

## THEORETICAL INSIGHTS AND A NEW CLASS OF VALID INEQUALITIES FOR THE TEMPORAL BIN PACKING PROBLEM WITH FIRE-UPS

John Martinovic<sup>1\*</sup> and Nico Strasdat<sup>2</sup>

Received February 20, 2024 / Accepted June 20, 2024

**ABSTRACT.** The temporal bin packing problem with fire-ups (TBPP-FU) is a two-dimensional packing problem where one geometric dimension is replaced by a time horizon. The given items (jobs) are characterized by a resource consumption, that occurs exclusively during an activity interval, and they have to be placed on servers so that the capacity constraint is respected at any time. For energy efficiency reasons, an optimal assignment shall minimize a weighted sum of the number of servers in use and the number of switch-on processes (so-called fire-ups) resulting from the selected configuration. The associated ILP formulations are typically large in size, so that, in the recent past, several model improvements have already been proposed in the literature. However, with only one exception, all these techniques do not address the quality of the LP bound which is another crucial factor for the size of the branch-and-bound trees generated during the solution process. To this end, we present a new class of valid inequalities, contributing to a stronger LP relaxation, and discuss their numerical benefits by computational experiments based on benchmark instances and new test sets. Remarkably, the new cuts also lead to theoretical results about the optimal value of the LP relaxation.

**Keywords:** cutting and packing, temporal bin packing, LP relaxation, valid inequalities.

### 1 INTRODUCTION

In view of the steadily growing energy demands of data centers in the IT industry, the low-resource management of server clusters has become a central concern in computer science and electrical engineering, see Andrae & Edler (2015); Jones (2018). In addition to concrete technical improvements, such as in the hardware components involved (see Fettweis et al. (2019)), operations research methods for optimally assigning jobs to computing units are acquiring increasing importance. Originating from early contributions to multiprocessor scheduling in the

---

\*Corresponding author

<sup>1</sup>Institute of Numerical Mathematics, Technische Universität Dresden, 01062 Dresden, Germany – E-mail: john.martinovic@tu-dresden.de – <http://orcid.org/0000-0002-5428-9351>

<sup>2</sup>Institute of Numerical Mathematics, Technische Universität Dresden, 01062 Dresden, Germany – E-mail: nico.strasdat@tu-dresden.de – <http://orcid.org/0000-0003-0953-3367>

1970s (see Coffman Jr. et al. (1978)), a wide variety of new application areas for discrete optimization has emerged as a result of these developments, see Lopez-Pires & Baran (2015) for a well-structured overview. Many of these problems are based on the classical *bin packing problem (BPP)*, where a set of given jobs (or items) has to be packed on as few servers as possible (Delorme et al., 2016; Scheithauer, 2018; Valério de Carvalho, 2002). Among a large number of possible generalizations, the *stochastic BPP (SBPP)* (see Cohen et al. (2019); Martinovic & Selch (2021); Wang et al. (2011)) and the *temporal BPP (TBPP)* (see De Cauwer et al. (2016); Dell'Amico et al. (2020)) have been identified as more application-oriented descriptions of the underlying real-world scenario. While the SBPP, which is not considered here in more detail, takes into account the generally imperfect information about the expected jobs, the TBPP primarily introduces an additional time dimension. By that, we mean that each item is characterized by a resource consumption occurring exclusively during a specific time interval. The given items are then to be distributed among as few servers of a given capacity as possible, so that none of the employed executing units is overcommitted at any time.

The TBPP was first described in a specific application from the field of computer science (see De Cauwer et al. (2016)) and has been thoroughly studied in the context of only one purely mathematical contribution, see Dell'Amico et al. (2020). In that publication, the authors present various exact and heuristic solution approaches and shed light on their mutual (dominance) relations. Finally, based on extensive test calculations, a sophisticated branch-and-bound method is identified as, currently, the best solution technique. Apart from that, the TBPP owes its theoretical relevance to the fact that it is naturally related to a large number of other problems from cutting and packing. These include, among others, knapsack problems (Cacchiani et al., 2022a,b), in particular the temporal knapsack problem (Bartlett et al., 2005; Caprara et al., 2013, 2016), and classical two-dimensional packing problems, such as the strip packing problem (Côté et al., 2014; Iori et al., 2021). However, the latter class is typically based on another item placement policy<sup>1</sup>, thus establishing the TBPP as an independent field of research.

Very recently, an extended problem called *temporal bin packing problem with fire-ups (TBPP-FU)* was introduced in Aydın et al. (2020). In that scenario, a job-to-server assignment is assessed by (i) the number of servers required and, in addition, (ii) the number of switch-on processes of the servers in use, referred to as *fire-ups*. More precisely, a fire-up has to be counted whenever a server goes from an inactive state (no load at all) into an active state (executing some job), and the total number of fire-ups is added to the objective function in a weighted-sum fashion. The resulting integer programs are typically very large in size, have many symmetric solutions, and possess a fundamentally different solution set than the TBPP. Thus, previous work in this area has mainly focused on possibilities to reduce the number of variables and constraints of the associated ILP models and to improve their computational performance. More precisely, given that the ILP models M1 and M2, introduced in Aydın et al. (2020), could not even successfully cope with half of the benchmark instances in a reasonable time, some significant progress has

---

<sup>1</sup>A more detailed explanation is later contained in Fig. 1, since we would like to preface it with the introductory notation in Section 2 for better understanding.

been made in this regard by the reduction methods proposed in Martinovic et al. (2021) and Martinovic et al. (2022), see Subsect. 4.1 for a short summary. However, the literature to date does not convincingly elaborate on the extent to which the TBPP and the TBPP-FU differ from a theoretical perspective. Moreover, some significant aspects and properties of the TBPP-FU models have remained almost untouched so far and will therefore be examined in more detail here. More specifically, the following main results are presented in this paper:

- We show that temporal decompositions cannot be applied to solve the TBPP-FU, in general. By that, we present a new major difference between the TBPP and its extension involving fire-ups. (→ Section 2)
- We introduce a new class of valid cuts (for M1) facilitating the correct recognition of fire-ups and contributing to better LP bounds. (→ Section 3)
- We constructively solve the LP relaxations of M1 and M2. This is an advantage over the only known result (Aydin et al., 2020, Proposition 4.1) in the literature and provides further insights into which properties of an instance influence the various LP bounds. (→ Section 3)
- Based on computational tests with benchmark instances and new test sets we show the numerical benefits of the additional cuts for different versions of M1-type formulations for the TBPP-FU. (→ Section 4)

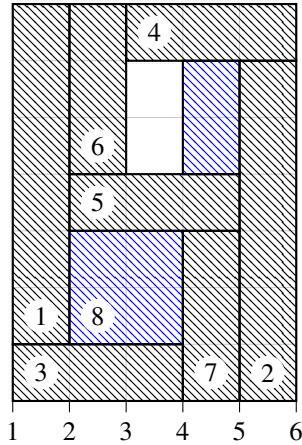
## 2 PRELIMINARIES AND BASIC MODELS

We consider a set of  $n \in \mathbb{N}$  items, also referred to as *jobs*, indexed by  $i \in I := \{1, \dots, n\}$ , that are specified by a *resource demand*  $c_i \in \mathbb{N}$ , and an *activity interval*  $[s_i, e_i]$  with  $s_i < e_i$  denoting the *starting time* and *ending time*, respectively. Moreover, we define the following sets:

- $T := \bigcup_{i \in I} \{s_i, e_i\}$  as the set of all relevant time instants,
- $T_S := \bigcup_{i \in I} \{s_i\}$  and  $T_E := \bigcup_{i \in I} \{e_i\}$  as the set of all starting and ending times, respectively,
- $I_t := \{i \in I \mid t \in [s_i, e_i]\}$  as the set of all items being active at time  $t \in T$ .

We assume the items (i) to require the  $c_i$  units of capacity at any time within their time window  $[s_i, e_i]$  and (ii) to be ordered with respect to non-decreasing starting times, where ties are broken arbitrarily. The TBPP-FU then searches for a feasible assignment of these items to a set  $K := \{1, \dots, n\}$  of servers each of which having capacity  $C \in \mathbb{N}$ , so that an objective function consisting of a weighted sum of the number of used servers and the number of necessary switch-on operations, resulting from the concrete arrangement of the jobs, is minimized.

Note that the capacity represents a renewable resource at any time and that the placement of an item, in contrast to classical two-dimensional packing problems, does not have to result in a contiguous rectangle (see Fig. 1).



**Figure 1** – An assignment of eight items to one bin of size  $C = 7$ , following the basic idea from (Dell’Amico et al., 2020, Fig. 2). The blue item  $i = 8$  with  $[s_i, e_i] = [2, 5]$  and  $c_i = 2$  is not placed in a contiguous manner. Hence, this configuration would not be feasible for classical 2D packing problems.

