Als Manuskript gedruckt

Technische Universität Dresden
Herausgeber: Der Rektor

Theoretical Insights and a New Class of Valid Inequalities for the Temporal Bin Packing Problem with Fire-Ups

John Martinovic, Nico Strasdat

Preprint MATH-NM-01-2022

February 2022

Theoretical Insights and a New Class of Valid Inequalities for the Temporal Bin Packing Problem with Fire-Ups

J. Martinovic[a,*], N. Strasdat[a]

[a] *Institute of Numerical Mathematics, Technische Universität Dresden, Germany*

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 and therefore difficult to handle, 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 test calculations based on benchmark instances. Remarkably, the new cuts also lead to theoretical results about the optimal value of the LP relaxation.

Keywords: Cutting and Packing, Temporal Bin Packing Problem, Fire-Ups, Valid Inequalities, Decomposition

1. Introduction

In view of the steadily growing energy demands of data centers in the IT industry the low-resource management of the related server clusters has become a central concern in computer science and electrical engineering in the recent past [1, 14]. In addition to concrete technical improvements, such as in the hardware components involved [12], operations research methods to cope with the typically challenging problem of, in a sense, optimally assigning jobs to computing units are successively acquiring importance. Originating from early contributions to multiprocessor scheduling in the 1970s, see [7], a wide variety of new application areas for discrete optimization have emerged as a result of these developments, see [15] for a well-structured overview. Many of these problems are somewhat 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 [11, 20, 21]. Among a large number of possible generalizations, in particular the *stochastic BPP (SBPP)*, see [6, 16, 22], and the *temporal BPP (TBPP)*, see [9, 10], have been identified as more application-oriented descriptions of the real-world scenario. While the former takes into account the generally imperfect information about the expected jobs, the latter primarily introduces an additional time dimension.

In the context of this article, we will deal exclusively with the temporal extension, since the

*Corresponding author

Email addresses: john.martinovic@tu-dresden.de (J. Martinovic), nico.strasdat@tu-dresden.de (N. Strasdat)

SBPP has so far largely eluded efficient numerical treatment by exact approaches and is thus considered to be particularly challenging, see [16]. In the TBPP, however, the given items can be illustrated as perfectly known rectangles in a capacity-time plane, i.e., each item is characterized by a resource consumption that occurs exclusively during a defined time interval. These 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, see also Fig. 1 for a visualization. It should be noted that the capacity of the servers represents a renewable resource, so the allocation of an item does not necessarily have to be done as a contiguous rectangle[1].

The TBPP was first described in more detail in a specific application from the field of computer science, see [9], and since then it has been studied in the context of only one purely mathematical contribution [10]. 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, 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, the temporal knapsack problem [3, 4, 5], which is in a sense a single-server variant of the TBPP, and classical two-dimensional packing problems, such as the strip packing problem [8, 13], which, unlike the TBPP, (a) do not fix the possible allocation points in one dimension, and (b) mandatorily preserve the rectangular shape of the items when they are placed. Consequently, the respective models and solution methods do not provide any significant benefits for the TBPP, so that the latter actually constitutes an independent field of research.

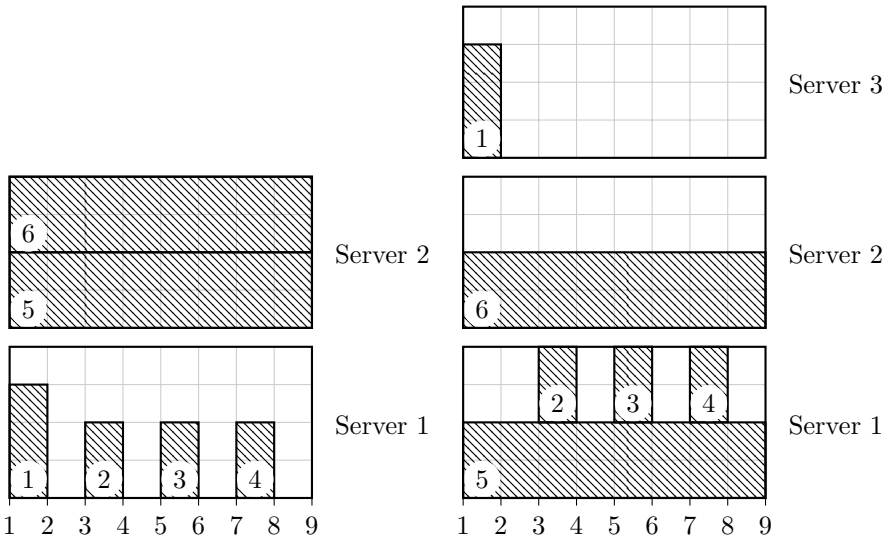

Figure 1: Two possible ways to feasibly schedule six jobs to given servers.

