceed the availability of the resource. Furthermore, the required energy is received by each task. For example, the energy received by Task 1 is $(2*5+1)+(2*5+1)+(2*1+1)+(2*1+1) = 11+11+3+3 = 28$, while its total resource consumption is equal to 12.

The CECSP comes from an industrial problem occurring in the context of energy-consuming production scheduling problem. In [1], a foundry application is presented where a metal is melted in induction furnaces. The electrical power of the furnaces, which can be adjusted at any time to avoid exceeding a maximum prescribed power limit, can be seen as a continuous function of time to be determined. However, the function must lie within a limit; thus, a minimum and a maximum power level must be satisfied for the melting operation. Additionally, the melting operation can be stopped once the necessary energy has been received, depending of the selected power function, so the duration of this operation is not known in advance. Moreover, if we increase the power of an electrical furnace to accelerate melting operations, the energy received by the operation is not identical to the electrical energy consumed but is linked to it via a function. Efficiency functions should then be considered for the furnaces. However, the paper did not consider them. The goal is to minimize the total electricity consumption.

Still without considering efficiency functions but in a continuous time setting, Artigues and Lopez [2] propose a constraint satisfiability test based on the energetic reasoning. Recently, Nattaf *et al.* [12] propose mixed integer linear programs, a constraint propagation algorithm and a hybrid branch-and-bound method to solve the problem with linear efficiency functions and considering continuous time. These methods rely on a theorem by Nattaf *et al.* stating that in the case of linear efficiency functions f_i, any solution can be transformed into a solution having the property that each resource allocation function $b_i(t)$ is piecewise constant and its breakpoints can be limited to the start and end times of tasks. Hence, it is sufficient to consider only solutions which satisfy this property.

In this paper, we decide to consider pseudo-linear efficiency functions for several reasons. First, because, even if the efficiency functions are not always linear in real-life application, we can always use a linear approximation of the function. Furthermore, the authors of [15] consider the same problem with concave piecewise linear efficiency functions and they show that the model presented in Section 2 can easily be adapted to this case by changing only one constraint set. Since this set is not involved in any of our results, this work is also valid for the case of concave piecewise linear efficiency functions. However, due to the difficulty of the problem and since the goal of the paper is to show the interest of the inequalities, we choose to only consider pseudo-linear efficiency function in this paper.

Different elements of the problem addressed in this paper have been studied by several authors. However, most of the papers consider generally only some part of the problem and/or assume a discrete time setting. Our problem belongs to the category of problems where the resource usage may vary continuously and such that the amount of resource required by a task may vary over time. Węglarz *et al.* [18] call this model the processing rate versus resource amount model. Providing a general framework for solving mixed discrete/continuous problems with concave processing rate functions, Józefowska *et al.* [8] show that once the sequence of sets of tasks to be scheduled in parallel is determined, the continuous resource allocation can be accomplished by a convex non-linear optimization problem. In the literature on parallel processor scheduling, the malleable task model also considers the possibility of changing the number of processors assigned to a task over time, with non-linear processing rate functions but these problems are generally preemptive [4]. A related work has also been carried out by Kis [9] for a discretized time problem with variable intensity tasks, who established polyhedral results and proposed a branch-and-cut procedure. Besides time discretization, the problem does not involve efficiency functions.

To complete the presentation of the relevant literature on the subject, the project scheduling problem with work-content constraint is also of interest. The problem considered in [6] involves among other constraints a minimum amount and a maximum amount of resource usage once the task is started. The resource requirement takes discrete values and the model does not involve efficiency functions. There is a single work-content resource and other resources (called dependent resources in [11]) such that the resource requirement of a task on any dependent resource at a given time is a non-decreasing function of the resource requirement on the work-content resource. Naber and Kolisch [11] present several discrete time MILP models for such problems, considering linear "dependency" functions. A continuous-time formulation based on events is proposed in [10]. The formulation involves events corresponding to task start times, end times and resource usage changes. In the following section, we present some properties of this problem that show among others that the "change" event is not necessary in our case.

This paper focuses on the on/off model from [12], providing several valid inequalities, which are used to improve the performance of the model in Section 2. Furthermore, a special set of inequalities is described in Section 3. These inequalities are used to give a minimal description of the polytope of all feasible assignments to the on/off variable z_{ie} for a single activity. This section also describes a special separation algorithm for these inequalities. Finally, computational results are presented in Section 4.

2. On/Off MILP formulation and valid inequalities

In the on/off formulation of [12], an event corresponds either to a task start or a task end time. These events are represented by a set of continuous variables t_e, which provides the occurence time of event e. The index set of these events is represented by $\mathcal{E} = \{1, \ldots, 2n\}$. The authors use a binary variable z_{ie} to assign the different event dates to the start and end time of the tasks. Indeed, z_{ie} is equal to 1 if and only if task i is in process during interval $[t_e, t_{e+1}[$. Note that, due to the definition of z_{ie}, we only need to define $z_{ie}, \forall i \in \mathcal{A}, \forall e \in \mathcal{E} \setminus \{2n\}$ and set $z_{i,2n}$ to 0, $\forall i \in \mathcal{A}$ (cf. constraints (22)). However, for ease of notation, we consider variables $z_{ie}, \forall i \in \mathcal{A}, \forall e \in \mathcal{E}$. In addition, two continuous variables b_{ie} and w_{ie} are also defined. These variables stand for the quantity of resource used by task i and for the energy received by i between events t_e and t_{e+1}, respectively.

Although the model described below is very similar to the one of [12], several small differences exist. Those differences are the strengthening of constraints (13), (14) and (16) and the addition of constraints (22).

$$\min \sum_{i \in A} \sum_{e \in \mathcal{E} \setminus \{2n\}} b_{ie} \tag{6}$$

$$t_e \leq t_{e+1} \qquad \forall e \in \mathcal{E} \setminus \{2n\} \tag{7}$$

$$r_i z_{ie} \leq t_e \qquad \forall (i,e) \in \mathcal{A} \times \mathcal{E} \tag{8}$$

$$t_e \leq s_i^{max}(z_{ie} - z_{i,e-1}) + (1 - (z_{ie} - z_{i,e-1}))|\mathcal{T}| \qquad \forall (i,e) \in \mathcal{A} \times \mathcal{E} \setminus \{1\} \tag{9}$$

$$e_i^{min}(z_{i,e-1} - z_{ie}) \leq t_e \qquad \forall (i,e) \in \mathcal{A} \times \mathcal{E} \setminus \{1\} \tag{10}$$

$$t_e \leq d_i(z_{i,e-1} - z_{ie}) + (1 - (z_{i,e-1} - z_{ie}))|\mathcal{T}| \qquad \forall (i,e) \in \mathcal{A} \times \mathcal{E} \setminus \{1\} \tag{11}$$

$$\sum_{e \in \mathcal{E} \setminus \{2n\}} z_{ie} \geq 1 \qquad \forall i \in \mathcal{A} \tag{12}$$

$$\sum_{e'=1}^{e-1} z_{ie'} \leq (e-1)(1-(z_{ie}-z_{i,e-1})) \qquad \forall (i,e) \in \mathcal{A} \times \mathcal{E} \setminus \{1,2n\} \tag{13}$$

$$\sum_{e'=e}^{2n-1} z_{ie'} \leq (2n-e)(1+(z_{ie}-z_{i,e-1})) \qquad \forall (i,e) \in \mathcal{A} \times \mathcal{E} \setminus \{1,2n\} \tag{14}$$

$$\sum_{i \in \mathcal{A}} b_{ie} \leq B(t_{e+1}-t_e) \qquad \forall e \in \mathcal{E} \setminus \{2n\} \tag{15}$$

$$b_{ie} \geq b_i^{min}(t_{e+1}-t_e) - (b_i^{min}(d_i-r_i)(1-z_{ie})) \qquad \forall (i,e) \in \mathcal{A} \times \mathcal{E} \setminus \{2n\} \tag{16}$$

$$b_{ie} \leq b_i^{max}(t_{e+1}-t_e) \qquad \forall (i,e) \in \mathcal{A} \times \mathcal{E} \setminus \{2n\} \tag{17}$$

$$z_{ie}(b_i^{max}(d_i-r_i)) \geq b_{ie} \qquad \forall (i,e) \in \mathcal{A} \times \mathcal{E} \setminus \{2n\} \tag{18}$$

$$\sum_{e \in \mathcal{E} \setminus \{2n\}} w_{ie} = W_i \qquad \forall i \in \mathcal{A} \tag{19}$$

$$w_{ie} \leq a_i b_{ie} + c_i(t_{e+1}-t_e) \qquad \forall (i,e) \in \mathcal{A} \times \mathcal{E} \setminus \{2n\} \tag{20}$$

$$w_{ie} \leq W_i z_{ie} \qquad \forall (i,e) \in \mathcal{A} \times \mathcal{E} \setminus \{2n\} \tag{21}$$

$$z_{i,2n} = 0 \qquad \forall i \in \mathcal{A} \tag{22}$$

$$t_e \geq 0 \qquad \forall e \in \mathcal{E} \tag{23}$$

$$b_{ie} \geq 0 \qquad \forall (i,e) \in \mathcal{A} \times \mathcal{E} \setminus \{2n\} \tag{24}$$

$$w_{ie} \geq 0 \qquad \forall (i,e) \in \mathcal{A} \times \mathcal{E} \setminus \{2n\} \tag{25}$$

$$z_{ie} \in \{0,1\} \qquad \forall (i,e) \in \mathcal{A} \times \mathcal{E} \tag{26}$$

with $s_i^{max} = d_i - \frac{W_i}{f_i(b_i^{max})}$ (respectively $e_i^{min} = r_i + \frac{W_i}{f_i(b_i^{max})}$) the latest start time (resp. earliest end time) of task i.