Moreover, the scaling parameter appearing in the objective function will be referred to as  $\gamma > 0$ . For convenience, we will mostly bundle all the relevant input data in the following way:

**Definition 1.** A tuple  $E = (n, C, \mathbf{c}, \mathbf{s}, \mathbf{e}, \gamma)$ , where  $\mathbf{c}, \mathbf{s}, \mathbf{e} \in \mathbb{Z}_+^n$  collect the item sizes, starting times, and ending times, respectively, is called an instance (of the TBPP-FU).

The TBPP-FU was first mentioned in Aydın et al. (2020) along with two ILP formulations called M1 and M2. To keep the explanations short, here we just focus on M1 which is composed of the following types of binary variables:

- We use  $z_k$  to indicate whether server  $k$  is used.
- We introduce  $x_{ik}$  to state whether job  $i$  is scheduled on server  $k$ .
- We have  $w_{tk}$  to recognize whether server  $k$  is activated at time  $t$ .
- By  $y_{tk}$  we express whether server  $k$  carries some items at time  $t$ .

The specific domains of the indices appearing in the various variable types will be clarified in the optimization problems presented below. According to Aydın et al. (2020), M1 is given by:

**Model 1 (M1)**

$$z^{(1)} = \sum_{k \in K} \left( z_k + \gamma \cdot \sum_{t \in T_S} w_{tk} \right) \rightarrow \min \quad (1)$$

$$\text{s.t.} \quad y_{tk} \leq \sum_{i \in I_t} c_i x_{ik} \leq y_{tk} C, \quad k \in K, t \in T, \quad (2)$$

$$\sum_{k \in K} x_{ik} = 1, \quad i \in I, \quad (3)$$

$$x_{ik} \leq y_{s_t, k}, \quad i \in I, k \in K, \quad (4)$$

$$y_{tk} \leq z_k, \quad k \in K, t \in T, \quad (5)$$

$$y_{tk} - y_{t-1, k} \leq w_{tk}, \quad k \in K, t \in T_S, \quad (6)$$

$$x_{ik} \in \{0, 1\}, \quad i \in I, k \in K, \quad (7)$$

$$y_{tk} \in \{0, 1\}, \quad k \in K, t \in T, \quad (8)$$

$$w_{tk} \in \{0, 1\}, \quad k \in K, t \in T_S, \quad (9)$$

$$z_k \in \{0, 1\}, \quad k \in K. \quad (10)$$

The objective function (1) represents a weighted sum involving the number of servers in use and the number of fire-ups necessary for their operation. For any feasible schedule, we need to ensure that:

- the items placed on a server take capacity into account and also match the activity state specified by the  $y$ -variables (see Constraints (2)),
- every item is assigned precisely once (see Constraints (3)),
- the variable types are coupled consistently (see Constraints (4)-(6)), so that, in particular, the fire-ups are only counted when a server status changes from idle to active operation (see Constraints (6)). Note that similar constraints can also be found in the field of production planning with start-up costs (see Magnanti & Sastry (2002); Magnanti & Vachani (1990); Wolsey (1989)).

For simplicity, we use  $t-1$  to refer to the predecessor of  $t$ , and define  $y_{t-1, k} := 0$  for  $t = \min T$ , where  $\min T$  is the minimum element (w.r.t.  $<$ ) of the finite set  $T \neq \emptyset$ .

As already recommended in Aydın et al. (2020), M1 and M2, the latter of which is not discussed here in detail, can be directly improved by imposing a lower bound  $\sum_{k \in K} z_k \geq h$ , where  $h \in \mathbb{N}$  is the number of servers obtained by column generation (applied to the ordinary TBPP), and sorting the servers according to  $z_k \geq z_{k+1}$ . We particularly note that, due to (Dell'Amico et al., 2020, Property 4),

$$h \geq \max_{t \in T_S} \left\{ \left\lceil \frac{\sum_{i \in I_t} c_i}{C} \right\rceil \right\} \geq 1, \quad (11)$$

is always satisfied.

As a first main contribution, we clarify that the problem under consideration is indeed a significant extension of the TBPP (without fire-ups). In the literature, this question has so far been examined on the basis of only one observation.

**Theorem 1 (see Aydın et al. (2020)).** *Let  $E$  be an instance of the TBPP-FU. If  $\gamma \leq 1/n$  holds, then the number of servers in an optimal solution of the TBPP-FU is equal to the number of servers in an optimal solution of the TBPP. For  $\gamma > 1/n$ , this statement does not hold, in general.*

To hone the profile of the TBPP-FU as a truly independent optimization problem, let us further study the possibility to partition a (difficult) instance into two (or more) smaller subinstances, which can then be solved independently of each other. To be more precise, let us consider a scenario in which the respective partition classes of  $I$  mutually do not overlap in a temporal sense. In any such scenario, for the TBPP it is clear that an optimal solution can be determined by a concatenation of the corresponding optimal schedules of the subproblems<sup>2</sup>. In the case of the TBPP-FU, however, this is generally no longer the case, as the following new result shows.

**Theorem 2.** *Let  $E$  be an instance of the TBPP-FU with*

$$\min_{t \in [\min T_S, \max T_S]} \left\{ \sum_{i \in I_t} c_i \right\} = 0.$$

*Then, we cannot apply temporal decomposition to solve  $E$ , in general.*

*Proof.* Let us consider an instance  $E$  having capacity  $C = 3$ ,  $n = 15$  items,  $\gamma = 1$ , and the following further input data:

$$\begin{aligned} \mathbf{c} &= (3, 3, 3, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2), \\ \mathbf{s} &= (1, 1, 1, 1, 1, 1, 3, 3, 3, 5, 5, 5, 5, 5, 5), \\ \mathbf{e} &= (2, 2, 2, 4, 4, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6). \end{aligned}$$

So, in fact, there are four different item types (see also Fig. 2): three items with  $c_i = 3$  and  $[s_i, e_i] = [1, 2)$ , three items with  $c_i = 1$  and  $[s_i, e_i] = [1, 4)$ , three items with  $c_i = 2$  and  $[s_i, e_i] = [3, 4)$ , and six items with  $c_i = 2$  and  $[s_i, e_i] = [5, 6)$ .

We notice that there is no activity at time  $t = 4$ , so we could try to split the instance in two parts. The first subinstance  $E_1$  would contain all items  $i \in \{1, \dots, 9\}$ , i.e., the items that are executed before  $t = 4$ , while the second subinstance  $E_2$  collects all the remaining items  $i \in \{10, \dots, 15\}$ , i.e.,

<sup>2</sup>To some extent, this observation is linked to the construction of lower bounds, based on BPP instances with item set  $I_t$ ,  $t \in T_S$ , proposed in (Dell'Amico et al., 2020, Section 4). A straightforward extension of that construction is given by the fact that solving a TBPP for any proper subset  $I' \subset I$  also leads to a lower bound for the optimal value of the original TBPP instance. Hence, partitioning the item set  $I$  and considering the maximum TBPP value obtained from all the partition classes is still a lower bound. This lower bound becomes exact if there are no temporal overlaps between the considered partition classes.



solution of the original instance, in general. This observation even holds, if there is no activity at all in some time interval. The latter fact represents a remarkable difference to the ordinary TBPP, and underlines another challenging aspect of the TBPP-FU. In the light of the previous observations, in this article we exclusively deal with the typically more difficult case  $\gamma > 1/n$  that allows neither direct mathematical reference to the TBPP solution methods, in the sense of Theorem 1, nor any kind of temporal decomposition.

### 3 A NEW REDUCTION METHOD FOR MODEL M1

In this section we introduce a new class of valid inequalities for M1. Even though, in the meantime, this formulation has been improved by numerous proposals in Martinovic et al. (2021) and, later also, Martinovic et al. (2022), these techniques almost exclusively addressed the size of the resulting ILP models or the symmetry of the solution space. However, at least since the publication of the famous book by Nemhauser & Wolsey (1988) it became widely accepted that, in addition to these structural features, the quality of the LP bound also has a decisive influence on the size of the branch-and-bound trees arising in the solution process and, thus, also on the overall performance. As regards the TBPP-FU, only one class of valid inequalities has so far been contributed to significantly raising the LP bound (see (Martinovic et al., 2021, Inequality (18))). Many other candidates

- either did not lead to any improvement at all, see, for example, Inequalities (19)-(21) in Martinovic et al. (2021),
- or could only raise the LP bound to a vanishingly small extent, as witnessed in (Martinovic et al., 2022, Subsect. 4.4)).