While, in the TBPP, a job-to-server assignment is evaluated only on the basis of the number of servers required, more recent publications suggest that the specific operating mode of these executing units should also be taken into account, especially by limiting energy-intensive repeated switching on and off processes. In [2] the authors therefore consider an extended problem statement, which additionally models the fire-ups resulting from a concrete server schedule and attach them (weighted with a parameter) to the objective function. It turns out that this so-called *temporal bin packing problem with fire-ups (TBPP-FU)* generally leads to very large in-

[1] A more detailed explanation of this assumption is contained in Fig. 2, but we would like to preface it with the introductory notation in Sect. 2 for better understanding.

teger problems having many symmetric solutions and, moreover, (for practically relevant choices of the scaling parameter) 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 ILP models and, thus, also to improve their computational performance. More precisely, given that the ILP models M1 and M2, introduced in [2], could not even successfully cope with half of the benchmark instances in a reasonable time, some significant progress has been made in this regard by the reduction methods proposed in [17] and [18] (which we will discuss in later sections of the article). However, the literature to date does not convincingly elaborate on the extent to which the two temporal problems presented earlier (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 obtained in this work:

- We give a new justification for the challenging nature of the TBPP-FU (compared to the ordinary TBPP) by showing that, typically, an instance cannot be decomposed (in a temporal sense) into subinstances of smaller size. ($\rightarrow$ Sect. 2)
- We present a new model improvement that facilitates the correct recognition of fire-ups in the case of continuous variables and thus contributes to better LP bounds. By that, we address an important issue of integer programming that has not been sufficiently covered yet in the context of the TBPP-FU. ($\rightarrow$ Sect. 3)
- This class of valid inequalities also allows to constructively state a solution of the continuous relaxation. This is an advantage over the only known result [2, Proposition 4.1] in the literature and provides further insights, especially into the relations between the LP relaxations of the models M1 and M2. ($\rightarrow$ Sect. 3)
- Based on computational tests with benchmark instances we show the numerical benefits of the additional cuts for different versions of M1-type formulations. ($\rightarrow$ Sect. 4)

2. Preliminaries and Basic Models

We consider a set of $n \in \mathbb{N}$ *items* (also referred to as *jobs*) that are specified by

- a *resource demand* $c_i \in \mathbb{N}$,
- an *activity interval* $[s_i, e_i)$ with $s_i < e_i$ denoting the *starting time* and *ending time*, respectively.

We define the set of items as $I := \{1, \ldots, n\}$, together with:

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

We assume an item to require the c_i units exclusively within the time window assigned to it. Without loss of generality, we further assume the items to be ordered with respect to non-decreasing starting times (where ties can be broken in an arbitrary way). The TBPP-FU then searches for an allocation of these items to a set $K := \{1, \ldots, n\}$ of servers each of which having capacity $C \in \mathbb{N}$, so that

i) the capacity of the used servers is not exceeded at any time,

ii) 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. 2. Moreover, the scaling parameter appearing in the objective function will be referred to as $\gamma > 0$. For the sake of convenience, we will mostly bundle all the relevant input data in the following way:

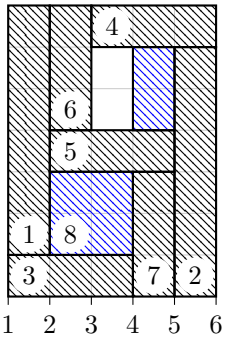

Figure 2: An assignment of eight items to one bin of size $C = 7$, following the basic idea from [10, Fig. 2]. The blue item $i = 8$ with $[s_i, e_i) = [2, 5)$ and $c_i = 2$ is not placed in a connected manner. Hence, this configuration would not be feasible for classical two-dimensional packing problems.

Definition 1. *A tuple $E = (n, C, \boldsymbol{c}, \boldsymbol{s}, \boldsymbol{e}, \gamma)$, where (in addition to the already known objects) $\boldsymbol{c}, \boldsymbol{s}$, and $\boldsymbol{e}$ are n-dimensional integer vectors collecting the item-specific input-data, is called an instance (of the TBPP-FU).*

The TBPP-FU was first mentioned in [2], where it was presented along with two ILP formulations (called M1 and M2) that are principally based on classical assignment variables. Effectively, the two models are 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 to 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.

Note that the first three sets of variables appear in both models, whereas the last is exclusive to model M1. The specific domains of the indices appearing in the various variable types will be clarified in the optimization problems presented below. According to [2], M1 is given by:

Model 1 (M1)

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

$$\text{s.t.} \quad y_{tk} \leq \sum_{i \in I_t} c_i x_{ik} \leq y_{tk} C, \qquad k \in K, t \in T, \tag{1}$$

$$\sum_{k \in K} x_{ik} = 1, \qquad i \in I, \tag{2}$$

$$x_{ik} \leq y_{s_i,k}, \qquad i \in I, k \in K, \tag{3}$$

$$y_{tk} \leq z_k, \qquad k \in K, t \in T, \tag{4}$$
$$y_{tk} - y_{t-1,k} \leq w_{tk}, \qquad k \in K, t \in T_S, \tag{5}$$
$$x_{ik} \in \{0,1\}, \qquad i \in I, k \in K, \tag{6}$$
$$y_{tk} \in \{0,1\}, \qquad k \in K, t \in T, \tag{7}$$
$$w_{tk} \in \{0,1\}, \qquad k \in K, t \in T_S, \tag{8}$$
$$z_k \in \{0,1\}, \qquad k \in K. \tag{9}$$

Obviously, the objective function 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 (1),
- every item is assigned precisely once, see (2),
- the variable types are coupled reasonably, see (3)-(5), so that, in particular, the fire-ups are only counted when a server status changes from idle to active operation, see (5).