The objective of minimizing the total resource consumption is described by (6). Constraint (7) arbitrarily orders the events. Inequalities (8)–(11) model the time window constraints. Constraint (12) stipulates that each task has to be scheduled once while constraints (13) and (14) make sure a task is not preempted during its execution. Finally, inequalities (15)–(18) model the resource constraints while inequalities (19)–(21) represent resource conversion and energy requirement constraints.

In the remainder of this section, we discuss various ways to strengthen the above formulation.

2.1. Maximum interval between two events

Here, we describe the first set of inequalities we derive for the CECSP. In the following, we suppose an event corresponds to one and only one task start/end time. Such solution always exists since there is $2 \times \#tasks$ events.

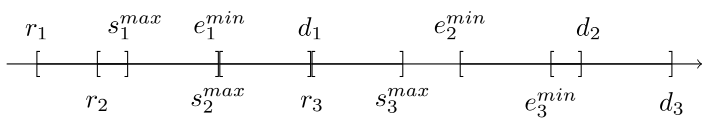
The inequalities defined in this section aim at giving an upper bound for the value of $t_{e+1} - t_e$, $\forall e \in \mathcal{E}$. To do so, we study the time window of each task start and end time, $[r_i, s_i^{max}]$, and $[e_i^{min}, d_i]$, respectively, for each task i. Let $\mathcal{S}$ be the set of all these $2n$ intervals. The main idea relies on the fact that an event must occur in each of these time windows. Therefore, we know there are at least two consecutive events in the union of two consecutive time windows.

We start by defining an ordering relation on the intervals in $\mathcal{S}$: $[L_i, R_i] \leq [L_j, R_j] \Leftrightarrow L_i < L_j \vee (L_i = L_j \wedge R_i \leq R_j)$. Let $I_1 \leq \cdots \leq I_{2n}$ be a total ordering of the intervals in $\mathcal{S}$ according to the relation just defined. Then we have:

$$t_{e+1} - t_e \leq M_e := |I_e \cup I_{e+1}| = \max\{R(e+1), R(e)\} - L(e), \quad \forall e \in \mathcal{E} \setminus \{2n\}, \qquad (27)$$

where $I_e = [L(e), R(e)]$ is the eth interval in the sequence of intervals in $\mathcal{S}$.

Example 2. Consider the following intervals:



An ordering of these intervals is: $[r_1, s_1^{max}] \leq [r_2, s_2^{max}] \leq [e_1^{min}, d_1] \leq [r_3, s_3^{max}] \leq [e_2^{min}, d_2] \leq [e_3^{min}, d_3]$.

And then, we have the following constraints:

- $t_2 - t_1 \leq s_2^{max} - r_1$
- $t_3 - t_2 \leq d_1 - r_2$
- $t_4 - t_3 \leq s_3^{max} - e_1^{min}$
- $t_5 - t_4 \leq d_2 - r_3$
- $t_6 - t_5 \leq d_3 - e_2^{min}$

We can add these constraints to the model and/or use these upper bounds in constraint (16), replacing $b_i^{min}(d_i - r_i)$ by $b_i^{min}|I_e \cup I_{e+1}|$, $\forall (i,e) \in \mathcal{A} \times \mathcal{E}$.

2.2. Maximum time of an event

A similar idea as the previous one is to use the intervals in $\mathcal{S}$ to order the event times and to compute upper bounds for these dates. To do so, we sort the right end of each interval, i.e., the values s_i^{max} and d_i, $\forall i \in \mathcal{A}$, in increasing order. Then, since an event must occur in each time window, i.e. before its right end, we have an upper bound for each event.

Indeed, let $R(1) \leq R(2) \leq \cdots \leq R(2n)$ be a total order of the right end points of the intervals in $\mathcal{S}$. Then, we have the following:

$$t_e \leq R(e), \quad \forall e \in \mathcal{E} \tag{28}$$

Example 3. Consider the intervals given in Example 2. Then, we have the following constraints:

- $t_1 \leq s_1^{max}$
- $t_2 \leq s_2^{max}$
- $t_3 \leq d_1$
- $t_4 \leq s_3^{max}$
- $t_5 \leq d_2$
- $t_6 \leq d_3$

Like the previous inequalities, we can use these bounds as additional constraints for the model.

2.3. Strengthening constraints (9) *and* (11)

Consider first (9).

$$t_e \leq s_i^{\max}(z_{i,e} - z_{i,e-1}) + M(1 - (z_{i,e} - z_{i,e-1})) \quad \forall\, i = 1, \dots, n,\ e = 2, \dots, 2n,$$

where we replaced $|\mathcal{T}|$ by constant M to be determined next. For given i and e, M and $2M - s_i^{\max}$ must be valid upper bounds on t_e. Let $d_{\max} := \max_i d_i$, and $R(e)$ an upper bound on t_e as described above. We do not lose optimal solutions, if $M \geq \min\{R(e), d_{\max}\}$ and $2M - s_i^{\max} \geq \min\{R(e), d_{\max}\}$ hold. Thus

$$M := \max\left\{\min\left\{R(e), d_{\max}\right\}, \min\left\{\frac{R(e) + s_i^{\max}}{2}, \frac{d_{\max} + s_i^{\max}}{2}\right\}\right\}$$

is a good choice.

Now let us turn to (11):

$$t_e \leq d_i(z_{i,e-1} - z_{i,e}) + M(1 - (z_{i,e-1} - z_{i,e})) \quad \forall\, i = 1, \dots, n,\ e = 2, \dots, 2n,$$

where again, we replaced $|\mathcal{T}|$ by constant M. For given i and e, M and $2M - d_i$ must be valid upper bounds on t_e. We do not lose optimal solutions, if $M \geq \min\{R(e), d_{\max}\}$ and $2M - d_i \geq \min\{R(e), d_{\max}\}$ hold. Thus

$$M := \max\left\{\min\left\{R(e), d_{\max}\right\}, \min\left\{\frac{R(e) + d_i}{2}, \frac{d_{\max} + d_i}{2}\right\}\right\}$$

will do.

2.4. *Strengthening constraints* (19)-(21)

We will strengthen (19)-(21) by eliminating the variables w_{ie}. Let $w'_{ie} = w_{ie} - a_i b_{ie}$, and the sign of w'_{ie} depends on c_i as we will see below. Then (19) is replaced by

$$\sum_{e=1}^{2n-1} (a_i b_{ie} + w'_{i,e}) \geq W_i, \quad \forall i \in \mathcal{A}. \tag{29}$$

If $c_i = 0$, then we fix $w'_{ie} = 0$, and (20) and (21) are not needed at all. Otherwise, (20) and (21) are replaced by

$$w'_{ie} \leq c_i(t_{e+1} - t_e), \quad \forall\ (i,e) \in \mathcal{A} \times \mathcal{E} \setminus \{2n\} \tag{30}$$

$$0 \leq w'_{ie}/c_i \leq M_e z_{ie}, \quad \forall\ (i,e) \in \mathcal{A} \times \mathcal{E} \setminus \{2n\}, \tag{31}$$

$$a_i b_{ie} + w'_{ie} \geq 0, \quad \forall\ (i,e) \in \mathcal{A} \times \mathcal{E} \setminus \{2n\} \text{ with } c_i < 0, \tag{32}$$

where M_e is an upper bound on $t_{e+1} - t_e$ as defined in eq. (27). Observe that $w'_{ie} \geq 0$ if $c_i > 0$, and $w'_{ie} \leq 0$ if $c_i < 0$.

2.5. *Valid inequalities from the knapsack problem*

Since the minimum intensity of the activities can be positive, we can consider the following knapsack type constraint for each $e \in \mathcal{E} \setminus \{2n\}$ from which one can easily derive valid inequalities:

$$\sum_{i \in \mathcal{A}^+} b_i^{min} z_{ie} \leq B, \tag{33}$$

where $\mathcal{A}^+$ is the subset of activities with positive b_i^{min} values. One may add this set of constraints to the initial formulation, and then let the solver strengthen the LP relaxation by cutting planes for these knapsack sets.

3. Polyhedral results and non-preemptive inequalities

In this section we describe some inequalities satisfied by all feasible solutions of the on/off MILP formulation presented in Section 2.

Sousa [16] proposed a pseudo-polynomial number of inequalities to describe a polyhedron whose extreme points are the so called p-connected vectors, which are the vectors of p consecutive ones and $n-p$ zeros. This gives an ideal formulation for the non-preemptive constraints for the on/off time indexed formulation. However, in our case the number of consecutive ones is unknown. To achieve this results, Sousa referred to a paper by Gröflin and Liebling [7], in which, among other results, the minimal description of the (connected) vectors representing the edge sets of subtrees (connected subgraphs) of a tree is given. In particular, they provide a minimal description of the polytope of all connected subchains of a chain. Their polyhedral descriptions are based on the concept of alternating vectors, and they are derived using deep results of Edmonds and Giles [5] and polyhedral polarity. They also devise two polynomial time algorithms to solve an optimization problem over this polyhedron for the case of trees, and also for chains. The inequalities we propose below can thus be derived from their polyhedral results, except one which excludes empty subchains. We present an alternative proof tailored to our particular case and we also propose an original polynomial time separation algorithm.