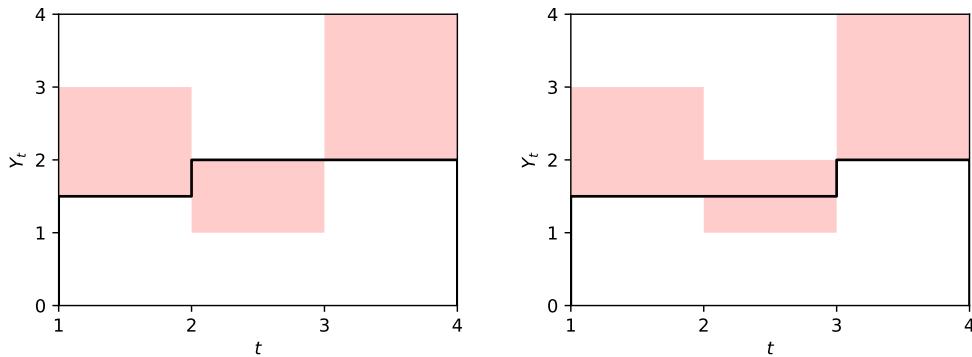
As a consequence of that observation, the new class of valid inequalities to be presented here is not just an arbitrary additional item in what has become a handsome list meanwhile, but actively contributes to bringing into focus an important aspect that has been dealt with rather insufficiently so far.

Before describing all the mathematical details, let us start with a motivating numerical example:

**Example 1.** *We consider an instance  $E$  with  $\gamma = 1$ , capacity  $C = 2$ , and  $n = 3$  items having  $\mathbf{s} = (1, 1, 3)$ ,  $\mathbf{e} = (4, 2, 4)$ , and  $\mathbf{c} = (2, 1, 2)$ . When solving the LP relaxation of M1, we obtain the optimal value  $z_{LP}^{(1),*} = 3.5$ . In the left-hand part of Fig. 3, one specific aspect of a corresponding optimal solution is depicted. To be more precise, the sum of all activity variables, that is  $Y_t := \sum_{k \in K} y_{tk}$ , is displayed over time. It can be seen that the optimal solution is able to increase the aggregated activity  $Y_t$  at the ending time  $t = 2$  of job  $i = 2$ , but this time point is not part of any  $w$ -variable in the objective function. Hence, the current model is able to hide a fraction of a fire-up. To mend this flaw, we can make use of  $w_{tk}$  and Constraints (6) for all  $t \in T$  which leads to the better LP value  $z_{LP}^{(1),*} = 4$ , because any (fraction of a) fire-up will be noticed correctly and counted by the objective function. Hence, we obtain the more natural scenario depicted in the*



right part of Fig. 3. Note that the complete data of the optimal solutions illustrated in Fig. 3 can be found in Appendix A.



**Figure 3** – The summed activity  $Y_t := \sum_{k \in K} y_{tk}$  over time in an optimal solution of  $E$ . The left picture shows the development of  $Y_t$  for Model 1, whereas the right picture visualizes the situation after having modified the model by using  $w_{tk}$  and Constraints (6) for all  $t \in T$ . In both cases, the red regions define some lower and upper bounds for  $Y_t$  that are derived from Constraints (2), (4), and (5) of M1. Even if this is not clearly visible in the illustrations, please note that the value of  $Y_4$  is zero in both scenarios.

Given the observations of Example 1, we propose the following new improvement of M1 to prevent the LP relaxation from hiding (fractional) fire-ups:

**(R0) Corrected fire-up recognition:** We introduce  $w_{tk} \in \{0, 1\}$  for all  $k \in K$  and  $t \in T$  (not just  $t \in T_S$ ), and also count all these variables in the objective function of M1. Moreover, we demand Constraints (6) for all  $k \in K$  and  $t \in T$ .

Note that we use the label (R0), since this improvement attaches directly to the original model from Aydın et al. (2020) and, on the other hand, we have to respect the fixed identifiers for the reductions already assigned in Martinovic et al. (2021) and Martinovic et al. (2022).

**Remark 2.** Even if (R0) concentrates on a correct detection of the fire-ups, it addresses both, the  $w$ - and the  $y$ -variables (see also Fig. 3). Since the latter are not present in Model M2 from Aydın et al. (2020), (R0) cannot be applied to that formulation.

**Remark 3.** As mentioned earlier, constraints of type (6) are also known from production planning and machine scheduling problems involving start up or changeover costs (see Magnanti & Sastry (2002); Magnanti & Vachani (1990); Wolsey (1989)). In these applications, production activity is typically carried out in discrete time periods, and the operator has to decide to run a machine or to shut it down for the entire period. Re-starting a machine or changing production from one item to another may lead to delays which are measured by additional costs in the objective function. In these scenarios, the time horizon consists of identical periods, and there is no difference between starting and ending times. Hence, constraints of type (6) are typically included in the associated

*ILP models for all periods right from the beginning, because set-up costs may appear at any point in time, even in the integer problem. However, for the TBPP-FU, the latter is not the case. Hence, the cuts proposed in (R0) are problem-specific and cannot be transferred to the aforementioned manufacturing problems.*

On top of raising the LP bound, adding (R0) to M1 also allows us to explicitly state a solution of the corresponding LP relaxation. To this end, we consider the so-called *aggregate relaxation*, where the variables summed up over  $k$  are central (see also Example 1). Thus, having defined  $Z := \sum_{k \in K} z_k$  as well as  $Y_t := \sum_{k \in K} y_{tk}$  and  $W_t := \sum_{k \in K} w_{tk}$  for  $t \in T$ , we obtain the

**Aggregate Relaxation of M1+(R0)**

$$z_{agg}^{(1)} = \gamma \cdot \sum_{t \in T} W_t + Z \rightarrow \min \tag{12}$$

$$\text{s.t.} \quad Y_t \leq \sum_{i \in I_t} c_i \leq Y_t \cdot C, \quad t \in T, \tag{13}$$

$$1 \leq Y_{s_i}, \quad i \in I, \tag{14}$$

$$Y_t - Y_{t-1} \leq W_t, \quad t \in T, \tag{15}$$

$$Y_t \in [0, n], \quad t \in T, \tag{16}$$

$$W_t \in [0, n], \quad t \in T, \tag{17}$$

$$Z \in [h, n], \tag{18}$$

where  $\sum_{k \in K} x_{ik} = 1, i \in I$ , was used. Note that the aggregated form of Constraints (5), i.e.,  $Y_t \leq Z$ , does not appear explicitly, because  $Z \in [h, n]$  is more restrictive due to (11) and (13). In addition, the optimal value of the LP relaxation does not change when applying this type of constraint aggregation.

**Lemma 1.** *The optimal value of the aggregate relaxation (of M1 + (R0)) is equal to the optimal value of the ordinary LP relaxation (of M1 + (R0)).*

*Proof.* First of all, we note that after having combined some of the constraints, the feasible set cannot be smaller than before, so that the aggregate relaxation cannot have a larger optimal value than the classic LP relaxation. On the other hand, any feasible point of the aggregate relaxation directly leads to a feasible point of the LP relaxation by defining

$$x_{ik} := \frac{1}{n}, \quad z_k := \frac{Z}{n}, \quad y_{tk} := \frac{Y_t}{n}, \quad w_{tk} := \frac{W_t}{n}$$

for any respective index or index pair. By that, we do not change the objective value, so that the optimal value of the aggregate relaxation cannot be smaller than that of the LP relaxation. Both observations together make sure that the claim is proved. □

Hence, we end up with an equivalent but less difficult optimization problem not containing  $k$  anymore. The solution of this new problem can be stated as follows.

**Theorem 3.** *Let  $E$  be an instance of the TBPP-FU. Then, an optimal solution of the aggregate relaxation of  $M1+(R0)$  is given by  $Z := h$  and*

$$Y_t := \begin{cases} \min \left\{ \sum_{i \in I_t} c_i, \max \left\{ Y_{t-1}, \frac{\sum_{i \in I_t} c_i}{C} \right\} \right\}, & \text{if } t \notin T_S, \\ \min \left\{ \sum_{i \in I_t} c_i, \max \left\{ 1, Y_{t-1}, \frac{\sum_{i \in I_t} c_i}{C} \right\} \right\}, & \text{if } t \in T_S, \end{cases} \quad (t \in T),$$

$$W_t := \max \{ Y_t - Y_{t-1}, 0 \}, \quad (t \in T),$$

where  $Y_t := Y_0 := 0$  for  $t = \min T$  is used in the above definitions.