For simplicity, we use $t-1$ to refer to the predecessor of t, and define $y_{t-1,k} := 0$ for $t = \min T$.

To define the second model, called M2, we require the sets $\delta_i := \{j < i \,|\, s_i < e_j\}$ and $\delta_i^+ := \{j < i \,|\, s_i \leq e_j\}$, $i \in I$. Note that the first one collects all jobs running parallel to job i (for at least one time step $[t, t+1)$), whereas the latter also adds those jobs that just have finished at s_i. As introduced in [2], these ingredients now lead to model M2:

Model 2 (M2)

$$z^{(2)} = \sum_{k \in K} \left(z_k + \gamma \cdot \sum_{t \in T_S} w_{tk} \right) \to \min$$

s.t.
$$\sum_{j \in \delta_i \cup \{i\}} c_j x_{jk} \leq C z_k, \qquad i \in I, k \in K, \tag{10}$$
$$\sum_{k \in K} x_{ik} = 1, \qquad i \in I, \tag{11}$$
$$x_{ik} \leq z_k, \qquad i \in I, k \in K, \tag{12}$$
$$w_{s_i,k} \geq x_{ik} - \sum_{j \in \delta_i^+} x_{jk}, \qquad i \in I, k \in K, \tag{13}$$
$$x_{ik} \in \{0,1\}, \qquad i \in I, k \in K, \tag{14}$$
$$w_{tk} \in \{0,1\}, \qquad t \in T_S, k \in K, \tag{15}$$
$$z_k \in \{0,1\}, \qquad k \in K. \tag{16}$$

Note that some terms already appeared in precisely the same (or at least a closely related) manner in M1, so that we just want to highlight the fact that the fire-up recognition is now handled differently by (13). To be more precise, a fire-up is caused by item i at time s_i (on some server k), whenever i itself but no item from δ_i^+ is packed.

As already recommended in the original publication [2], in the following we would like to understand under M1 and M2 the previously presented ILP formulations, but including two very simple techniques to slightly improve the LP relaxation. These are the specification of 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 a sorting of the servers according to $z_k \geq z_{k+1}$. We particularly note that, due to [10, Property 4],

$$h \geq \max_{t \in T_S} \left\{ \left\lceil \frac{\sum_{i \in I_t} c_i}{C} \right\rceil \right\} \geq 1, \tag{17}$$

is always satisfied.

Remark 1. *In fact, the literature also contains a model M3 whose fundamental difference to the approaches presented here is that the indices of the assignment variables are based on a job-to-job relationship. Even though this model, introduced in [17], initially showed promising numerical properties, the later contributions in [18] were able to transfer them to a large extent to M1 and M2 as well, so that from today's point of view especially these are of scientific relevance. For this reason, we would like to deal exclusively with the two previously mentioned formulations in the context of this article.*

Before we now proceed to a brief discussion of possible reduction methods, we would like to clarify that the problem under consideration is indeed a significant extension of the TBPP (without fire-ups). Indeed, the TBPP-FU has evolved in an obvious way from the ordinary TBPP by adding a second objective and can therefore be considered as an independent optimization problem only if fundamental theoretical properties, which may also concern the solution process, have been lost as a consequence of this generalization. In the literature, this question has so far been examined in more detail on the basis of only one criterion, that is, the relationship between the solution sets of the two problems.

Theorem 2 (see [2])**.** *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.*

Especially the second part of this result shows that the solution sets can be different, at least for the more relevant case that the scaling parameter does not become too small. Although this is probably one of the most significant indicators for the disparity of two problem classes, optimization problems with formally different solution sets can, nevertheless, reveal a decent amount of theoretical similarities in their solution processes[2]. To hone the profile of the TBPP-FU as a truly independent optimization problem, we would therefore like to highlight an important property of the TBPP, which is generally no longer guaranteed after the transition to the TBPP-FU. More precisely, let us study the possibility to decompose a (difficult) instance into two (or more) smaller partial instances, which can then be solved independently of each other. In such a case, the solution of the complete instance simply results from an appropriate (and numerically less costly) composition of all partial solutions. For the TBPP, it is clear that an optimal solution is determined by the the maximum of the optimal values (and a concatenation of the corresponding schedules) the of the subproblems. In the case of the TBPP-FU, however, this is generally no longer the case, as the following result shows.

Theorem 3. *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:

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

[2]One possible example is the relationship between the cutting stock problem (CSP) and the skiving stock problem (SSP), which are not dual formulations of each other, but they are amenable to a large number of similar solution procedures.

So, in fact, there are four different item types: 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 $i \in \{1, \ldots, 9\}$ (so the items that are executed before $t = 4$), while the second subinstance E_2 collects all the remaining items $i \in \{10, \ldots, 15\}$ (so the items that have to be processed after $t = 4$). Now we can collect some important information about the respective solutions:

- Any optimal solution of E_1 requires four servers and seven fire-ups.
- Any feasible solution of E_1 with six fire-ups uses exactly six servers.
- Any optimal solution of E_2 requires six servers and six fire-ups.