3.1. Non-preemptive inequalities

Since each activity must be processed without preemption, in any feasible schedule for each activity i, the z_{ie} satisfy

$$\sum_{e_k \in S} (-1)^k z_{i,e_k} \leq 1 \tag{34}$$

where $S = \{e_0, e_1, \ldots, e_{2\ell}\}$ is a subset of $\mathcal{E}^* := \mathcal{E} \setminus \{2n\}$ of odd cardinality such that $e_k < e_{k+1}$ for $k = 0, \ldots, 2\ell - 1$.

Note that in [17], inequalities (2.13) are a special case of our non-preemptive cuts, defined over three events only. Moreover, Proposition 2.6, which states that neither the contiguity constraints of the on/off model, nor the special non-preemptive cuts are needed to get optimal solutions for RCPSP with the makespan minimization objective, does not hold in general for the CECSP.

Consider the following polyhedron:

$$ZP_i := \{z_i \in \mathbb{R}^{\mathcal{E}^*} \mid z_{ie} \geq 0, z_{ie} \leq 1 \text{ for } e \in \mathcal{E}^*, \text{ and } z_i \text{ satisfies (12) and (34)}\}$$

On the other hand, let $ZQ_i := \text{conv}\{z_i \in \{0,1\}^{\mathcal{E}^*} \mid z_i \text{ satisfies } (12) - (14)\}$.

Theorem 1. *$ZP_i = ZQ_i$.*

Proof. We will use the Farkas lemma in order to derive a description of ZQ_i in terms of linear inequalities. The vertices of ZQ_i are precisely the $|\mathcal{E}^*|$-dimensional vectors

$$z_{i,e}^{k\ell} = \begin{cases} 1 & k \leq e \leq \ell \\ 0 & \text{otherwise} \end{cases} \qquad \forall k, \ell \in \mathcal{E}^*, k \leq \ell.$$

Consider the linear system

$$\sum_{k\leq \ell} z^{k\ell}_{i,e}\lambda_{k\ell} = \bar{z}_{i,e}, \quad e \in \mathcal{E}^* \tag{35}$$

$$\sum_{k\leq \ell} \lambda_{k\ell} = 1 \tag{36}$$

$$\lambda \geq 0 \tag{37}$$

Clearly, $\bar{z}_i \in ZQ_i$ if and only if this system admits a feasible solution. By the Farkas lemma, the system (35)–(37) admits a feasible solution if and only if for all μ satisfying the dual system

$$\sum_{e=k}^{\ell} \mu_e + \mu_0 \leq 0, \quad k \leq \ell \tag{38}$$

μ also satisfies the condition

$$\sum_{e\in\mathcal{E}^*} \mu_e \bar{z}_{i,e} + \mu_0 \leq 0. \tag{39}$$

In order to prove our theorem, it suffices to find all the extreme rays of cone (38), since they define all the linear inequalities needed to describe ZQ_i. We will show that there is a one-to-one correspondence between the extreme rays of cone (38), and the inequalities of ZP_i. In order to find all the extreme rays of the cone (38), it suffices to distinguish between 3 cases:

$\mu_0 = 1$. Then for each $e \in \mathcal{E}^*$, $\mu_e \leq -1$ follows from (38) by considering the inequalities for $k = \ell = e$. Then (39) yields

$$\sum_{e\in\mathcal{E}^*} -z_{i,e} \leq -1,$$

which, by the Farkas lemma, is a valid inequality for ZQ_i. Notice that it is equivalent to (12).

$\mu_0 = 0$. Then we still have a cone, whose extreme rays are the negative unit vectors in $\mathbb{R}^{\mathcal{E}^*}$. These extreme rays give the inequalities $-z_{i,e} \leq 0$, which are the non-negativity constraints valid for ZQ_i.

$\mu_0 = -1$. We argue that there is a one-to-one correspondence between the extreme points of the polyhedron $M \subseteq \mathbb{R}^{|\mathcal{E}^*|}$ defined by

$$\sum_{e=k}^{\ell} \mu_e \leq 1, \quad k \leq \ell \tag{40}$$

and the inequalities (34).

First we claim that the coefficient vector of the left-hand-side of each inequality in (34) is an extreme point solution of (40). Let $S = \{e_0, e_1, \ldots, e_{2\ell}\}$ be a set of events with $e_i < e_{i+1}$ for $i = 0, \ldots, 2\ell - 1$. The corresponding vector $\bar{\mu}$ is defined as

$$\bar{\mu}_e = \begin{cases} (-1)^k, & \text{if } e_k \in S \\ 0, & \text{if } e \in \mathcal{E}^* \setminus S. \end{cases}$$

We claim that $\bar{\mu}$ is an extreme point solution of (40). To prove our claim, we exhibit a subsystem L of (40) consisting of $|\mathcal{E}^*|$ linearly independent inequalities such that each inequality in L holds at equality in $\bar{\mu}$. The subsystem contains the inequalities

$$\sum_{e=k}^{e_0} \mu_e \leq 1, \quad k = 1, \ldots, e_0 - 1.$$

and

$$\sum_{e=e_{2\ell}}^{k} \mu_e \leq 1, \quad k = e_{2\ell}, \ldots, |\mathcal{E}^*|.$$

Furthermore, for each three consecutive events $e_{2k}, e_{2k+1}, e_{2k+2} \in S$, subsystem L also comprises the inequalities

$$\sum_{e=e_{2k}}^{e_{2k+2}} \mu_e \leq 1,$$
$$\sum_{e=e_{2k}}^{t} \mu_e \leq 1, \quad t = e_{2k}, \ldots, e_{2k+1} - 1$$
$$\sum_{e=t}^{e_{2k+2}} \mu_e \leq 1, \quad t = e_{2k+1} + 1, \ldots, e_{2k+2} - 1$$

It is easy to verify that subsystem L consists of $|\mathcal{E}^*|$ linearly independent inequalities, and $\bar{\mu}$ satisfies each of them at equality, which proves our claim.

Now we claim that any extreme point solution $\bar{\mu}$ of (40) is equivalent to an inequality in (34). First, notice that the constraint matrix of (40) is totally unimodular, thus any vertex of this polyhedron is an integral vector. Also observe that $\bar{\mu}_e \leq 1$ for all $e \in \mathcal{E}$, since $\mu_e \leq 1$ is an inequality of (40) for each $e \in \mathcal{E}^*$. Let k_1 be the first index such that $\bar{\mu}_e \neq 0$. We claim that $\bar{\mu}_{k_1} = 1$. Suppose not, i.e., $\bar{\mu}_{k_1} \leq -1$ (recall that the coordinates of $\bar{\mu}$ are integers). Since $\bar{\mu}$ is an extreme point of M, there must exist a subset L of $|\mathcal{E}^*|$ linearly independent inequalities from (40) that are satisfied at equality in $\bar{\mu}$. Observe that L must contain an inequality involving the variable μ_{k_1}, otherwise this variable may be made arbitrarily negative while still satisfying all inequalities in L, and thus $\bar{\mu}$ would not be an extreme point of M, a contradiction. Since $\bar{\mu}_e = 0$ for $e < k_1$, such an inequality must be of the form $\sum_{e=k_1}^{\ell_1} \mu_e \leq 1$. Since it must hold at equality in $\bar{\mu}$, and $\bar{\mu}_e \leq 1$ as we have already noticed, it follows that $\bar{\mu}$ must admit at least two coordinates q_1 and q_2 such that $k_1 < q_1 < q_2 \leq \ell_1$ with $\bar{\mu}_{q_1} = \bar{\mu}_{q_2} = 1$, and $\bar{\mu}_e = 0$ for $q_1 < e < q_2$. But then, $\bar{\mu}$ would violate the inequality $\sum_{e=q_1}^{q_2} \mu_e \leq 1$, a contradiction.

So, the first non-zero coordinate of $\bar{\mu}$ has value 1. The next nonzero coordinate, say k_2, cannot have value 1 by the previous argument. So, it must be a negative integer. If it were smaller than -1, then the sum of coordinates of $\bar{\mu}$ up to and including k_2 would be a negative integer. But then we could argue as above to show that k_2 must be involved in an inequality $\alpha \in L$, and then to reach a similar contradiction as above. Therefore, the second nonzero coordinate of $\bar{\mu}$ must be -1. Moreover, α must involve a variable μ_{k_3} of value 1 in $\bar{\mu}$, otherwise it cannot hold at equality in $\bar{\mu}$. Continuing this argument if μ still has nonzero coordinates after k_3, we recognize that μ has the pattern of $1/-1$ as in the inequalities (34).

□

Next, we show that each inequality in the description of ZP_i is facet defining. We begin by a simple observation.

Observation 1. The dimension of ZQ_i is $2n-1$.