*Proof.* We start by noting that  $Z = h$  has to hold due to Constraint (18). As a consequence of (15), in an optimal solution we definitely require  $W_t = \max \{ Y_t - Y_{t-1}, 0 \}$  for all  $t \in T$ . Hence, we just need to justify the choice of  $Y_t$ ,  $t \in T$ , presented above. To this end, let  $Y_0 := 0$  represent an auxiliary initial state. Then, for given  $Y_{t-1}$ , there are several aspects to be considered when defining the best possible follow-up variable  $Y_t$ :

- **local view:** To keep  $W_t$  as small as possible (due to (12)), we should always favor small values of  $Y_t$ . Hence,  $Y_t$  should be equal or close to  $\frac{\sum_{i \in I_t} c_i}{C}$  (for  $t \notin T_S$ ) and  $\max \{ 1, \frac{\sum_{i \in I_t} c_i}{C} \}$  (for  $t \in T_S$ ) as a consequence of Constraints (13) and (14).
- **global view:** Reducing the value of  $Y_t$  below that of  $Y_{t-1}$  is not required, since  $W_t$  would not change. Moreover, such unnecessary reductions of  $Y_t$  could later, say at time  $\tau \in T$ , lead to a situation where  $Y_\tau$  has to be increased too much compared to  $Y_{\tau-1}$ , so that  $W_\tau$  and the overall objective function might increase with no need. Hence, for an optimal choice of  $Y_t$  the value of  $Y_{t-1}$  should also serve as a lower bound. So far, the maximum appearing in the definition of  $Y_t$  in the theorem is justified.
- **feasibility:** Trying to keep the status  $Y_t = Y_{t-1}$ , as it is intended by the 'global view' observation, is only possible if this is allowed by the left-hand side in Constraints (13). If not, then  $Y_t$  has to fall down below  $Y_{t-1}$ , but only until it reaches the bounding value  $\sum_{i \in I_t} c_i$  imposed by that restriction. This prevents the trajectory of  $Y_t$  from making unnecessarily high jumps later in time, which would again cause too large values for some fire-up variables. This fact leads to the additional outer minimum operator in the definition of  $Y_t$  presented above.

Altogether, the point specified in the theorem is feasible and the values of  $Y_t$ ,  $t \in T$ , are chosen optimally.  $\square$

**Remark 4.** *The intention behind the item called 'global view' in the proof of Theorem 3 is also visible in the right picture in Fig. 3. At  $t = 2$ , it would be possible to reduce the total activity  $Y_t = Y_2$ , but this would lead to a larger jump at  $t = 3$ , so that the fire-up variable  $W_t = W_3$  will be larger than necessary.*

**Remark 5.** Effectively, the nested minimum and maximum appearing in the definition of  $Y_t$  in Theorem 3 can be arranged a bit more clearly.

- If  $Y_{t-1} \geq \sum_{i \in I_t} c_i$  holds, the value of  $Y_t$  is always given by  $Y_t = \sum_{i \in I_t} c_i$ . This is correct because the hypothesis would also lead to

$$Y_{t-1} \geq \sum_{i \in I_t} c_i \geq \frac{\sum_{i \in I_t} c_i}{C} \quad \text{and} \quad Y_{t-1} \geq \sum_{i \in I_t} c_i \geq 1,$$

where the latter just holds for  $t \in T_S$ , in general. However, the maximum appearing in the definition of  $Y_t$  would always be given by  $Y_{t-1}$ , which would then be beaten by the term  $\sum_{i \in I_t} c_i$  in the outer minimum.

- If  $Y_{t-1} < \sum_{i \in I_t} c_i$  holds instead, then we can at least skip the outer minimum, meaning that

$$Y_t := \begin{cases} \max \left\{ Y_{t-1}, \frac{\sum_{i \in I_t} c_i}{C} \right\}, & \text{if } t \notin T_S, \\ \max \left\{ 1, Y_{t-1}, \frac{\sum_{i \in I_t} c_i}{C} \right\}, & \text{if } t \in T_S, \end{cases}$$

is true for any  $t \in T$ . Indeed, due to

$$\frac{\sum_{i \in I_t} c_i}{C} \leq \sum_{i \in I_t} c_i \quad \text{and} \quad 1 \leq \sum_{i \in I_t} c_i,$$

where the latter at least holds for all  $t \in T_S$ , the minimum is always given by the term that has won the inner maximum before.

The procedure described in Remark 5 is relatively easy to implement so that also the LP relaxation of M1+(R0) can be solved without requiring any optimization software. However, given the rather technical terms derived before, we do not state an explicit formula for the optimal LP value here.

**Remark 6.** Similarly, the LP bound of M2 from Aydın et al. (2020) can be shown to be equal to  $z_{LP}^{(2),*} = h + \gamma \cdot g(E)$ , with  $g(E) := \left| \{t \in T_S \mid \exists i \in I : t = s_i, \delta_i^+ = \emptyset\} \right|$ . In particular, any instance with

$$\min_{t \in [\min T_S, \max T_S]} \left\{ \sum_{i \in I_t} c_i \right\} > 0,$$

i.e., any instance without mandatory “temporal gaps”, leads to an LP bound comparable to  $h$  and, thus, of rather poor quality.

The new results from Theorem 3 and Remark 6 thus extend the existing literature in that they

- contain explicit constructions and closed-form terms of the solutions to the considered LP relaxations,
- provide further insights on the relation  $z_{LP}^{(1),*} \geq z_{LP}^{(2),*}$  which was just stated on an abstract level in (Aydın et al., 2020, Proposition 4.1).

As can be clearly seen in Remark 6, the LP relaxation of M2 is only able to perceive “integer steps” in the fire-up term appearing in the objective function – but they can only appear for instances having temporal gaps. In any other case, the associated LP bound is very close to the material bound and rather weak. In contrast, the LP relaxation of M1+(R0) also collects fractional increases of the activity variables  $y_{tk}$ , which typically leads to much better bounds due to Constraints (6). By that, we provide a new theoretical justification for the poor performance of M2, closing an open question from Aydın et al. (2020).

## 4 COMPUTATIONAL TESTS

In this section, we consider the numerical effects of the proposed model improvement using various test calculations. For this purpose, we first briefly repeat the most important parameters of a benchmark set from the literature and use this to gain initial insights into the effect of (R0) on different variants of the model M1. In the second part, we then define a new set of benchmark instances that is parameterized, among other things, by the percentage of pure ending times. For these instances, the positive effects of (R0) will be particularly evident in the numerical test calculations.

### 4.1 General Information and Methodology

To discuss the numerical effects of (R0), we consider 160 benchmark instances described in (Aydın et al., 2020, Section 5), consisting of a total of 32 groups of 5 instances each. Each of these groups is uniquely described by a combination of the following four indicators:

- **Number of items:** We consider  $n \in \{50, 100, 150, 200\}$ .
- **Time horizon:** The starting times  $s_i, i \in I$ , are uniformly distributed on  $[0, \bar{s}] \cap \mathbb{Z}_+$ , where  $\bar{s} \in \{n, 1.2n\}$  either represents a rather dense ( $\bar{s} = n$ ) or a more relaxed ( $\bar{s} = 1.2n$ ) scenario with respect to possible item interactions.
- **Duration:** The *item durations*  $d_i := e_i - s_i, i \in I$ , are either chosen to be *short* (i.e.,  $d_i \in [10, 30] \cap \mathbb{Z}_+$ ) or *long* (i.e.,  $d_i \in [20, 60] \cap \mathbb{Z}_+$ ). For convenience, these constellations will be abbreviated by  $d_S$  and  $d_L$ , respectively.
- **Item sizes:** The capacity demands of the items can be *low* (i.e.,  $c_i \in [25, 50] \cap \mathbb{Z}_+$ ) or *high* (i.e.,  $c_i \in [25, 75] \cap \mathbb{Z}_+$ ). In the following, we will refer to these scenarios by  $c_L$  and  $c_H$ , respectively.

In the numerical test calculations, we will not only investigate the effects of (R0) on the original model M1 from Aydın et al. (2020), but also incorporate this strategy into two more competitive versions of M1, namely the variants M1\* and M1\*\* introduced in Martinovic et al. (2022). For the sake of a better understanding, we therefore briefly list here the changes that have occurred over time from M1 to M1\*\* adhering to the notation used in the respective references. At first, let us start with the contributions from Martinovic et al. (2021):

- (R1) To reduce symmetry, we only consider job-to-server assignments  $(i, k)$  with  $k \leq i$ . By that, we also can move to server-dependent time sets  $T(k)$  and  $T_S(k)$ .
- (R2) We introduce valid cuts  $z_k \leq \sum_{t \in T_S(k)} w_{tk}$  for any  $k \in K$  so that at least one fire-up is perceived on any server in use. By that, we establish a mutual dependency of the two variable types appearing in the objective function.
- (R3) We lift the item sizes  $c_i$ ,  $i \in I$ , to possibly raise some coefficients appearing in the set of constraints (see Dell'Amico et al. (2020)). To be more precise, we first define the sets  $Q_i := \{j \in I \setminus \{i\} \mid [s_i, e_i) \cap [s_j, e_j) \neq \emptyset\}$  for all  $i \in I$ . Then, we compute

$$\varepsilon(i) := \max \left\{ \sum_{j \in Q_i} c_j \cdot x_j \mid \sum_{j \in Q_i} c_j \cdot x_j \leq C - c_i, x_j \in \{0, 1\} \text{ for all } j \in Q_i \right\},$$

i.e., the maximum capacity required for a temporal knapsack containing job  $i$ . Whenever  $\varepsilon(i) < C - c_i$  holds,  $c_i$  can be modified to  $\tilde{c}_i := C - \varepsilon(i)$ .