So, putting together the individual optimal solutions of E_1 and E_2, we would end up with six servers and 13 fire-ups, i.e., an objective value of $z = 19$. However, it is better to combine a feasible solution of E_1 using six servers (and six fire-ups) together with an optimal solution of E_2 (also using six servers), because this only leads to an objective value of $z = 18$. Hence, there is no combination of optimal solutions of E_1 and E_2 which is optimal with respect to the complete instance E. □ □

Remark 4. *As a direct consequence of Theorem 2, decomposing an instance of the TBPP-FU is possible whenever $\gamma \leq 1/n$ holds.*

Altogether, the previous example proves for the first time(!) that it is not possible to decompose an instance of the TBPP-FU, in general, even if there is no activity at all in some time window. This 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 would like to deal exclusively with the typically more difficult case $\gamma > 1/n$ that does neither allow direct mathematical reference to the TBPP solution methods (in the sense of Theorem 2) nor any kind of temporal decomposition. Consequently, particularly for these parameter settings, the TBPP-FU has to be considered as an independent problem in cutting and packing.

3. A New Reduction Method

In this section we would like to introduce a new class of valid inequalities for the model M1. Even though, in the meantime, this formulation has been improved by numerous proposals in [17] and, later also, [18], these techniques almost exclusively addressed the size of the resulting ILP models or the symmetry of the solution space, thus naturally leading to a better handling of the optimization problems passed to the solvers. However, at least since the publication of the famous book [19], a milestone in integer programming, 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).

However, in the literature only one class of valid inequalities has so far been contributed to significantly raise the LP bound, see [17, Inequality (18)]. Many other candidates either did not lead to any improvement at all (see, for example, the inequalities (19)-(21) in [17]) or could only raise the LP bound to a vanishingly small extent (as witnessed in [18, Section 4.4]) – although, in the latter case, they have been applied quite successfully to neighboring problems, such as the uncapacitated lot sizing problem [23]. Thus, finding competitive, valid inequalities for the TBPP-FU has to be regarded as difficult. In addition, the only class found so far already makes a very large contribution to raising the LP bound, so that a further improvement of these values is even more hampered.

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 bring into focus an important aspect (of the TBPP-FU) that has been rather insufficiently dealt with so far. Remarkably, this improvement can already be built directly into the original model from [2] without much effort, but at first it seems less obvious or even counter-intuitive. Its concrete derivation and benefits can be revealed better on closer examination of the LP solution of a motivating example instance:

Example 1. *We consider the instance E with capacity $C = 2$, $n = 3$ items, and $\gamma = 1$, illustrated in Figure 3:*

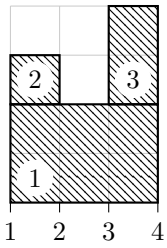

Figure 3: An illustration of the instance E from the current example.

When solving the LP relaxation of M1, we obtain the optimal value $z_{LP}^{(1),\star} = 3.5$ which is provided, for instance, by the optimal solution (partly) depicted in the left part of Fig. 4. Here, the sum of all activity variables, that is $Y_t := \sum_{k \in K} y_{tk}$, over time is displayed, and it can be seen that the optimal solution is able to hide a fraction of a fire-up at the ending time $t = 2$ of job $i = 2$. More precisely, at $t = 2$ the activity of the servers can be raised without being perceived by the fire-up constraints (5), because the latter are just present for $t \in T_S$. To mend this flaw, we can make use of w_{tk} and Constraints (5) for all $t \in T$ which helps to raise the LP bound in many cases. For the instance considered here, this modification leads to the better LP value $z_{LP}^{(1),\star} = 4$, because any (fraction of a) fire-up will be noticed correctly and counted by the objective function. Hence, there is no longer any motivation to raise Y_t at an ending time (like $t = 2$) of a job, and we obtain the more natural scenario depicted in the right part of Fig. 4.

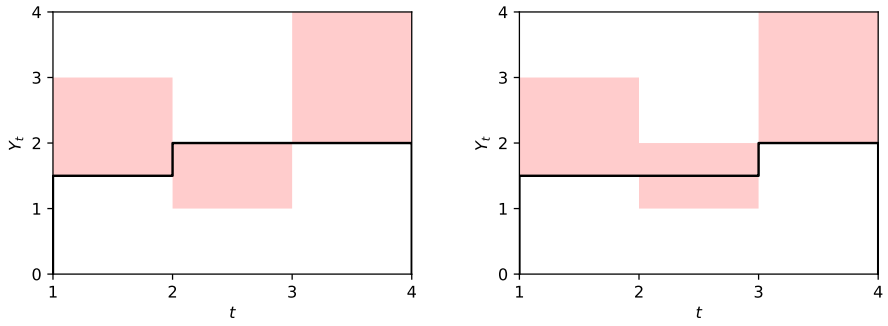

Figure 4: The summed activity $Y_t := \sum_{k \in K} y_{tk}$ over time in an optimal solution of E. The left picture shows the development of Y_t for Model 1, whereas the right picture visualizes the situation after having modified the model by using w_{tk} and Constraints (5) for all $t \in T$. In both cases, the red regions define some lower and upper bounds for Y_t that are derived from Constraints (1), (3), and (4) of M1.