Proof. It suffices to provide a point which satisfies all the defining inequalities of ZQ_i by strict inequality. Let $d = 2n-1$. Such a point is, e.g., $(\frac{1}{d}+\varepsilon, \frac{1}{d}+\varepsilon, \dots, \frac{1}{d}+\varepsilon) \in ZQ_i$ for $\varepsilon > 0$ sufficiently small. □

In order to show that each inequality in the definition of ZP_i is facet defining, it suffices to provide a point in the relative interior of the corresponding face, which satisfies all other inequalities strictly.

Theorem 2. *Each inequality in the definition of ZP_i is facet defining.*

Proof. It is easy to see that dropping any of the inequalities $z_{ie} \geq 0$ or $z_{ie} \leq 1$ for $e \in \mathcal{E}^*$ from the definition of ZP_i yields a polyhedron strictly containing ZP_i, so all these inequalities define facets. Now consider (12). Let $d := 2n-1$. The point $v := (\frac{1}{d}, \frac{1}{d}, \dots, \frac{1}{d})$ satisfies it with equality, while all other inequalities are satisfied by v with strict inequality. Hence, removing (12) from the definition of ZP_i would yield a larger polytope. Finally, consider any inequality from (34) with $S = \{e_0, e_1, \dots, e_{2\ell}\}$. We assume that $\ell \geq 1$, since if S has a unique element, then the corresponding inequality is $z_{ie} \leq 1$, which has been already considered above. Define the vector v as follows.

$$v_e = \begin{cases} \frac{1}{\ell+1} + \varepsilon, & e = e_{2k} \text{ for some } k \in \{0, \dots, \ell\}, \\ \frac{\ell+1}{\ell}\varepsilon, & e = e_{2k+1} \text{ for some } k \in \{0, \dots, \ell-1\}, \\ 3\varepsilon, & e \notin S, \end{cases}$$

where $0 < \varepsilon < 1/(n+1)^2$. Notice that $\sum_{k=0}^{2\ell}(-1)^k v_{e_k} = (\ell+1)(\frac{1}{\ell+1}+\varepsilon) - \ell(\frac{\ell+1}{\ell}\varepsilon) = 1$, and thus it satisfies the chosen inequality with equation. However, it satisfies all other defining inequalities of ZP_i with strict inequality. Therefore, for each $S \subseteq \mathcal{E}^*$ of odd cardinality, the corresponding inequality of (34) defines a facet of ZP_i. □

3.2. Separation algorithm for the non-preemptive inequalities

In this section we describe a polynomial-time separation procedure for the inequalities (34).

Separation procedure. The main idea is that we find a longest path in a properly defined acyclic digraph. There is a unique source and a unique sink node (indexed by 0 and $2n$, respectively), and there is a node for each event in $\mathcal{E}^*$. The arcs fall into three categories: (i) there are "starting arcs" from node 0 to every node with index $h \leq 2n-3$, (ii) "intermediate arcs" starting from some node h with $1 \leq h \leq 2n-3$, and ending at some node k with $h+2 \leq k \leq 2n-1$, and (iii) "terminal arcs" starting at some node h with $3 \leq h \leq 2n-1$ and ending in the sink node $2n$. The cost of each starting arc is 0. The cost of each intermediate arc (h,k) is $cost(h,k) = \bar{z}_{i,e_h} - \min\{\bar{z}_{i,e_\ell} \ : \ \ell = h+1, ..., k-1\}$. Finally, the cost of each terminal arc $(h, 2n)$ is $\bar{z}_{i,e_h}$.

To separate a vector $\bar{z}_i \in \mathbb{R}^{\mathcal{E}^*}$ we compute the values

$$F(e_k) = \max\{F(e_h) + cost(h,k) \ : \ h = 1, .., k-2\}, \tag{41}$$

where $F(1) = F(2) = 0$.

Then, for each $F(e_k)$ we compute $F(e_k) + \bar{z}_{i,e_k}$ and compare it to the length of the longest path with alternating sign pattern found so far. If it is greater, then we store it. In the end, we obtain the largest value of a path.

The computation time for computing the cost of all the intermediate arcs from some fixed node h is $O(n)$. Since the number of arcs is $O(n^2)$, the time complexity of the entire separation procedure is $O(n^2)$.

4. Experiments

This section describes the computational results we obtain with the inequalities (see Section 2.1, 2.2, 2.5 and 3.1) and the strengthened constraints (see Section 2.3 and 2.4) described in this paper.

The experiments are conducted on a workstation with 8GB RAM and Intel(R) Xeon(R) CPU E5-2630 v4 of 2.20 GHz, and under Linux operating system. We use IBM CPLEX 12.6.3.0 with one thread and a time limit of 1000 seconds for solving the MILP models. In those runs, where the non-preemptive cuts (34) are separated, a maximum number of 500 of them are generated in the search-tree nodes of depth at most 10, using the separation algorithm described in Section 3.2.

The benchmark instances we use to perform these experiments are, on the one hand, those of [12] (Family 1, 2 and 3) and, on the other hand, some new instances we have generated (Family 4 and 5). In total, five sets of instances were used.

In [12], three families of instances were detailed. Family 1 consists of 5 10-task instances, 11 20-task instances, 9 25-task instances and 10 30-task instances. Families 2 and 3 have the same number of tasks. Family 1 consists of instances with the following characteristics:

- the resource availability B is set to 10;
- the other data are randomly generated in their corresponding interval: $W_i \in [1, 2.5*B]$, $b_i^{min} \in [1, 1 + 0.25 * W_i]$, $b_i^{max} \in [B - 0.25 * B, B]$, $r_i \in [0, 0.5 * n]$ and $d_i \in [r_i + \frac{W_i}{b_i^{min}}, r_i + \frac{W_i}{b_i^{min}} + n]$;
- f_i is the identity function, i.e., $f_i(b) = b, \ \forall i \in \mathcal{A}$.

Then, three families of instances are derived from Family 1. More precisely, there is a one-to-one mapping between an instance of Family 1 and an instance of another family. Given one instances of Family 1, the corresponding instances use the same value for b_i^{min}, b_i^{max}, r_i and d_i and then, power processing rate functions f_i are generated by setting the parameters of the function a_i and c_i, $\forall i \in A$, to a random number within the interval $[1, 10]$. Finally, the new energy requirement W_i^{new} is set to one of the following values (W_i^{old} is the value of W_i in Family 1, which is equivalent to a resource quantity since f_i is the identity function):

- a random number between 0 and $f_i(W_i^{old})$ for the second family (Family 2);
- a random number between $\frac{f_i(W_i^{old})}{2}$ and $f_i(W_i^{old})$ for the third family (Family 3).
- For instances of Family 4, W_i^{new} is computed using a time-indexed linear program (TILP) described in [12]. First, we use (TILP) to solve the instances of Family 1. In this model, the planning horizon is discretized in T time periods of size 1 and a variable b_{it} is used to represent the resource consumption of task i in period t. Then, we set W_i^{new} to $\sum_{t=1}^{T} f_i(b_{it})$.

Note that Family 4 consists of 5 10-task instances, 10 20-task instances, 8 25-task instances and 8 30-task instances. Indeed, we were only able to use instances of Family 1 that were feasible.

Finally, Family 5 consists of 5 10-task instances, 10 20-task instances, 10 25-task instances and 10 30-task instances. Instances of this family have been generated using an adaptation of the generation procedure of Family 1. More precisely, Family 5 consists of instances with the following characteristics:

- the resource availability B is set to 10;
- f_i is generated by setting the parameters of the function a_i and c_i, to a random number within the interval $[1, 10]$;
- the other data are randomly generated in their corresponding interval: $W_i \in [f_i(1), f_i(2.5 * B)]$, $b_i^{min} \in [1, 1 + 0.25 * f_i^{-1}(W_i)]$. The intervals for b_i^{max}, r_i and d_i are the same as for Family 1.

Table 1 presents the results obtained on the first family of instances (Family 1). In the table, the first column (*#tasks*) describes the number of tasks in the instance. The second column (*config*) shows the inequalities added to the model: Int or I denotes the maximum interval inequalities, $Time$ or T the maximum time inequalities, $Knap$ or K the inequalities from knapsack problems and $\overline{Preemp}$ or $\overline{P}$ the non-preemptive inequalities. Finally, Def corresponds to the default CPLEX configuration. We also test our model without CPLEX built-in cuts: NC.

The following two columns (*time* 1^{st} *sol* and *gap* 1^{st} *sol*) exhibit the time needed to find a feasible integral solution and its gap computed as

$$100 \times \left(1 - \frac{obj_{best}}{obj_{first}}\right)$$

with obj_{first} the objective value of the first integral solution found by CPLEX.

Finally, the last columns expose the final results obtained during the experiments:

- *time end*: the average time needed to solve optimally the instance (with a penalty of 1000 seconds if the instance is not solved to optimality or if no feasible solution is found);
- *gap end*: the average gap of instances for which a feasible solution has been found;
- *%solv*: the percentage of instances for which a feasible solution has been found;
- *%opt*: the percentage of instances solved to optimality;
- *#nodes*: the average number of nodes generated by CPLEX;
- *#cuts*: the average number of non-preemptive inequalities added to the model.

Tables 2, 3, 4 and 5 present the same results for Families 2, 3, 4 and 5, respectively.