On top of that, the following techniques were proposed in Martinovic et al. (2022):

- (a) We apply a set of smaller modifications to the inequality chain (2) in M1.
- (b) We limit the possible item combinations by using clique-based cuts.
- (c) We use heuristic information to massively reduce the number  $|K|$  of servers to be initialized.

All six contributions, i.e., (R1)–(R3) and (a)–(c), lead to an intermediate version of M1, called M1\* in Martinovic et al. (2022), showing convincing numerical results. However, the application of further cuts known from Wolsey (1989) for a neighboring problem even lead to a slightly more powerful ILP formulation, called M1\*\*.

In all the following experiments, we coded the formulations in Python (version 3.7.7) and applied Gurobi (version 9) with a time limit of  $t_{\max} = 30$  minutes to handle the ILP models. Moreover, we note that the computations were run on an AMD A10-5800K processor with 16 GB RAM. Given the large number of instances attempted, we typically just display average results. However, for the strongest ILP formulations, referred to as M1\*\* and M1\*\*+(R0) in the remainder of this section, the log-files produced by the ILP solver can be found at <https://github.com/wotzlauff/tbpp-cf3/tree/main/logs>. From these files, the instance-specific results can be deduced.

## 4.2 Computational Results for the Instances from Aydın et al. (2020)

First, in Table 1 we investigate the impact of the new cuts on the LP bound by listing the average LP values of three different basic configuration, namely (i) the literature version of M1 as given in Section 2, (ii) the version M1\* that has gone through all six improvement steps (see Martinovic et al. (2022)), and (iii) the currently best version M1\*\*, which emerged from M1\* by adding the

**Table 1** – An overview of the (average) LP bounds obtained from different variants of M1-type formulations.

$n$	$\bar{s}$	$d_i$	$c_i$	M1	M1+(R0)	M1*	M1*+(R0)	M1**	M1**+(R0)	
50	50	$d_S$	$c_L$	18.0	19.2	19.6	19.6	19.6	19.6	
			$c_H$	22.2	24.4	25.6	25.6	25.6	25.6	
		$d_L$	$c_L$	26.8	29.4	30.0	30.0	30.0	30.0	30.0
			$c_H$	40.0	43.8	46.0	46.0	46.0	46.0	
	60	$d_S$	$c_L$	15.4	16.6	16.8	16.8	16.8	16.8	
			$c_H$	19.6	22.0	23.6	23.6	23.6	23.6	
		$d_L$	$c_L$	27.0	28.8	29.6	29.6	29.6	29.6	
			$c_H$	35.2	39.6	41.6	41.6	41.6	41.6	
Average				25.5	28.0	29.1	29.1	29.1	29.1	
100	100	$d_S$	$c_L$	18.8	21.8	22.4	22.4	22.4	22.4	
			$c_H$	25.2	30.8	33.6	33.6	33.6	33.6	
		$d_L$	$c_L$	31.0	34.2	34.4	34.4	34.4	34.4	
			$c_H$	41.6	47.4	49.2	49.2	49.4	49.4	
	120	$d_S$	$c_L$	17.2	19.6	20.0	20.0	20.0	20.0	
			$c_H$	22.0	25.6	27.2	27.4	27.4	28.4	
		$d_L$	$c_L$	25.0	30.0	30.8	30.8	30.8	30.8	
			$c_H$	37.8	42.8	44.4	44.4	44.4	44.4	
	Average				27.3	31.5	32.8	32.8	32.8	32.9
	150	150	$d_S$	$c_L$	17.8	21.6	21.6	21.6	21.6	21.6
				$c_H$	29.2	32.0	34.8	36.0	34.8	36.8
			$d_L$	$c_L$	33.0	38.4	38.8	38.8	38.8	38.8
$c_H$				43.8	50.8	52.8	52.8	52.8	52.8	
180		$d_S$	$c_L$	15.2	19.2	19.6	19.6	19.6	19.6	
			$c_H$	21.8	26.4	28.0	31.2	28.0	32.4	
		$d_L$	$c_L$	26.6	31.0	31.6	31.6	31.6	31.6	
			$c_H$	37.6	44.6	47.6	47.6	47.6	47.6	
Average				28.1	33.0	34.4	34.9	34.4	35.1	
200		200	$d_S$	$c_L$	18.8	23.6	24.4	24.4	24.4	24.4
				$c_H$	25.2	30.0	31.6	32.4	31.6	33.2
			$d_L$	$c_L$	32.4	37.4	38.0	38.0	38.0	38.0
	$c_H$			43.4	51.4	53.6	53.6	53.6	53.6	
	240	$d_S$	$c_L$	16.0	20.6	21.2	21.2	21.6	21.6	
			$c_H$	23.8	28.2	30.6	34.6	30.6	37.0	
		$d_L$	$c_L$	27.6	31.8	32.4	32.4	32.4	32.4	
			$c_H$	36.6	43.4	46.0	46.2	46.0	46.2	
	Average				28.0	33.3	34.7	35.3	34.8	35.8
	Total: Average				27.2	31.5	32.7	33.0	32.8	33.2

constraints introduced in Wolsey (1989) (see (Martinovic et al., 2022, Subsect. 4.4)). To any of these variants, we also add (R0) resulting in a total of six versions of M1.

The main findings are:

- From the average LP values, we see that Reduction (R0) is able to improve each of the three given states of M1. The influence of (R0) is, of course, particularly large (roughly 16%) for the raw version from the literature where no other reduction is involved yet.
- Typically, with increasing number of applied reductions it becomes, on the one hand, more difficult to further improve the LP bound, but on the other hand the interaction of different types of valid inequalities may also result in an additional booster effect. For instance, the latter can be observed for the hardest subset of instances with  $n = 200$  items. Here, we see that adding (R0) to the intermediate version  $M1^*$  is able to improve the LP bound by approximately 1.7%, whereas the already more refined version  $M1^{**}$  benefits from (R0) by almost 3% better LP values.
- According to Table 1, (R0) seems to be particularly useful for the configuration  $(d_S, c_H)$ . This observation is logical, because in these scenarios the occurrence of fire-ups is typically more likely than for the other choices, due to the relatively short jobs, which at the same time have to be distributed over many servers because of their high resource demands.

Altogether, it can be concluded that applying (R0) is recommended for each variant of the M1-type formulation. Remarkably, even the LP bounds of the currently best known version  $M1^{**}$  can still benefit to a non-marginal amount from the new valid inequalities.

**Remark 7.** *In Aydın et al. (2020), further more difficult instance sets are introduced, but only treated heuristically. In fact, they cannot be solved exactly even with the improved compact models from Martinovic et al. (2021) or Martinovic et al. (2022). Nevertheless, the data collected in Table 2 underlines that the previous observations all apply also to these very large and challenging instances and, moreover, even gain more importance.*

*On average, for  $n = 500$  (R0) is able to raise the LP bound of the currently best compact formulation  $M1^{**}$  by an additional 11%, while for  $n = 1000$  the increase is already above 23%. Particularly remarkable effects of between 50% (for  $n = 500$ ) and 70% (for  $n = 1000$ ) improvement can be observed for the configuration  $(d_S, c_H)$  that typically contains many fire-ups. Thus, especially for those cases, the much better lower bounds can be used to assess the quality of heuristic solutions much more accurately than before.*



**Table 2** – An overview of the (average) LP bounds obtained from different variants of M1-type formulations for very large instances.