Given the observations of Example 1, we propose the following improvement of M1 to prevent the LP relaxation from hiding (fractional) fire-ups, because the resulting LP solutions are unrealistic in terms of some of their properties and difficult to interpret from the integer model's point of

view. To express, on the one hand, that this improvement attaches directly to the original model from [2] and, on the other hand, not to conflict with the fixed identifiers (for the reductions) already assigned in [17] and [18], we will label this new class of valid inequalities introduced here as (R0).

(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 (5) for all $k \in K$ and $t \in T$.

For the original models M1 and M2, a dominance relation between the LP bounds is known, but without having gained further insight into the structure of these LP solutions or being able to express the concrete optimal values using closed-form terms.

Theorem 5 (see Proposition 4.1 in [2])**.** *Let E be an instance of the TBPP-FU. Then, we have $z_{LP}^{(1),\star} \geq z_{LP}^{(2),\star}$.*

In this article, we would like to contribute to improve this rather abstract result in the sense that we can give a concrete construction scheme of a solution of both LP relaxations. This will also allow us to state the optimal value and to specify properties of an instance which, for example, might be expected to have a very clear dominance relation, that is, a large difference $z_{LP}^{(1),\star} - z_{LP}^{(2),\star}$. Such a result for the TBPP-FU, which has so far been mainly the subject of purely numerical investigations, represents an important cornerstone for the further theoretical analysis of LP-based lower bounds as well as their mutual relations.

Remark 6. *For the improved versions of M1, M2, and M3, there are generally no dominance relations between the associated LP bounds, as could be shown in [17, Theorem 1]. While more detailed knowledge on the construction of an LP solution would be desirable for those formulations as well, a direct examination of these (optimized) models proves to be difficult. This may sound paradoxical at first, since the investigation of a smaller compact formulation might be thought to be somewhat easier, but this supposed advantage is clearly impeded by a very irregular and, moreover, highly instance-specific structure of the constraint matrix, which is a consequence of the numerous reductions. Thus, there can be prospects for successful analysis of the associated optimization problems in the future only if a fundamental understanding, including appropriate theoretical results, of the underlying original formulations has been obtained beforehand.*

In the following, we would now like to establish concrete procedures to construct an LP solution of M1 and M2, where, for the former, the argumentation will be based on the additional inequalities proposed as (R0) and elaborated on in Example 1. What the proofs will have in common is that they operate on a relaxation of the classical LP relaxation in which, as already indicated in Example 1, the variables summed up (over k) are central. This *aggregate relaxation* would look as follows in the case of M2, with which we want to start here since the proof is somewhat more instructive and easier to follow:

Aggregate Relaxation of M2

$$z_{agg}^{(2)} = \gamma \cdot \sum_{t \in T_S} W_t + Z \to \min$$

$$\text{s.t.} \quad \sum_{j \in \delta_i \cup \{i\}} c_j \leq C \cdot Z, \qquad i \in I, \tag{18}$$

$$h \leq Z, \tag{19}$$

$$W_{s_i} \geq 1 - |\delta_i^+|, \qquad i \in I, \tag{20}$$

$$W_t \in [0, n], \qquad t \in T_S, \tag{21}$$

$$Z \in [0, n]. \tag{22}$$

Observe that we basically just summed up any of the constraint sets appearing in M2 over k and used

$$Z := \sum_{k \in K} z_k, \quad W_t := \sum_{k \in K} w_{tk}, \ (t \in T_S), \quad \sum_{k \in K} x_{ik} = 1, \ (i \in I)$$

to define the above relaxation. We highlight that Condition (19) would first appear as $Z \geq 1$, but (as a consequence of Inequality (17)) we already merged it with the lower bound h presented in Sect. 2.

Lemma 7. *The optimal value of the aggregate relaxation (of M2) is equal to the optimal value of the ordinary LP relaxation (of M2).*

Proof. First of all, we note that after having combined some constraints, the feasible set cannot be smaller than before, so that the aggregate relaxation cannot have a larger optimal value than the 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

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

for any respective index or index pair. By that, we also respect the sorting $z_k \geq z_{k+1}$ of the z variables, and 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. □ □

The previous lemma helps us to find the optimal value of the LP relaxation (of M2) by considering a less difficult optimization problem which does not contain the index k anymore.

Theorem 8. *Let E be an instance of the TBPP-FU. Then the optimal value of the LP relaxation of M2 is given by*

$$z_{LP}^{(2),\star} = h + \gamma \cdot g(E), \tag{23}$$

with

$$g(E) := \left| \left\{ t \in T_S \,\middle|\, \exists i \in I \,:\, t = s_i, \, \delta_i^+ = \emptyset \right\} \right|.$$

Proof. Based on the previous observation, it is sufficient to consider the aggregate relaxation of M2. Then, it becomes obvious that there are no coupling conditions between the two types of variables (Z and W_t, $t \in T_S$) appearing in the objective function, so that they can be made as small as possible independent of each other. As a direct consequence of Inequality (17), we define $Z := h$, because this is the smallest possible value satisfying Constraints (18) and (19). From Conditions (20) we can derive

$$W_t := \begin{cases} 0, & \text{if we have } \delta_i^+ \neq \emptyset \text{ for all } i \in I \text{ with } t = s_i, \\ 1, & \text{if there is some } i \in I \text{ with } t = s_i \text{ and } \delta_i^+ = \emptyset, \end{cases}$$

for any $t \in T_S$. Hence, the case $W_t = 1$, $t \in T_S$, happens if and only if $t - 1 \notin \bigcup_{i \in I} [s_i, e_i)$ holds, i.e., precisely for all $t \in T_S$ with no activity at all at the preceding instant of time (in the set T). By definition, this can be observed for the very first starting time s_1 of item $i = 1$, but also after any "temporal gap" appearing in the instance. This proves (23). □ □

In particular, we can see that 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.