Family 1. All of the 10-task instances are solved optimally in a split of seconds, however, adding inequalities I, T or K reduces the number of investigated nodes, moreover, in that case most of the instances are solved at the root.

All of the 20-task instances but one are solved optimally, however, adding inequalities I, T, K, or separating inequalities $\overline{P}$ mostly increases the execution time. Probably, it is because of that in most cases finding the first solution consumes more time.

For the biggest, especially for the 30-task instances the usage of inequalities I, T and K, and the separation of non-preemptive inequalities $\overline{P}$ increase the number of instances that are solved, and highly reduce the solution time. For example, in case of configuration I & T & K & $\overline{P}$ all but one 30-instances are solved optimally, and significantly faster than in case of any other configuration.

Family 2. All of the 10-task instances are solved. The default configuration solves only 2 out of 5 instances optimally which is less than in case of some other configurations. For example, all instances are solved optimally, if inequalities T are applied, furthermore, the knapsack constraints K are also useful.

Although, all of the 20-task and 25-task instances are solved, none of them is solved optimally, however, most of the configurations yield better results (that is, smaller final gap) than the default one. The best results are yielded when inequalities I and T are used.

Family 3. Again, all of the 10-task instances are solved. When inequalities T are used, more instances (in some cases all of them) are solved optimally than in the default configuration.

None of the 20-task instances is solved optimally, however, all of them are solved. Most of the configurations (especially when inequalities T are used) yield smaller final gap than the default one.

For the biggest instances using inequalities T also proved to be useful, that is, in case of these configurations more instances are solved than in case of the default configuration.

Table 1: Results of experiments for Family 1

#tasks	config	time 1^{st} sol	gap 1^{st} sol	time end	gap end	%solv	%opt	#nodes	#cuts
10	NC	0.1	0.5	0.1	0.0	100.0	100.0	26.6	0.0
	Def	0.1	2.4	0.1	0.0	100.0	100.0	4.0	0.0
	Int	0.1	2.6	0.1	0.0	100.0	100.0	11.4	0.0
	$Time$	0.1	4.1	0.1	0.0	100.0	100.0	1.8	0.0
	$Knap$	0.1	3.1	0.2	0.0	100.0	100.0	26.0	0.0
	$\overline{Preemp}$	0.1	2.4	0.1	0.0	100.0	100.0	4.0	0.0
	I & T & K	0.1	1.6	0.1	0.0	100.0	100.0	0.0	0.0
	I & T & $\overline{P}$	0.1	4.1	0.1	0.0	100.0	100.0	1.8	0.0
	I & K & $\overline{P}$	0.1	1.4	0.1	0.0	100.0	100.0	0.0	0.0
	T & K & $\overline{P}$	0.1	1.6	0.1	0.0	100.0	100.0	0.0	0.0
	I & T & K & $\overline{P}$	0.1	1.6	0.1	0.0	100.0	100.0	0.0	0.0
20	NC	8.4	0.2	98.9	0.0	90.9	90.9	41869.4	0.0
	Def	7.8	0.6	100.7	0.0	90.9	90.9	20948.8	0.0
	Int	8.9	0.4	100.7	0.0	90.9	90.9	17159.7	0.0
	$Time$	6.7	0.8	102.3	0.0	90.9	90.9	17143.3	0.0
	$Knap$	7.8	0.1	131.0	0.0	90.9	90.9	21768.9	0.0
	$\overline{Preemp}$	13.8	0.3	103.8	0.0	90.9	90.9	17388.4	64.9
	I & T & K	6.1	0.8	97.7	0.0	90.9	90.9	18931.1	0.0
	I & T & $\overline{P}$	6.3	0.4	97.3	0.0	90.9	90.9	21638.6	114.5
	I & K & $\overline{P}$	26.6	1.5	115.4	0.0	90.9	90.9	19096.6	99.8
	T & K & $\overline{P}$	6.3	0.9	97.0	0.0	90.9	90.9	25111.9	106.0
	I & T & K & $\overline{P}$	55.0	1.0	141.4	0.0	90.9	90.9	25971.9	133.7
25	NC	154.3	1.3	349.3	0.0	77.8	77.8	65050.6	0.0
	Def	52.3	0.2	159.4	0.0	88.9	88.9	14067.6	0.0
	Int	39.5	0.1	147.0	0.0	88.9	88.9	11786.8	0.0
	$Time$	17.7	0.2	127.3	0.0	88.9	88.9	15276.0	0.0
	$Knap$	50.7	0.5	158.0	0.0	88.9	88.9	16853.1	0.0
	$\overline{Preemp}$	28.0	0.3	251.4	0.0	77.8	77.8	17257.0	140.6
	I & T & K	21.0	0.5	130.8	0.0	88.9	88.9	17244.0	0.0
	I & T & $\overline{P}$	20.1	0.3	129.0	0.0	88.9	88.9	7263.9	144.7
	I & K & $\overline{P}$	39.6	0.5	147.2	0.0	88.9	88.9	9659.1	142.9
	T & K & $\overline{P}$	28.0	0.2	244.2	0.0	77.8	77.8	28190.3	170.4
	I & T & K & $\overline{P}$	33.8	0.1	141.3	0.0	88.9	88.9	15282.6	172.9
30	NC	67.8	0.2	544.0	0.0	50.0	50.0	48660.7	0.0
	Def	127.9	0.1	390.2	0.0	70.0	70.0	21237.3	0.0
	Int	139.5	0.6	312.8	0.0	80.0	80.0	19084.5	0.0
	$Time$	110.0	0.3	134.4	0.0	90.0	90.0	9703.5	0.0
	$Knap$	71.2	0.5	275.2	0.0	80.0	80.0	12459.8	0.0
	$\overline{Preemp}$	51.6	0.4	249.7	0.0	80.0	80.0	12102.1	143.3
	I & T & K	33.0	0.1	131.5	0.0	90.0	90.0	7509.0	0.0
	I & T & $\overline{P}$	89.6	0.2	159.8	0.0	90.0	90.0	10393.0	96.9
	I & K & $\overline{P}$	77.5	0.2	284.1	0.0	80.0	80.0	16016.5	100.2
	T & K & $\overline{P}$	54.7	1.1	61.6	0.0	90.0	90.0	4850.7	172.6
	I & T & K & $\overline{P}$	40.7	0.2	48.8	0.0	90.0	90.0	3375.8	182.6

Family 4. All of the 10-task instances are solved, and most of the configurations yield more optimally solved instances than the default.

For the 20-task instances configuration I & T & K & $\overline{P}$ yields the best result, since all instances but one are solved.

Most of the biggest instances are unsolved, just a few configurations can solve some of them. The non-preemptive inequalities $\overline{P}$ are seemed to be useful for these instances.

Family 5. Similar to the previous families, all the 10-task instances are solved, however, most of the configurations solve more instances optimally than the default.

In case of 20-task and 25-task instances some configurations using inequalities T yield