$n$	$\bar{s}$	$d_i$	$c_i$	M1	M1+(R0)	M1*	M1*+(R0)	M1**	M1**+(R0)	
500	500	$d_S$	$c_L$	19.6	24.0	24.8	25.4	24.8	26.0	
			$c_H$	28.2	34.0	37.0	47.0	37.6	49.2	
		$d_L$	$c_L$	32.6	40.0	40.8	40.8	40.8	41.0	
			$c_H$	44.4	55.0	59.2	59.2	59.2	59.4	
	600	$d_S$	$c_L$	16.6	21.8	22.4	23.0	22.4	25.0	
			$c_H$	22.8	28.0	30.8	46.6	32.6	49.8	
		$d_L$	$c_L$	27.8	35.2	35.6	35.6	35.6	35.6	
			$c_H$	39.4	48.0	50.0	50.8	50.0	51.0	
Average				28.9	35.8	37.6	41.0	37.9	42.1	
1000	1000	$d_S$	$c_L$	20.2	24.4	24.8	26.6	25.0	28.2	
			$c_H$	30.6	37.0	45.4	72.6	47.4	77.6	
			$d_L$	$c_L$	32.4	39.0	39.6	39.6	39.6	39.6
				$c_H$	48.0	58.6	63.2	64.6	63.2	65.2
		1200	$d_S$	$c_L$	17.0	21.4	22.0	26.0	22.0	29.2
				$c_H$	26.8	31.8	41.2	71.2	44.8	76.4
			$d_L$	$c_L$	30.6	35.0	36.0	36.0	36.0	36.0
				$c_H$	40.2	48.4	52.0	52.0	52.0	53.4
	Average				30.7	37.0	40.5	48.6	41.2	50.7

In a second experiment, we now focus on a broader range of performance metrics, but only for the currently best formulation M1\*\* presented in Martinovic et al. (2022). To this end, we solve the integer programs related to the 160 benchmark instances introduced in Subsect. 4.1 and list the results in Table 3. The abbreviations used therein for the first time are as follows

- $n_{var}$ ,  $n_{con}$ , and  $n_{nz}$  refer to the average number of variables, constraints, and nonzero coefficients, respectively, appearing in the integer program.
- $opt$  counts the number of instances solved to proven optimality, whereas  $t$  is the average time required for computation. Whenever an instance is not solved to proven optimality, the time  $t$  is set to the time limit.
- The row called 'Exit gap' measures the relative deviation of the best objective value available at the end of the time limit compared to the best lower bound, meaning that an optimally solved instance does not add anything to that row.

It can be observed that (R0) is responsible for an increase of about 8% in terms of variables and 12% in terms of constraints, so that the ILP model becomes a bit larger. Relative to these values, however, the number of non-zero elements in the constraint matrix does not increase proportionally, but to a much lesser extent. We attribute this to the fact that any of the new inequalities just contains two variables so that the rows corresponding to (R0) are very sparse.

**Table 3** – A brief numerical comparison for two versions of  $M1^{**}$ , with and without (R0). The column ‘Diff’ measures the percentage difference between the new and the old value.

indicator	units	$M1^{**}$	$M1^{**}+(R0)$	Diff
$n_{var}$	$10^3$	6.1	6.6	+8.2%
$n_{con}$	$10^3$	12.6	14.1	+11.9%
$n_{nz}$	$10^3$	110.0	114.0	+2.7%
$z_{LP}$	—	32.8	33.2	+1.2%
Exit gap	%	3.4	3.0	−11.8%
$opt$	—	104	106	+1.9%
$t$	s	705.5	703.7	−0.3%

Despite this (slightly) increased model size, however, there is a positive effect in the overall performance of the setting  $M1^{**}+(R0)$ , which again underlines the contribution of the improved lower bounds. More precisely, within the identical average computation time, two additional instances are solved to proven optimality and the average exit gap decreases by almost 12%, so that in many cases (when reaching the time limit) a better objective function value can also be expected.

A more detailed overview of which subset of instances was tackled more successfully by which of the two formulations can be found in Table B1 in Appendix B. Given the observations from our earlier discussions, the new cuts especially help to find integer optimal solutions of previously unsolved instances for the configuration  $(d_S, c_H)$ . Due to the same reasons, we sometimes have an almost opposite effect for the case of short jobs with low resource demands (scenario  $(d_S, c_L)$ ). There, the probability of fire-ups in a feasible assignment is still rather high due to the relatively short job durations, but much fewer active servers are required in an optimal solution. Thus, much denser packings are typically possible, also in the LP relaxation, which is supported by the optimal LP values in Table 1. Hence, although all instances of both tested models are solved in a comparatively short time, the variant  $M1^{**}+(R0)$  requires more than twice as much computing time in some of these cases.

### 4.3 An Outlook: The Impact of (R0) on Further Instance Classes

Although we were able to identify some beneficial aspects of the new valid cuts (R0), the benchmark sets given in the literature are not sufficient to fully cover the impact of that type of model improvement.

**Remark 8.** *Apart from the instances already considered, there is just one other relevant benchmark set which has been proposed in Dell’Amico et al. (2020) in the context of the classical TBPP. However, these instances are characterized by  $|T| = |T_S| + 1$ , basically meaning that each ending time is the starting time of another job. So, in fact, there is no difference at all between the models  $M1^{**}$  and  $M1^{**}+(R0)$ .*

Hence, to observe some more pronounced effects of (R0), new sets of benchmark instances have to be designed. In doing so, we mainly focus on an aspect that has not been part of the numerical studies so far. More precisely, in the light of Remark 8, a measure related to the “percentage” of pure ending times seems to be a promising additional parameter in the construction of instance sets. By “pure ending time”, we mean that the time point considered does not represent the starting time of another job. To this end, we basically overtook the ideas applied to build the instances from Aydın et al. (2020), but post-processed them in a way that the temporal characteristics of every single instance are consistent with a given value  $r \in [0, 1)$  defined by

$$r := \frac{|\{i \in I : s_i \in T_E\}|}{n}.$$

Observe that this definition of the parameter  $r$  is just one possibility to quantify the aforementioned percentage of pure ending times. More precisely,  $r = 0$  means that there is no intersection between starting and ending times, whereas  $r \rightarrow 1$  leads to scenarios comparable to those of Dell’Amico et al. (2020) described in Remark 8. Since the two models to be compared are (almost) identical for  $r \rightarrow 1$ , here we just consider the values  $r \in \{0, 0.25, 0.5, 0.75\}$ . Moreover, we restrict our experiment to  $n \in \{100, 150, 200\}$  and the setups with short jobs ( $d_S$ ), according to the lessons learned in Subsect. 4.2. For any pair  $(n, r)$ , a set of 20 instances, derived from the original instances appearing in Aydın et al. (2020), will be investigated. To focus on the general trends, we usually do not specify the various indicators (like  $d_S$  or  $c_L$ ) in the following tables, but include them in the discussion if necessary.

**Remark 9.** For completeness, we mention that the  $r$  value for the original data set from Aydın et al. (2020) averages 0.54, if the same classes  $n \in \{100, 150, 200\}$  and  $d_S$  are considered. Note that, in contrast to our newly generated instances, this average value is not a constant one for any single instance. Instead, the instance-specific value ranges from 0.39 to 0.66, so that the original set of instances is sufficiently different from the scenario  $r = 0.5$  in our new experiment.

In a first experiment, we study the development of the average LP values (see Table 4).

**Table 4** – The average LP bound obtained from M1\*\* and M1\*\*+(R0). By ‘Gap’, here we mean the relative discrepancy between the two LP values (transformed into a percentage scale).

The averages are built based on 20 instances each.

$n$	$r = 0.00$		$r = 0.25$		$r = 0.50$		$r = 0.75$	
	M1**	M1**+(R0)	M1**	M1**+(R0)	M1**	M1**+(R0)	M1**	M1**+(R0)
100	25.9	26.6	25.9	26.4	25.9	26.1	25.9	26.0
150	25.9	27.9	25.9	27.9	26.0	27.6	26.2	27.3
200	27.0	30.3	27.0	29.6	27.0	29.2	27.3	28.8
Aver.	26.2	28.3	26.3	27.9	26.3	27.6	26.5	27.3
Gap	8.0%		6.0%		4.6%		3.0%	

The main observations can be summarized as follows:

- On average, we see that the difference between the two LP bounds decreases if  $r$  grows. More precisely, the percentage gap goes down from 8.0% (for  $r = 0.00$ ) to 3% (for  $r = 0.75$ ). Note that, for small values of  $r$ , a relatively large number of additional cuts (of type (R0)) is added to  $M1^{**}$ , since there are many ending times not representing the starting time of another job. Of course, adding many inequalities considerably restricts the solution space and, thus, increases the probability to obtain better LP values.
- We also see that, with increasing value of  $r$ , the average LP value of  $M1^{**}$  increases, whereas the opposite trend can be observed for the improved model. The former is related to the fact that hiding a fire-up is more and more restricted (if  $r$  grows), so that better LP values have to be expected. On the other hand, instances with small values of  $r$  naturally possess more fire-ups in an optimal solution since many ending times do not represent the starting time of another job. By that, the LP value typically decreases (if  $r$  grows), unless the fire-ups can be saved from being perceived in the objective function (like in  $M1^{**}$ ).

A more detailed look at the individual results (not tabulated here) reveals that the LP value of  $M1^{**}+(R0)$  is typically strictly better than that of  $M1^{**}$  in most of the constellations  $(d_S, c_H)$ . For  $(d_S, c_L)$ , a similar effect can only be observed for small values of  $r$ , i.e., scenarios where many cuts are added.