Similarly, the LP relaxation of M1 can be solved. Having applied (R0), also here, we just focus on the

Aggregate Relaxation of M1

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

$$\text{s.t.} \quad Y_t \leq \sum_{i \in I_t} c_i \leq Y_t \cdot C, \qquad t \in T, \tag{24}$$

$$1 \leq Y_{s_i}, \qquad i \in I, \tag{25}$$

$$h \leq Z, \tag{26}$$

$$Y_t - Y_{t-1} \leq W_t, \qquad t \in T, \tag{27}$$

$$Y_t \in [0, n], \qquad t \in T, \tag{28}$$

$$W_t \in [0, n], \qquad t \in T, \tag{29}$$

$$Z \in [0, n], \tag{30}$$

where we again modified the original form of Constraints (26), i.e., $Y_t \leq Z$, with respect to the lower bound h and the information contained in Inequality (17). With exactly the same arguments as above, we can prove that this relaxation leads to the same optimal value as the ordinary LP relaxation.

Lemma 9. *The optimal value of the aggregate relaxation (of M1) is equal to the optimal value of the ordinary LP relaxation (of M1).*

However, here it is not so easy to define an optimal solution of the former one since the different variable types are partly coupled.

Theorem 10. *Let E be an instance of the TBPP-FU. Then, an optimal solution of the aggregate relaxation of M1 is given by $Z := h$ and*

$$\begin{aligned} Y_t &:= \begin{cases} \min\left\{\sum_{i \in I_t} c_i, \max\left\{Y_{t-1}, \frac{\sum_{i \in I_t} c_i}{C}\right\}\right\}, & \text{if } t \notin T_S, \\ \min\left\{\sum_{i \in I_t} c_i, \max\left\{1, Y_{t-1}, \frac{\sum_{i \in I_t} c_i}{C}\right\}\right\}, & \text{if } t \in T_S, \end{cases} \quad (t \in T), \\ W_t &:= \max\{Y_t - Y_{t-1}, 0\}, \quad (t \in T), \end{aligned}$$

where $Y_0 := 0$ is used in the definition of Y_t and W_t if t is the minimal element of T.

Proof. We start by noting that $Z = h$ has to hold due to Constraints (26). Moreover, as a consequence of (27), 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 (which is important for the minimality of the objective function), 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 (24) and (25).
- **global view:** It is not reasonable to reduce the value of Y_t below that of Y_{t-1}, since W_t would not change. Moreover, an unnecessary reduction of Y_t (in the sense explained in the previous sentence) could at a later time $\tau \in T$ lead to a situation where Y_τ has to be increased too much (compared to $Y_{\tau-1}$), so that W_τ 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:** However, trying to keep the status $Y_t = Y_{t-1}$ (as it is intended by the previous observation) is only possible if this is allowed by the left hand side in Constraints (24). 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 with respect to a local and global point of view, so that they do not unnecessarily raise any of the variables W_t, $t \in T$, appearing in the objective function. This proves the claim. □ □

Remark 11. *The intention behind the item called "global view" in the previous proof can also be understood when having a look at the right picture in Fig. 4 again. 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 12. *Effectively, the nested minimum and maximum appearing in the definition of Y_t in the previous theorem 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 $\sum_{i \in I_t} c_i$ when considering 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 the previous remark is relatively easy to implement so that also the LP relaxation of M1 can be solved without requiring any optimization software. However, in contrast to M2, here we cannot state a simple formula for the optimal LP value. Of course, we know all the ingredients, but putting them together in the way defined by the objective function would lead to an even more complicated term containing further case studies and nested minimum or maximum operators. Due to this reason, we do not state an explicit result here.

As a conclusion of this section, we would like to highlight that the dominance of M1 towards M2 (in terms of the LP bound) is mainly connected to the way the fire-ups are processed. As can be clearly seen, 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 also collects fractional increases of the server activity variables y_{tk}, which typically leads to much better bounds due to Constraints (5).

4. Computational Tests

Although this article is mainly intended to make theoretical contributions to a better understanding of the TBPP-FU, we would also like to discuss the effects of the valid inequalities (R0) using some appropriate test computations. For this purpose, we consider the 160 benchmark instances described in [2, Section 5], which consist 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* $d_i := e_i - s_i$, $i \in I$, is 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 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 a *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 [2], but also incorporate this strategy into two more competitive versions of M1, namely the variants M1* and M1** introduced in [18]. 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 of [17]:

(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.

On top of that, the following techniques were proposed in [18]:

(a) We apply a set of smaller modifications to the inequality chain (1) appearing 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 lead to an intermediate version of M1, called M1* in [18], showing convincing numerical results. However, the application of further cuts known from [23] for a neighboring problem even lead to a slightly more powerful ILP formulation, called M1**, with respect to the benchmark set considered here.