Table 2: Results of experiments for Family 2

#tasks	config	time 1^{st} sol	gap 1^{st} sol	time end	gap end	%solv	%opt	#nodes	#cuts
10	NC	0.1	17.3	1000.0	35.4	100.0	0.0	1706369.8	0.0
	Def	0.2	9.9	667.3	8.6	100.0	40.0	417152.6	0.0
	Int	0.2	16.0	747.6	4.8	100.0	40.0	374184.6	0.0
	$Time$	0.1	13.2	32.5	0.0	100.0	100.0	19004.0	0.0
	$Knap$	0.1	18.9	436.4	0.8	100.0	80.0	254775.4	0.0
	$\overline{Preemp}$	0.2	9.9	666.1	8.5	100.0	40.0	408366.0	0.0
	I & T & K	0.1	12.7	800.1	5.8	100.0	20.0	516149.0	0.0
	I & T & $\overline{P}$	0.1	8.2	253.9	0.0	100.0	80.0	103391.0	0.0
	I & K & $\overline{P}$	0.1	13.5	731.8	3.0	100.0	40.0	328591.6	0.0
	T & K & $\overline{P}$	0.1	14.2	246.6	0.1	100.0	80.0	100104.6	0.0
	I & T & K & $\overline{P}$	0.1	12.7	800.1	5.8	100.0	20.0	513904.8	0.0
20	NC	27.4	18.7	1000.0	71.1	100.0	0.0	200556.5	0.0
	Def	23.8	11.9	1000.0	27.7	100.0	0.0	125373.5	0.0
	Int	26.3	9.8	1000.0	33.8	100.0	0.0	127210.8	0.0
	$Time$	6.4	11.4	1000.0	22.0	100.0	0.0	157911.8	0.0
	$Knap$	6.5	13.3	1000.0	33.8	100.0	0.0	135737.6	0.0
	$\overline{Preemp}$	28.1	12.2	1000.0	27.7	100.0	0.0	108983.2	216.5
	I & T & K	4.5	10.7	1000.0	20.0	100.0	0.0	121328.7	0.0
	I & T & $\overline{P}$	7.1	13.1	1000.0	20.2	100.0	0.0	164106.0	25.9
	I & K & $\overline{P}$	3.7	14.9	1000.0	31.1	100.0	0.0	133074.3	0.0
	T & K & $\overline{P}$	4.4	15.0	1000.0	20.8	100.0	0.0	135226.5	20.6
	I & T & K & $\overline{P}$	3.7	11.0	1000.0	20.2	100.0	0.0	120837.5	7.3
25	NC	189.3	11.6	1000.0	82.2	100.0	0.0	78445.7	0.0
	Def	132.2	12.4	1000.0	62.9	100.0	0.0	66810.8	0.0
	Int	150.8	12.8	1000.0	55.4	100.0	0.0	59970.3	0.0
	$Time$	55.0	16.8	1000.0	48.8	100.0	0.0	67085.3	0.0
	$Knap$	49.6	20.7	1000.0	64.5	100.0	0.0	52675.8	0.0
	$\overline{Preemp}$	142.2	8.8	1000.0	63.0	100.0	0.0	47572.1	363.7
	I & T & K	45.3	15.0	1000.0	51.5	100.0	0.0	65229.1	0.0
	I & T & $\overline{P}$	102.9	13.8	1000.0	49.7	100.0	0.0	76823.2	220.7
	I & K & $\overline{P}$	127.0	16.6	1000.0	69.5	100.0	0.0	40168.2	224.9
	T & K & $\overline{P}$	75.8	19.1	1000.0	54.2	100.0	0.0	61507.1	56.3
	I & T & K & $\overline{P}$	61.1	17.5	1000.0	51.4	100.0	0.0	60476.2	128.6
30	NC	550.6	14.4	1000.0	84.6	70.0	0.0	28517.2	0.0
	Def	564.7	12.6	1000.0	73.0	80.0	0.0	25094.1	0.0
	Int	457.6	13.9	1000.0	80.2	50.0	0.0	26694.0	0.0
	$Time$	361.7	13.4	1000.0	62.8	80.0	0.0	36519.0	0.0
	$Knap$	170.8	16.1	1000.0	78.5	80.0	0.0	14299.4	0.0
	$\overline{Preemp}$	415.4	12.9	1000.0	79.5	40.0	0.0	12161.2	464.7
	I & T & K	287.7	15.9	1000.0	57.5	80.0	0.0	24293.8	0.0
	I & T & $\overline{P}$	328.3	19.0	1000.0	41.7	70.0	0.0	21213.2	401.7
	I & K & $\overline{P}$	360.6	17.5	1000.0	68.6	60.0	0.0	7261.2	354.1
	T & K & $\overline{P}$	222.5	22.8	1000.0	50.4	70.0	0.0	19840.8	286.8
	I & T & K & $\overline{P}$	167.3	17.7	1000.0	57.0	50.0	0.0	13915.3	382.3

the best results.

All but one of the 30-task instances are unsolved.

Table 6 presents a comparison of the strength of the various cutting plane settings in terms of the bounds they provide. That is, for each setting of cut generation and for each set of instances, we indicate how many times the final upper or lower bound is the best among all the settings on the corresponding set of instances. That is, in a particular cell of the table, x/y means that out of y instances (the number of instances in the corresponding family) the particular setting gives x times the best upper bound (UB), or lower bound (LB).

Table 3: Results of experiments for Family 3

#tasks	config	time 1^{st} sol	gap 1^{st} sol	time end	gap end	%solv	%opt	#nodes	#cuts
10	NC	0.7	22.2	1000.0	65.3	100.0	0.0	1144192.6	0.0
	Def	0.4	8.5	413.4	21.2	100.0	60.0	212291.6	0.0
	Int	0.3	12.4	553.9	20.1	100.0	60.0	258658.0	0.0
	$Time$	0.2	15.4	215.3	1.5	100.0	80.0	89669.2	0.0
	$Knap$	0.1	17.5	258.1	13.0	100.0	80.0	143616.0	0.0
	$\overline{Preemp}$	0.4	11.5	413.4	21.2	100.0	60.0	220165.0	17.4
	I & T & K	0.1	12.1	230.0	1.8	100.0	80.0	83606.2	0.0
	I & T & $\overline{P}$	0.2	15.9	3.1	0.0	100.0	100.0	849.6	0.0
	I & K & $\overline{P}$	0.2	18.0	684.1	24.4	100.0	40.0	255240.6	0.0
	T & K & $\overline{P}$	0.1	13.7	54.9	0.0	100.0	100.0	12744.4	0.0
	I & T & K & $\overline{P}$	0.1	12.1	230.1	1.8	100.0	80.0	84035.6	0.0
20	NC	153.8	11.1	1000.0	77.6	100.0	0.0	139239.1	0.0
	Def	78.6	7.4	1000.0	26.0	100.0	0.0	67115.4	0.0
	Int	124.7	13.2	1000.0	37.0	100.0	0.0	83033.3	0.0
	$Time$	74.4	6.1	1000.0	16.3	100.0	0.0	76683.6	0.0
	$Knap$	57.8	9.0	1000.0	27.2	100.0	0.0	65957.9	0.0
	$\overline{Preemp}$	118.8	7.8	1000.0	26.0	100.0	0.0	61463.4	418.6
	I & T & K	38.6	14.0	1000.0	13.7	100.0	0.0	66283.3	0.0
	I & T & $\overline{P}$	102.6	11.3	1000.0	13.2	100.0	0.0	64360.6	266.4
	I & K & $\overline{P}$	61.4	10.5	1000.0	26.9	100.0	0.0	58488.7	281.5
	T & K & $\overline{P}$	74.2	12.0	1000.0	13.8	100.0	0.0	52603.2	215.0
	I & T & K & $\overline{P}$	36.8	13.5	1000.0	13.7	100.0	0.0	65961.1	176.3
25	NC	293.5	10.8	1000.0	73.6	66.7	0.0	50796.1	0.0
	Def	396.6	7.8	1000.0	36.4	88.9	0.0	43135.2	0.0
	Int	281.5	9.3	1000.0	47.7	100.0	0.0	49704.4	0.0
	$Time$	265.9	8.5	1000.0	28.2	88.9	0.0	37930.0	0.0
	$Knap$	298.9	11.9	1000.0	39.1	77.8	0.0	33752.7	0.0
	$\overline{Preemp}$	371.8	8.8	1000.0	30.4	77.8	0.0	29643.4	383.0
	I & T & K	176.6	7.0	1000.0	22.0	88.9	0.0	34853.8	0.0
	I & T & $\overline{P}$	273.7	9.1	1000.0	19.7	100.0	0.0	30966.2	426.1
	I & K & $\overline{P}$	358.1	10.6	1000.0	39.7	88.9	0.0	21387.3	340.4
	T & K & $\overline{P}$	289.4	7.3	1000.0	19.1	100.0	0.0	29566.4	420.4
	I & T & K & $\overline{P}$	310.6	8.0	1000.0	22.4	88.9	0.0	23558.0	391.8
30	NC	417.7	12.7	1000.0	81.6	30.0	0.0	22856.2	0.0
	Def	458.8	12.5	1000.0	53.3	20.0	0.0	19414.7	0.0
	Int	327.8	5.8	1000.0	28.8	10.0	0.0	22624.0	0.0
	$Time$	419.8	10.3	1000.0	21.6	30.0	0.0	24614.9	0.0
	$Knap$	199.3	13.9	1000.0	53.8	20.0	0.0	13285.8	0.0
	$\overline{Preemp}$	603.7	11.2	1000.0	22.4	10.0	0.0	6807.1	480.2
	I & T & K	251.6	10.2	1000.0	28.1	40.0	0.0	16166.1	0.0
	I & T & $\overline{P}$	559.5	13.8	1000.0	27.5	50.0	0.0	13563.6	449.9
	I & K & $\overline{P}$	489.0	0.8	1000.0	63.7	30.0	0.0	2429.1	430.2
	T & K & $\overline{P}$	667.0	13.0	1000.0	29.3	40.0	0.0	9474.4	457.9
	I & T & K & $\overline{P}$	527.9	12.5	1000.0	28.9	40.0	0.0	11283.0	456.3

Final conclusions. The results are diversified. We cannot point out a configuration which always outperforms the other ones, however, in most cases configuration I & T & K (especially, because of inequalities T) yields one of the best results in the sense of number of solved instances or final gap (see Tables 1–5).

Although non-preemptive inequalities $\overline{P}$ are not proved to be useful individually, in most of the cases combining them with inequalities I and T (i.e., configuration I & T & $\overline{P}$) yields the best results in the sense of number of times the configuration reaches the best lower bound (see Table 6). That is, the non-preemptive inequalities could help to strengthen the lower bound, however, sometimes it causes not to find a strong upper bound.