Let us now study the numerical behavior of the two models when solving the integer problems associated to the given instances. The average results can be found in Table 5 and Table 6. Among others, we highlight the following observations:

- For any choice of  $r$ , the new model is able to solve at least 3 (out of 60) additional instances. In total, the number of additionally solved instances sums up to 22. This represents an increase of 15.5% compared to  $M1^{**}$ . This positive trend is also supported by a constantly better exit gap for  $M1^{**}+(R0)$  in any scenario.
- In general, the lower the value  $r$  the fewer instances can be solved to proven optimality by either of the models. Note that small values of  $r$  correspond to large sets  $T$  (since starting and ending times highly differ from each other) leading to large integer programs to be solved. Hence, we see the lowest numbers of optimal solutions for  $r = 0$ . Similarly, the decreasing average solution time (for increasing values of  $r$ ) can be justified.
- Interestingly, the largest benefits of (R0) can be observed for  $r = 0.25$ . We attribute this to a trade-off between two conflicting trends. First, for large choices of  $r$ , the two ILP models become more and more similar. On the other hand, for very small values of  $r$  the ILP models are more likely to be too large in size, so that many instances can neither be solved in the original nor in the improved setup. In our experiments, it seems that for  $r = 0.25$  the new cuts help to obtain tractable formulations in many cases. To some extent,

this empirical observation together with Remark 9 may also explain why there was less success for the original data set from Aydın et al. (2020).

**Table 5** – Average solution times and number of instances solved to optimality (in brackets) for  $M1^{**}$  and  $M1^{**}+(R0)$  where  $r \in \{0, 0.25\}$ . Moreover, the average exit gap  $(UB - LB)/LB$  given by the solver at the end of the time limit is reported. The averages are built based on 20 instances each.

$n$	$r = 0.00$				$r = 0.25$			
	$M1^{**}$		$M1^{**}+(R0)$		$M1^{**}$		$M1^{**}+(R0)$	
100	870.9	(11)	774.3	(12)	831.2	(12)	467.5	(18)
150	976.9	(10)	1013.5	(9)	902.4	(11)	868.6	(11)
200	1201.6	(7)	1042.7	(10)	1192.7	(7)	915.0	(10)
Total	1016.5	(28)	943.5	(31)	975.4	(30)	750.4	(39)
Exit Gap	14.7%		11.1%		6.1%		3.8%	

**Table 6** – Average solution times and number of instances solved to optimality (in brackets) for  $M1^{**}$  and  $M1^{**}+(R0)$  where  $r \in \{0.5, 0.75\}$ . Moreover, the average exit gap  $(UB - LB)/LB$  given by the solver at the end of the time limit is reported. The averages are built based on 20 instances each.

$n$	$r = 0.50$				$r = 0.75$			
	$M1^{**}$		$M1^{**}+(R0)$		$M1^{**}$		$M1^{**}+(R0)$	
100	288.1	(18)	91.0	(20)	142.4	(19)	39.7	(20)
150	773.2	(12)	662.7	(14)	612.0	(16)	396.5	(17)
200	1081.6	(9)	910.7	(10)	918.9	(10)	809.8	(13)
Total	714.3	(39)	554.8	(44)	557.7	(45)	415.3	(50)
Exit Gap	6.1%		3.8%		8.2%		5.7%	

Altogether, (R0) is able to boost the performance of  $M1^{**}$  to a more pronounced extent for the new benchmark instances discussed in this subsection. In addition to the fact that short jobs turn out to favor the success of (R0), we also identified another impactful parameter of an instance that was not dealt with before in the literature. More precisely, since (R0) is meant to avoid hidden fire-ups at ending times, the percentage of pure ending times (here quantified by  $r$ ) plays an important role for the extent, the new valid cuts can improve the computational behavior of  $M1^{**}$ . While there is an obvious contribution of (R0) for any choice of  $r$  considered, its impact is specifically high when  $r \approx 0.25$  holds for a given instance.

## 5 CONCLUSIONS

In this article we have dealt with selected theoretical and numerical aspects of the TBPP-FU, which have so far received only little attention in the relevant literature. First, we succeeded in disproving the validity of temporal decompositions for the TBPP-FU, thus working out a significant difference of both optimization problems and establishing the TBPP-FU as an independent problem in cutting and packing. Furthermore, a new class of valid inequalities was proposed

leading to improved versions of M1-type formulations, also according to our test calculations. In particular, these additional cuts contribute to a noticeable increase of the LP bound even when other reduction methods have already been applied. From a theoretical point of view, this also paves the way to theoretically analyze the relationship of the LP bounds of the compact models M1 and M2 from the literature, thus extending a previously only very abstract result from Aydın et al. (2020). In an outlook section, we also introduced a new set of benchmark instances, involving a parameter controlling the percentage of pure ending times. For these instances, the effect of (R0) was even more pronounced, and we were able to identify some general features of instances supporting the success of (R0).

For the future, it is of great interest to see whether the insights into the structure of the LP solution can be extended to the improved models from Martinovic et al. (2021) and Martinovic et al. (2022) as well. Furthermore, it would be desirable to be able to specify the optimal value as precisely as possible even for the very large instances with  $n \in \{500, 1000\}$ . This requires both good heuristics and strong bounds for evaluating the approximate solutions thus obtained. As we have seen in particular in Remark 7, the latter could be clearly promoted, especially for some instance parameters, in the context of this work.

## Declarations

Funding: Not applicable. Conflicts of interest: The authors declare that they do not have any conflicts of interest. Availability of data and material: The instances used in Subsect. 4.2 were originally designed in Aydın et al. (2020) and can be found online, see <https://github.com/sibirbil/TemporalBinPacking>. The new instances introduced in Subsect. 4.3 have been gathered together as set “A1r” at <https://github.com/wotzlaff/tbpbp-instances>. For the strongest ILP models (i.e., M1\*\* and M1\*\*+(R0)), the log-files used to build Table 1, Table 3, and Table B1 can be found at <https://github.com/wotzlaff/tbpbp-cf3/tree/main/logs/mip>. In addition, the log-files used to build Table 4, Table 5, and Table 6 can be found at [https://github.com/wotzlaff/tbpbp-cf3/tree/main/logs/mip\\_r](https://github.com/wotzlaff/tbpbp-cf3/tree/main/logs/mip_r). With the help of these data, the instance-specific results can be deduced.

## References

- ANDRAE ASG & EDLER T. 2015. On global electricity usage of communication technology: trends to 2030. *Challenges*, **6**(1): 117–157.
- AYDIN N, MUTER İ & BIRBİL Ş. 2020. Multi-objective temporal bin packing problem: An application in cloud computing. *Computers & Operations Research*, **121**: 104959.
- BARTLETT M, FRISCH AM, HAMADI Y, MIGUEL I, TARIM SA & UNSWORTH C. 2005. The temporal knapsack problem and its solution. In: *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems: Second International Conference, CPAIOR 2005, Prague, Czech Republic, May 31-June 1, 2005*. pp. 34–48. Springer.

CACCHIANI V, IORI M, LOCATELLI A & MARTELLO S. 2022a. Knapsack problems – An overview of recent advances. Part I: Single knapsack problems. *Computers & Operations Research*, **143**: 105692.

CACCHIANI V, IORI M, LOCATELLI A & MARTELLO S. 2022b. Knapsack problems – An overview of recent advances. Part II: Multiple, multidimensional, and quadratic knapsack problems. *Computers & Operations Research*, **143**: 105693.

CAPRARA A, FURINI F & MALAGUTI E. 2013. Uncommon Dantzig-Wolfe reformulation for the temporal knapsack problem. *INFORMS Journal on Computing*, **25**(3): 560–571.

CAPRARA A, FURINI F, MALAGUTI E & TRAVERSI E. 2016. Solving the temporal knapsack problem via recursive Dantzig–Wolfe reformulation. *Information Processing Letters*, **116**(5): 379–386.

COFFMAN JR EG, GAREY MR & JOHNSON DS. 1978. An Application of Bin-Packing to Multiprocessor Scheduling. *SIAM Journal on Computing*, **7**(1): 1–17.

COHEN MC, KELLER PW, MIRROKNI V & ZADIMOGHADDAM M. 2019. Overcommitment in cloud services: Bin packing with chance constraints. *Management Science*, **65**(7): 3255–3271.

CÔTÉ JF, DELL’AMICO M & IORI M. 2014. Combinatorial Benders’ cuts for the strip packing problem. *Operations Research*, **62**(3): 643–661.

DE CAUWER M, MEHTA D & O’SULLIVAN B. 2016. The temporal bin packing problem: an application to workload management in data centres. In: *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*. pp. 157–164. IEEE.

DELL’AMICO M, FURINI F & IORI M. 2020. A branch-and-price algorithm for the temporal bin packing problem. *Computers & operations research*, **114**: 104825.

DELORME M, IORI M & MARTELLO S. 2016. Bin packing and cutting stock problems: Mathematical models and exact algorithms. *European Journal of Operational Research*, **255**(1): 1–20.