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.

First, we would like to investigate the actual influence of the new valid inequalities on the LP bound. To this end, we list the average LP values of three different basic configuration, namely

- the literature version of M1 as given in Section 2,
- the version M1* that has gone through all six improvement steps, see [18],
- the currently best version M1**, which emerged from M1* by adding the constraints introduced in [23], see [18, Section 4.4],

and add the new improvement (R0) in each case, resulting in a total of six versions of M1, see Tab. 1. 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, somewhat 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.
- As indicated in the previous sections, there are two other compact formulations in the literature, M2 and M3, which have experienced a similar improvement process over time. In [17] and [18], it was shown that for the benchmark instances considered here, the LP values of all three formulations (with a comparable amount of applied reductions) were identical. The value given in the table for M1* therefore applies equally to M2* and M3* from [18]. Having a more detailed look at the instance-specific results that led to the averages in Tab. 1, we now notice that already the LP bound of M1*+(R0) dominated[3] that of the other models (that is, M2* and M3*) in exactly 13 cases, showing on average 13% (and, in the maximum, even 40%) better values for these instances.
- All these 13 instances contained $n \geq 100$ items and belonged to the constellation (d_S, c_H), with only one single exception. The fact that (R0) seems to be particularly useful for that configuration can be clearly seen in most of the comparisons (between a fixed formulation and its extension using (R0)) in Tab. 1. This observation is reasonable, because in these scenarios the occurrence of fire-ups is typically somewhat 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. Hence, a model improvement that precisely addresses the accurate reconstruction of server activities (and, thus, implicitly also fire-ups) in the case of continuous variables (such as (R0)) shows a sometimes significant positive effect for the corresponding LP bounds.

[3] This dominance would become even more clear when comparing M1** with the other compact models, but the focus of this study is to present the benefits of (R0) itself.

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
			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

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

Altogether, it can be concluded that applying (R0) is reasonable 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 13. *In [2], even larger instances than those mentioned here were considered, but treated exclusively heuristically due to the huge model size. An optimal solution of these instances is generally not possible in reasonable time even with the improved compact models, so we cannot include them here in our more in-depth investigations. Nevertheless, in the light of our observations so far, it seems worthwhile to us to at least briefly outline the importance of Reduction (R0) for the LP bounds of these instances. In Tab. 2, we impressively see that the previous observations all apply also to these very large 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.

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

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

In a second experiment, we would now like to focus on a broader range of performance metrics, but only for the currently best formulation M1** presented in [18]. To this end, we first list some average indicators to get an overview of the resulting approaches, see Tab. 3.

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%

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.

It can be seen 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. 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.

To conclude, in Tab. 4 we would like to take a closer look at the overall numerical performance

of both variants of M1, including the configuration parameters from our benchmark set. By that, we intend to derive insights into which of the formulations seems particularly suitable for which combination of input parameters. The main results are as follows:

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

Table 4: 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).

- Overall, both formulations show similar results in the test calculations and are relatively close to each other in the average numbers. Nevertheless, we can see that except for the very simple instances with $n = 50$, the setting M1**+(R0) always performs slightly better. Tab. 1 shows that for the aforementioned instances, there is no effect with respect to the LP bound, but the model size increases somewhat (due to the additional inequalities). We think that this is crucial for not recognizing any positive contribution from our improvements for these (rather easy) instances. For all other values of n, the unfavorable effect of a larger optimization problem can be counterbalanced by the generally better bounds, so that a positive effect can be observed.

- As already stated in the discussion of Tab. 1, the new valid inequalities prove to be particularly useful for the LP bound when short jobs with high capacity requirements are present (scenario (d_S, c_H)). This can also be clearly seen in many places in Tab. 4. For example, in the scenario considered here, setup M1**+(R0) actually succeeds in solving all instances for $(n, \bar{s}) = (100, 120)$ and also determines two additional optimal solutions for the particularly hard instances with $(n, \bar{s}) = (200, 240)$. Thus, especially for this combination of input data, the formulation M1**+(R0) is highly recommended.

- This is different for the case of short jobs with low resource demands (scenario (d_S, c_L)). Here, 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 (also in LP relaxation) are typically possible, which is supported by the optimal LP values in Tab. 1, and an impact of our new inequalities on the LP bound is therefore not detectable. 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 cases. For the scenario (d_S, c_L), the valid inequalities from (R0) should therefore generally be omitted.

5. Conclusions

In this article we have dealt with selected aspects in the analysis of the TBPP-FU, which have so far received only little attention in the relevant literature. By this we mean, on the one hand, theoretical contributions to a relatively young field of research, which has previously been the subject of mainly practical investigations, but at the same time also a further improvement of the numerical properties by purposefully targeting those model parameters which the techniques known from [17] and [18] have not yet been able to address sufficiently well. First, we succeeded in disproving the validity of temporal decompositions, a fundamental property (especially from the point of view of the solution process) of the ordinary TBPP, for the TBPP-FU and, thus, to work out a significant difference of both optimization problems, which finally establishes the TBPP-FU as an independent problem in cutting and packing. Furthermore, a new class of valid inequalities was proposed which, according to our test calculations, leads to the improvement of different versions of M1-type formulations. In particular, this is due to the fact that these additional cuts contribute to a noticeable increase of the LP bound even when (numerous) other reduction methods have already been applied. From a theoretical point of view, this also paves the way to accurately analyze the relationship of the LP bounds of M1 and M2, thus generalizing a previously only very abstract result from the related literature.