Table 4: Results of experiments for Family 4

#tasks	config	time 1^{st} sol	gap 1^{st} sol	time end	gap end	%solv	%opt	#nodes	#cuts
10	NC	0.3	16.5	1000.0	46.9	100.0	0.0	945764.2	0.0
	Def	0.3	16.3	419.8	1.9	100.0	60.0	63411.4	0.0
	Int	0.4	7.6	203.9	0.1	100.0	80.0	26271.4	0.0
	$Time$	0.4	11.6	174.7	0.0	100.0	100.0	21129.8	0.0
	$Knap$	0.2	15.0	327.5	0.0	100.0	80.0	46623.6	0.0
	$\overline{Preemp}$	0.3	16.3	419.8	1.9	100.0	60.0	63194.0	0.0
	I & T & K	0.1	8.3	203.1	0.0	100.0	80.0	17921.8	0.0
	I & T & $\overline{P}$	0.4	11.9	600.6	4.4	100.0	40.0	154622.8	0.0
	I & K & $\overline{P}$	0.4	13.8	212.3	0.2	100.0	80.0	19092.8	0.0
	T & K & $\overline{P}$	0.2	9.3	232.3	0.0	100.0	80.0	32876.8	0.0
	I & T & K & $\overline{P}$	0.1	8.3	203.1	0.0	100.0	80.0	17747.0	0.0
20	NC	398.1	15.0	1000.0	78.6	70.00	0.00	133094.6	0.0
	Def	339.0	9.2	1000.0	21.3	60.00	0.00	39884.4	0.0
	Int	398.1	6.5	1000.0	11.5	70.00	0.00	29403.9	0.0
	$Time$	274.7	7.7	1000.0	11.0	80.00	0.00	37946.5	0.0
	$Knap$	223.5	7.4	1000.0	11.3	70.00	0.00	35217.7	0.0
	$\overline{Preemp}$	447.2	9.0	1000.0	21.8	60.00	0.00	26216.5	471.1
	I & T & K	270.2	8.2	1000.0	11.5	80.00	0.00	35761.0	0.0
	I & T & $\overline{P}$	265.7	9.5	1000.0	12.8	60.00	0.00	26944.9	502.2
	I & K & $\overline{P}$	367.4	6.2	1000.0	14.9	80.00	0.00	24175.6	448.8
	T & K & $\overline{P}$	276.5	5.6	1000.0	10.9	80.00	0.00	24986.9	452.0
	I & T & K & $\overline{P}$	438.3	6.7	1000.0	12.1	90.00	0.00	27712.2	470.8
25	NC	-	-	1000.0	-	0.0	0.0	41790.5	0.0
	Def	-	-	1000.0	-	0.0	0.0	22356.8	0.0
	Int	-	-	1000.0	-	0.0	0.0	23163.9	0.0
	$Time$	-	-	1000.0	-	0.0	0.0	20394.8	0.0
	$Knap$	-	-	1000.0	-	0.0	0.0	15837.1	0.0
	$\overline{Preemp}$	345.6	11.7	1000.0	27.2	12.5	0.0	17164.4	502.5
	I & T & K	786.3	6.0	1000.0	12.0	12.5	0.0	14878.1	0.0
	I & T & $\overline{P}$	-	-	1000.0	-	0.0	0.0	12999.6	458.9
	I & K & $\overline{P}$	-	-	1000.0	-	0.0	0.0	7156.0	478.8
	T & K & $\overline{P}$	-	-	1000.0	-	0.0	0.0	9611.6	506.3
	I & T & K & $\overline{P}$	-	-	1000.0	-	0.0	0.0	7599.9	499.8
30	NC	-	-	1000.0	-	0.0	0.0	22912.3	0.0
	Def	-	-	1000.0	-	0.0	0.0	14394.1	0.0
	Int	-	-	1000.0	-	0.0	0.0	18693.3	0.0
	$Time$	-	-	1000.0	-	0.0	0.0	16196.3	0.0
	$Knap$	-	-	1000.0	-	0.0	0.0	9304.1	0.0
	$\overline{Preemp}$	-	-	1000.0	-	0.0	0.0	6993.6	509.4
	I & T & K	971.9	5.7	1000.0	24.2	25.0	0.0	12398.5	0.0
	I & T & $\overline{P}$	800.3	12.2	1000.0	17.5	25.0	0.0	9757.0	505.0
	I & K & $\overline{P}$	-	-	1000.0	-	0.0	0.0	2381.4	493.9
	T & K & $\overline{P}$	755.5	7.7	1000.0	17.8	12.5	0.0	3745.4	505.1
	I & T & K & $\overline{P}$	-	-	1000.0	-	0.0	0.0	6556.4	504.1

5. Conclusion and further researches

In this paper, we have presented four sets of inequalities improving the existing event-based MILP of [12]. Indeed, these sets of inequalities allow us to solve more instances. They also help us to obtain solution (first and final) of better quality and/or more quickly, especially when the number of tasks grows. Moreover, one of these sets is used to give a minimal description of the polyhedra of all feasible assignments to the on/off binary variable for a single activity. A polynomial separation algorithm is also provided for these inequalities.

There are numerous directions for further research. In particular, a challenge should

Table 5: Results of experiments for Family 5

#tasks	config	time 1^{st} sol	gap 1^{st} sol	time end	gap end	%solv	%opt	#nodes	#cuts
10	NC	0.6	27.7	1000.0	36.6	100.0	0.0	888960.0	0.0
	Def	1.7	12.2	403.0	0.9	100.0	60.0	37382.8	0.0
	Int	9.4	11.4	406.6	4.3	100.0	60.0	65144.2	0.0
	$Time$	1.1	14.3	470.7	1.6	100.0	60.0	63337.4	0.0
	$Knap$	0.4	12.4	238.5	1.9	100.0	80.0	32054.4	0.0
	$\overline{Preemp}$	1.1	11.0	402.0	0.9	100.0	60.0	35571.8	27.6
	$I \& T \& K$	1.0	16.4	370.6	7.2	100.0	80.0	85735.0	0.0
	$I \& T \& \overline{P}$	1.1	12.5	402.9	5.9	100.0	60.0	96260.0	47.0
	$I \& K \& \overline{P}$	0.2	11.6	323.3	1.5	100.0	80.0	43412.4	0.0
	$T \& K \& \overline{P}$	1.5	12.5	509.1	5.9	100.0	60.0	90695.4	48.8
	$I \& T \& K \& \overline{P}$	0.9	14.3	402.4	7.2	100.0	80.0	91051.8	12.8
20	NC	237.4	16.1	1000.0	73.9	80.0	0.0	134351.9	0.0
	Def	189.1	11.7	1000.0	29.2	90.0	0.0	52460.2	0.0
	Int	106.0	12.0	1000.0	27.7	70.0	0.0	42400.3	0.0
	$Time$	141.6	9.1	1000.0	16.1	90.0	0.0	44561.3	0.0
	$Knap$	126.4	9.9	1000.0	28.1	80.0	0.0	42095.2	0.0
	$\overline{Preemp}$	83.9	10.9	1000.0	26.0	60.0	0.0	35379.8	437.1
	$I \& T \& K$	201.3	10.9	1000.0	17.0	90.0	0.0	39237.3	0.0
	$I \& T \& \overline{P}$	186.6	9.3	1000.0	21.3	90.0	0.0	34446.4	393.3
	$I \& K \& \overline{P}$	219.9	10.3	1000.0	26.9	90.0	0.0	30679.5	433.5
	$T \& K \& \overline{P}$	185.7	10.9	1000.0	26.3	70.0	0.0	29844.3	413.1
	$I \& T \& K \& \overline{P}$	129.5	11.0	1000.0	19.5	70.0	0.0	27032.6	449.6
25	NC	807.3	8.7	1000.0	77.2	20.0	0.0	45998.4	0.0
	Def	459.9	8.8	1000.0	23.9	30.0	0.0	24725.3	0.0
	Int	405.6	14.7	1000.0	34.3	40.0	0.0	23985.2	0.0
	$Time$	491.1	8.5	1000.0	13.6	30.0	0.0	24753.5	0.0
	$Knap$	405.4	8.0	1000.0	21.1	50.0	0.0	21172.4	0.0
	$\overline{Preemp}$	495.4	8.1	1000.0	23.8	20.0	0.0	12495.1	506.9
	$I \& T \& K$	294.1	10.9	1000.0	18.1	50.0	0.0	18970.1	0.0
	$I \& T \& \overline{P}$	415.7	11.7	1000.0	10.7	30.0	0.0	16447.4	507.2
	$I \& K \& \overline{P}$	380.8	10.6	1000.0	19.1	10.0	0.0	7705.9	479.5
	$T \& K \& \overline{P}$	586.2	6.2	1000.0	21.5	40.0	0.0	10152.1	504.0
	$I \& T \& K \& \overline{P}$	559.1	10.8	1000.0	19.2	50.0	0.0	11850.0	452.1
30	NC	-	-	1000.0	-	0.0	0.0	23571.9	0.0
	Def	-	-	1000.0	-	0.0	0.0	15964.4	0.0
	Int	-	-	1000.0	-	0.0	0.0	18814.4	0.0
	$Time$	-	-	1000.0	-	0.0	0.0	13495.2	0.0
	$Knap$	-	-	1000.0	-	0.0	0.0	10445.9	0.0
	$\overline{Preemp}$	-	-	1000.0	-	0.0	0.0	7881.9	491.2
	$I \& T \& K$	470.9	5.0	1000.0	17.7	10.0	0.0	12079.1	0.0
	$I \& T \& \overline{P}$	-	-	1000.0	-	0.0	0.0	10247.8	486.5
	$I \& K \& \overline{P}$	-	-	1000.0	-	0.0	0.0	3359.2	496.9
	$T \& K \& \overline{P}$	-	-	1000.0	-	0.0	0.0	4154.1	466.2
	$I \& T \& K \& \overline{P}$	-	-	1000.0	-	0.0	0.0	5495.5	501.1

be to find symmetry breaking inequalities along with a minimal description of the polyhedra of all asymmetric feasible assignments of the on/off variable not for a single activity but for all of them. The adaptation of this work to the Resource-Constrained Project Scheduling Problem is also an interesting direction for additional results.