FETTWEIS G, DÖRPINGHAUS M, CASTRILLON J, KUMAR A, BAIER C, BOCK K, ELLINGER F, FERY A, FITZEK F, HÄRTIG H, JAMSHIDI K, KISSINGER T, LEHNER W, MERTIG M, NAGEL W, NGUYEN GT, PLETTEMEIER D, SCHRÖTER M & STRUFE T. 2019. Architecture and advanced electronics pathways toward highly adaptive energy-efficient computing. *Proceedings of the IEEE*, **107**(1): 204–231.

IORI M, DE LIMA VL, MARTELLO S, MIYAZAWA FK & MONACI M. 2021. Exact solution techniques for two-dimensional cutting and packing. *European Journal of Operational Research*, **289**(2): 399–415.

JONES N. 2018. How to stop data centres from gobbling up the world’s electricity. *Nature*, **561**(7722): 163–166.

LOPEZ-PIRES F & BARAN B. 2015. Virtual machine placement literature review. *arXiv preprint arXiv:1506.01509*, .

MAGNANTI TL & SASTRY T. 2002. Facets and reformulations for solving production planning with changeover costs. *Operations Research*, **50**(4): 708–719.

MAGNANTI TL & VACHANI R. 1990. A strong cutting plane algorithm for production scheduling with changeover costs. *Operations Research*, **38**(3): 456–473.

MARTINOVIC J & SELCH M. 2021. Mathematical models and approximate solution approaches for the stochastic bin packing problem. *Computers & Operations Research*, **135**: 105439.

MARTINOVIC J, STRASDAT N & SELCH M. 2021. Compact integer linear programming formulations for the temporal bin packing problem with fire-ups. *Computers & Operations Research*, **132**: 105288.

MARTINOVIC J, STRASDAT N, VALÉRIO DE CARVALHO JM & FURINI F. 2022. Variable and constraint reduction techniques for the temporal bin packing problem with fire-ups. *Optimization Letters*, **16**: 2333–2358.

NEMHAUSER GL & WOLSEY LA. 1988. *Integer and combinatorial optimization*. John Wiley & Sons.

SCHEITHAUER G. 2018. *Introduction to cutting and packing optimization: Problems, modeling approaches, solution methods*. vol. 263. Springer.

VALÉRIO DE CARVALHO JM. 2002. LP models for bin packing and cutting stock problems. *European Journal of Operational Research*, **141**(2): 253–273.

WANG M, MENG X & ZHANG L. 2011. Consolidating virtual machines with dynamic bandwidth demand in data centers. In: *2011 Proceedings IEEE INFOCOM*. pp. 71–75. IEEE.

WOLSEY LA. 1989. Uncapacitated lot-sizing problems with start-up costs. *Operations Research*, **37**(5): 741–747.

### How to cite

MARTINOVIC J & STRASDAT N. 2024. Theoretical Insights and a New Class of Valid Inequalities for the Temporal Bin Packing Problem with Fire-Ups. *Pesquisa Operacional*, **44**: e283503. doi: 10.1590/0101-7438.2023.043.00283503.



## APPENDIX A DETAILED SOLUTION FOR EXAMPLE 1

- **Model M1:** Optimal value  $z_{LP}^{(1),*} = 3.5$  given by  $z_1 = z_2 = 1$  and

$x_{ik}$	1	2	3
1	0.5	0.5	0
2	1	0	0
3	0.5	0.5	0

$y_{tk}$	1	2	3
1	1	0.5	0
2	1	1	0
3	1	1	0
4	0	0	0

$w_{tk}$	1	2	3
1	1	0.5	0
3	0	0	0

- **Model M1+(R0):** Optimal value  $z_{LP}^{(1),*} = 4$  given by  $z_1 = z_2 = 1$  and

$x_{ik}$	1	2	3
1	0.5	0.5	0
2	1	0	0
3	0.5	0.5	0

$y_{tk}$	1	2	3
1	1	1	0
2	1	1	0
3	1	1	0
4	0	0	0

$w_{tk}$	1	2	3
1	1	1	0
2	0	0	0
3	0	0	0
4	0	0	0

In M1, there are no  $w$ -variables for  $t = 2$ , so the increase of  $y_{tk}$  on server  $k = 2$  cannot be detected. In M1+(R0), this inconsistency does not appear anymore. From an overall point of view, note that the two solutions depicted succeed in not shutting down the servers (i.e., in saving fire-ups) due to the fractional  $x$ -variables and the left-hand side of Constraints (2).

## APPENDIX B DETAILED NUMERICAL RESULTS

**Table B1** – An overall comparison of M1\*\* and M1\*\*+(R0). We use boldface to indicate the better formulation (with respect to instances solved to proven optimality, breaking ties by the smaller required computation time).

<i>n</i>	$\bar{s}$	<i>d<sub>i</sub></i>	<i>c<sub>i</sub></i>	M1**		M1**+(R0)		
				t	opt	t	opt	
50	50	<i>d<sub>S</sub></i>	<i>c<sub>L</sub></i>	2.3	(5)	4.0	(5)	
			<i>c<sub>H</sub></i>	2.5	(5)	1.3	(5)	
		<i>d<sub>L</sub></i>	<i>c<sub>L</sub></i>	360.3	(4)	360.8	(4)	
			<i>c<sub>H</sub></i>	0.4	(5)	0.5	(5)	
	60	<i>d<sub>S</sub></i>	<i>c<sub>L</sub></i>	1.7	(5)	1.2	(5)	
			<i>c<sub>H</sub></i>	1.8	(5)	1.7	(5)	
		<i>d<sub>L</sub></i>	<i>c<sub>L</sub></i>	1.6	(5)	11.2	(5)	
			<i>c<sub>H</sub></i>	1.3	(5)	0.6	(5)	
Average (Sum)				<b>46.5</b>	<b>(39)</b>	47.7	(39)	
100	100	<i>d<sub>S</sub></i>	<i>c<sub>L</sub></i>	1.1	(5)	3.6	(5)	
			<i>c<sub>H</sub></i>	120.3	(5)	78.6	(5)	
		<i>d<sub>L</sub></i>	<i>c<sub>L</sub></i>	1442.7	(1)	1449.8	(1)	
			<i>c<sub>H</sub></i>	1090.1	(2)	1092.9	(2)	
		120	<i>d<sub>S</sub></i>	<i>c<sub>L</sub></i>	69.2	(5)	85.3	(5)
				<i>c<sub>H</sub></i>	474.8	(4)	83.6	(5)
	<i>d<sub>L</sub></i>		<i>c<sub>L</sub></i>	433.2	(4)	685.9	(4)	
			<i>c<sub>H</sub></i>	628.1	(4)	546.9	(4)	
	Average (Sum)				532.4	(30)	<b>503.3</b>	<b>(31)</b>
	150	150	<i>d<sub>S</sub></i>	<i>c<sub>L</sub></i>	49.8	(5)	85.7	(5)
				<i>c<sub>H</sub></i>	1722.5	(1)	1464.1	(1)
			<i>d<sub>L</sub></i>	<i>c<sub>L</sub></i>	1182.8	(2)	1462.2	(1)
<i>c<sub>H</sub></i>				1485.7	(2)	1372.0	(2)	
180			<i>d<sub>S</sub></i>	<i>c<sub>L</sub></i>	15.1	(5)	37.2	(5)
				<i>c<sub>H</sub></i>	939.8	(3)	853.9	(3)
		<i>d<sub>L</sub></i>	<i>c<sub>L</sub></i>	1408.5	(2)	1198.4	(3)	
			<i>c<sub>H</sub></i>	1509.8	(1)	1494.0	(1)	
Average (Sum)				1039.2	(21)	<b>995.9</b>	<b>(21)</b>	
200		200	<i>d<sub>S</sub></i>	<i>c<sub>L</sub></i>	44.4	(5)	99.2	(5)
				<i>c<sub>H</sub></i>	1800.0	(0)	1800.0	(0)
			<i>d<sub>L</sub></i>	<i>c<sub>L</sub></i>	1494.2	(1)	1800.0	(0)
	<i>c<sub>H</sub></i>			1496.1	(1)	1624.6	(1)	
	240		<i>d<sub>S</sub></i>	<i>c<sub>L</sub></i>	83.6	(5)	123.1	(5)
				<i>c<sub>H</sub></i>	1800.0	(0)	1693.9	(2)
		<i>d<sub>L</sub></i>	<i>c<sub>L</sub></i>	1110.7	(2)	1201.9	(2)	
			<i>c<sub>H</sub></i>	1800.0	(0)	1800.0	(0)	
	Average (Sum)				1203.6	(14)	<b>1267.8</b>	<b>(15)</b>
	Total: Average (Sum)				705.5	(104)	<b>703.7</b>	<b>(106)</b>