For the future, it is of great interest to see whether these insights into the structure of the LP solution can be extended to similar results for the improved models from [17] and [18] as well. By that, it might be possible to justify, among other things, the empirical dominance of the LP bound of the configuration M1**+(R0) over those of the other formulations (M2* and M3*), which was observed in the course of this work, also theoretically. 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 13, the latter could be clearly promoted (especially for some instance parameters) in the context of this work.

References

[1] Andrae, A.S.G., Edler, T.: On Global Electricity Usage of Communication Technology: Trends to 2030. Challenges 6(1), 117–157 (2015)

[2] Aydin, N., Muter, I., Ilker Birbil, S.: Multi-objective temporal bin packing problem: An application in cloud computing. Computers & Operations Research 121, Article 104959 (2020)

[3] Bartlett, M., Frisch, A.M., Hamadi, Y., Miguel, I., Tarim, S., Unsworth, C.: The temporal knapsack problem and its solution. Lecture Notes in Computer Science 3524, 34–48 (2005)

[4] Caprara, A., Furini, F., Malaguti, E.: Uncommon Dantzig-Wolfe reformulation for the temporal knapsack problem. INFORMS Journal on Computing 25(3), 560–571 (2013)

[5] Caprara, A., Furini, F., Malaguti, E., Traversi, E.: Solving the temporal knapsack problem via recursive Dantzig-Wolfe reformulation. Information Processing Letters, 116(5), 379–386 (2016)

[6] Cohen, M.C., Keller, P.W., Mirrokni, V., Zadimoghaddam, M.: Overcommitment in Cloud Services: Bin Packing with Chance Constraints. Management Science 65(7), 3255–3271 (2019)

[7] Coffman Jr., E.G., Garey, M.R., Johnson, D.S.: An Application of Bin Packing to Multiserver Scheduling. SIAM Journal on Computing 7(1), 1–17 (1978)

[8] Côté, J.F., Dell'Amico, M., Iori, M.: Combinatorial Benders' cuts for the strip packing problem. Operations Research 62(3), 643–661 (2014)

[9] de Cauwer, M., Mehta, D., O'Sullivan, B.: The Temporal Bin Packing Problem: An Application to Workload Management in Data Centres. Proceedings of the 28th IEEE International Conference on Tools with Artificial Intelligence, 157–164, (2016)

[10] Dell'Amico, M., Furini, F., Iori, M.: A Branch-and-Price Algorithm for the Temporal Bin Packing Problem. Computers & Operations Research 114, Article 104825 (2020)

[11] Delorme, M. Iori, M., Martello, S.: Bin packing and Cutting Stock Problems: Mathematical Models and Exact Algorithms. European Journal of Operational Research 255, 1–20 (2016)

[12] 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, G.T., Plettemeier, D., Schröter, M., Strufe, T.: Architecture and advanced electronics pathways towards highly adaptive energy-efficient computing. Proceedings of the IEEE 107(1), 204–231 (2019)

[13] Iori, M., de Lima, V.L., Martello, S., Miyazawa, F.K., Monaci, M.: Exact solution techniques for two-dimensional cutting and packing. European Journal of Operational Research 289(2), 399–415 (2021)

[14] Jones, N.: How to stop data centres from gobbling up the world's electricity. Nature 561, 163–166 (2018)

[15] López-Pires, F., Barán, B.: Virtual Machine Placement Literature Review. arXiv Preprint (*available online:* `http://arxiv.org/abs/1506.01509`) (2015)

[16] Martinovic, J., Selch, M.: Mathematical Models and Approximate Solution Approaches for the Stochastic Bin Packing Problem. Computers & Operations Research 135, Article 105439 (2021)

[17] Martinovic, J., Strasdat, N., Selch, M.: Compact Integer Linear Programming Formulations for the Temporal Bin Packing Problem with Fire-Ups. Computers & Operations Research 132, Article 105288 (2021)

[18] Martinovic, J., Strasdat, N., Valério de Carvalho, J., Furini, F.: Variable and constraint reduction techniques for the temporal bin packing problem with fire-ups. Optimization Letters (2021) (https://link.springer.com/article/10.1007/s11590-021-01825-x)

[19] Nemhauser, G., Wolsey, L.: Integer and Combinatorial Optimization. Wiley, New York (1988)

[20] Scheithauer, G.: Introduction to Cutting and Packing Optimization – Problems, Modeling Approaches, Solution Methods. International Series in Operations Research & Management Science 263, Springer, 1.Edition (2018)

[21] Valério de Carvalho, J.M.: LP models for bin packing and cutting stock problems. European Journal of Operations Research 141(2), 253–273 (2002)

[22] Wang, M., Meng, X., Zhang, L.: Consolidating Virtual Machines with Dynamic Bandwidth Demand in Data Centers. Proceedings of the IEEE INFOCOM, 71–75 (2011)

[23] Wolsey, L.A.: Uncapacitated Lot-Sizing Problems with Start-Up Costs. Operations Research 37(5), 741-747 (1989)