Acknowledgements

This work of Tamás Kis has been supported by the OTKA grant K112881, and by the GINOP-2.3.2-15-2016-00002 grant of the Ministry of National Economy of Hungary.

Table 6: Number of times a method reaches the best upper and lower bounds

		Family 1		Family 2		Family 3		Family 4		Family 5	
#tasks	config	UB	LB	UB	LB	UB	LB	UB	LB	UB	LB
10	NC	5/5	5/5	5/5	0/5	5/5	0/5	4/5	0/5	5/5	0/5
	Def	5/5	5/5	5/5	2/5	5/5	3/5	5/5	3/5	5/5	3/5
	Int	5/5	5/5	5/5	2/5	5/5	3/5	5/5	4/5	5/5	3/5
	$Time$	5/5	5/5	5/5	5/5	5/5	4/5	5/5	5/5	5/5	3/5
	$Knap$	5/5	5/5	5/5	4/5	5/5	4/5	5/5	4/5	5/5	4/5
	$\overline{Preemp}$	5/5	5/5	5/5	2/5	5/5	3/5	5/5	3/5	5/5	3/5
	I & T & K	5/5	5/5	5/5	1/5	5/5	4/5	5/5	4/5	5/5	4/5
	I & T & $\overline{P}$	5/5	5/5	5/5	4/5	5/5	5/5	5/5	2/5	5/5	3/5
	I & K & $\overline{P}$	5/5	5/5	5/5	2/5	5/5	2/5	5/5	4/5	5/5	4/5
	T & K & $\overline{P}$	5/5	5/5	5/5	4/5	5/5	5/5	5/5	4/5	5/5	3/5
	I & T & K & $\overline{P}$	5/5	5/5	5/5	1/5	5/5	4/5	5/5	4/5	5/5	4/5
20	NC	10/11	11/11	11/11	0/11	11/11	0/11	6/10	0/10	7/10	0/10
	Def	10/11	11/11	11/11	3/11	11/11	0/11	2/10	2/10	5/10	0/10
	Int	10/11	11/11	11/11	0/11	11/11	0/11	4/10	3/10	5/10	4/10
	$Time$	10/11	11/11	11/11	4/11	10/11	2/11	3/10	2/10	4/10	3/10
	$Knap$	10/11	11/11	11/11	1/11	11/11	1/11	6/10	2/10	4/10	2/10
	$\overline{Preemp}$	10/11	11/11	11/11	3/11	10/11	0/11	4/10	2/10	4/10	0/10
	I & T & K	10/11	11/11	11/11	3/11	10/11	4/11	5/10	1/10	6/10	2/10
	I & T & $\overline{P}$	10/11	11/11	11/11	3/11	11/11	5/11	2/10	2/10	6/10	2/10
	I & K & $\overline{P}$	10/11	11/11	11/11	0/11	9/11	1/11	4/10	3/10	6/10	2/10
	T & K & $\overline{P}$	10/11	11/11	11/11	3/11	10/11	2/11	5/10	4/10	4/10	4/10
	I & T & K & $\overline{P}$	10/11	11/11	11/11	3/11	11/11	4/11	2/10	1/10	5/10	1/10
25	NC	7/9	9/9	9/9	0/9	6/9	0/9	6/8	0/8	0/10	0/10
	Def	8/9	9/9	9/9	0/9	6/9	0/9	2/8	2/8	0/10	0/10
	Int	8/9	9/9	9/9	1/9	7/9	1/9	4/8	3/8	0/10	2/10
	$Time$	8/9	9/9	9/9	6/9	6/9	2/9	3/8	2/8	0/10	1/10
	$Knap$	8/9	9/9	9/9	0/9	5/9	0/9	6/8	2/8	0/10	0/10
	$\overline{Preemp}$	7/9	9/9	9/9	0/9	6/9	0/9	4/8	2/8	1/10	0/10
	I & T & K	8/9	9/9	9/9	4/9	6/9	3/9	5/8	1/8	1/10	1/10
	I & T & $\overline{P}$	8/9	9/9	8/9	4/9	7/9	5/9	2/8	2/8	0/10	1/10
	I & K & $\overline{P}$	8/9	9/9	9/9	1/9	5/9	0/9	4/8	3/8	0/10	1/10
	T & K & $\overline{P}$	7/9	9/9	9/9	2/9	6/9	3/9	5/8	4/8	0/10	1/10
	I & T & K & $\overline{P}$	8/9	9/9	9/9	4/9	5/9	1/9	2/8	1/8	0/10	3/10
30	NC	5/10	10/10	6/10	0/10	3/10	0/10	0/8	0/8	0/10	0/10
	Def	7/10	10/10	5/10	1/10	0/10	0/10	0/8	0/8	0/10	0/10
	Int	8/10	10/10	4/10	0/10	1/10	0/10	0/8	0/8	0/10	0/10
	$Time$	9/10	9/10	8/10	1/10	1/10	2/10	0/8	2/8	0/10	3/10
	$Knap$	8/10	10/10	8/10	1/10	2/10	0/10	0/8	0/8	0/10	1/10
	$\overline{Preemp}$	8/10	10/10	4/10	1/10	1/10	0/10	0/8	0/8	0/10	0/10
	I & T & K	9/10	10/10	8/10	2/10	2/10	4/10	0/8	1/8	1/10	2/10
	I & T & $\overline{P}$	9/10	9/10	6/10	9/10	3/10	4/10	2/8	2/8	0/10	1/10
	I & K & $\overline{P}$	8/10	10/10	4/10	1/10	1/10	0/10	0/8	1/8	0/10	0/10
	T & K & $\overline{P}$	9/10	9/10	6/10	3/10	2/10	2/10	0/8	2/8	0/10	4/10
	I & T & K & $\overline{P}$	9/10	9/10	4/10	2/10	2/10	4/10	0/8	0/8	0/10	2/10

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 (2014). Energetic reasoning for energy-constrained scheduling with a continuous resource. *Journal of Scheduling*, 18(3): 225–241.

[3] J. Błażewicz, K. H. Ecker, E. Pesch, G. Schmidt and J. Węglarz (2001). *Scheduling computer and manufacturing processes*, Springer-Verlag, Berlin/Heidelberg.

[4] J. Błażewicz, M. Machowiak, J. Węglarz, M. Kovalyov and D. Trystram (2004). Scheduling malleable tasks on parallel processors to minimize the makespan. *Annals of Operations Research*, 129(1-4): 65–80.

[5] J. Edmonds and R. Giles (1977). A min-max relation for submodular functions on graphs. *Studies in Integer Programming*, Annals of Discrete Mathematics 1, 185-204.

[6] C.-U. Fündeling and N. Trautmann (2010). A priority-rule method for project scheduling with work-content constraints. *European Journal of Operational Research*, 203(3): 568–574.

[7] H. Gröflin and T. M. Liebling (1981). Connecting and alternating vectors: polyhedra and algorithms. *Mathematical Programming* 20, 233-244.

[8] J. Józefowska, M. Mika, R. Różycki, G. Waligóra and J. Węglarz (1999). Project scheduling under discrete and continuous resources. In: J. Węglarz (Ed.), *Project Scheduling: Recent Models, Algorithms, and Applications.* Kluwer Academic Publishers, Boston, pp. 289–308.

[9] T. Kis (2005). A branch-and-cut algorithm for scheduling of project with variable-intensity activities. *Mathematical Programming*, Series A, 103: 515–539.

[10] A. Naber (2017). Resource-constrained project scheduling with flexible resource profiles in continuous time. *Computers & Operations Research*, 84: 33–45.

[11] A. Naber and R. Kolisch (2014). MIP models for resource-constrained project scheduling with flexible resource profiles. *European Journal of Operational Research*, 239: 335–348.

[12] M. Nattaf, C. Artigues, P. Lopez and D. Rivreau (2016). Energetic reasoning and mixed-integer linear programming for scheduling with a continuous resource and linear efficiency functions. *OR Spectrum*, 38: 459-492.

[13] M. Nattaf (2016). *Scheduling under energy constraints (In French)*, Université Paul Sabatier - Toulouse III.

[14] M. Nattaf, C. Artigues and P. Lopez (2015). A hybrid exact method for a scheduling problem with a continuous resource and energy constraints. *Constraints*, 20(3): 304–324.

[15] M. Nattaf, C. Artigues and P. Lopez (2017). Cumulative scheduling with variable task profiles and concave piecewise linear processing rate functions. *Constraints*, 22(4): 530–547.

[16] J. P. Sousa (1989). *Time indexed formulations of non-preemptive single-machine scheduling problems*, Ph.D. thesis, Université Catholique de Louvain, Louvain-la-Neuve, Belgium

[17] A. Tesch (2015). *Compact MIP Models for the Resource-Constrained Project Scheduling Problem.* Technische Universität Berlin.

[18] J. Węglarz, J. Józefowska, M. Mika and G. Waligóra (2009). Project scheduling with finite or infinite number of activity processing modes - A survey. *European Journal of Operational Research*, 208: 177–205.