Als Manuskript gedruckt

Technische Universität Dresden
Herausgeber: Der Rektor

Worst-Case Analysis of Heuristic Approaches for the Temporal Bin Packing Problem with Fire-Ups

John Martinovic, Nico Strasdat

Preprint MATH-NM-03-2022

August 2022

Worst-Case Analysis of Heuristic Approaches 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

We consider the temporal bin packing problem with fire-ups (TBPP-FU), a branch of operations research recently introduced in multi-objective cloud computing. In this scenario, any item is equipped with a resource demand and a lifespan meaning that it requires the bin capacity only during that time interval. We then aim at finding a schedule minimizing a weighted sum of the total number of bins required and the number of switch-on processes (so-called fire-ups) caused during operation. So far, research on the TBPP-FU has mainly focused on exact approaches and their improvement by valid cuts or variable reduction techniques. Although these studies have revealed the problem considered here to be very difficult to cope with, theoretical contributions to heuristic solution methods have not yet been presented in the available literature. Hence, in this article we investigate the worst-case behavior of some approximation algorithms, ranging from classic online algorithms to a more sophisticated look-ahead heuristic specifically designed for the TBPP-FU. As a main contribution, we constructively show that the feasible solutions obtained by all these approaches can be arbitrarily bad. By doing so, we establish another previously unknown difference between the classical TBPP and the extended problem with fire-ups, rendering the latter the more difficult problem even from a heuristic point of view.

Keywords: Cutting and Packing, Temporal Bin Packing, Fire Ups, Heuristics, Worst-Case Analysis

1. Introduction

The *temporal bin packing problem (TBPP)* generalizes the classic BPP, see [10, 23], with respect to an additional time dimension. More precisely, any item $i \in I := \{1, \dots, n\}$ is specified by a *resource demand* (or *item size*) $c_i \in \mathbb{Z}_+$ that has to be satisfied only during the *lifespan* $[s_i, e_i)$ of that item, where $s_i, e_i \in \mathbb{Z}_+$ with $s_i < e_i$ denote the *starting and ending time*, respectively. A set of given items then has to be assigned to as few bins as possible while respecting the bin capacity $C \in \mathbb{Z}_+$ at any instant of time. It is important to note that even though the relationships to two-dimensional packing problems seem obvious, the TBPP is an independent problem in operations research. This is particularly due to the fact that the bin capacity represents a renewable resource at every instant of time, and, consequently, items do not have to occupy the same units (of the bin) over their entire lifespan, see [9, 19] for a more detailed explanation.

Although the TBPP is a fairly natural extension of the extensively studied BPP, its scientific foundations have been driven mainly by previous research on the *temporal knapsack problem (TKP)*, see [2, 3, 15]. Consequently, the TBPP was first described rather lately in the relevant literature in an application-oriented publication from the field of computer science, see [8]. Nonetheless, addressing the exact solution of the TBPP has been successfully advanced by two sophisticated approaches, namely a branch-and-bound algorithm (using Ryan-Foster branching together with a wide variety of different bounds), see [9], and a layer-based combinatorial arcflow model of manageable exponential size, see [21]. Surprisingly, searching the relevant literature for contributions on heuristic methods for the TBPP does not immediately lead to the desired results. This is not because such approaches do not exist at all, but rather because they were

*Corresponding author

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

already discussed in early publications on the so-called *dynamic bin packing problem* about 40 years ago (and thus well before the introduction of the term TBPP), see [6]. As a consequence of that, it seems that any follow-up article dealing with that topic, like [4, 5], has stuck to this original terminology rather than harmonizing it with the parallely evolving "temporal notations". These publications mainly focussed on rather simple heuristics the properties of which were already well studied for the classical BPP. To be more precise, special emphasis was given to the following iterative online[1] algorithms:

- *any-fit (AF)*, scheduling the current item to an arbitrary open bin, see [4],
- *first-fit (FF)*, assigning the current item to the lowest-indexed open bin, see [4, 5, 6]
- *best-fit (BF)* and *worst-fit (WF)*, trying to pack an item into the open bin with the currently largest (BF) or smallest (WF) load, respectively, see [4].

For all these heuristics, the quality of the feasible solutions obtained has been studied thoroughly and, in many cases, tight approximation factors could be found. In this context, it is remarkable that none of these publications is cited in the most recent TBPP literature, such as [9, 21], suggesting that the existence of these theoretical contributions is largely unknown to the cutting and packing community. For this reason, we will briefly summarize some of the results in the next section, also to better display the differences that arise when so-called *fire-ups* are included.

Considering fire-ups in item-to-bin assignments is a relatively new aspect of modelling and optimization introduced as the *temporal bin packing problem with fire-ups (TBPP-FU)* in an application from the field of cloud computing, see [1]. The basic idea is that an unused server (or bin) can be temporarily put into some idle mode to save energy, see [13], but it has to be re-activated later if necessary[2]. Any such transition from an empty state into active operation is counted as one fire-up and it is rather energy-intense, meaning that, as a second objective, the number of fire-ups should be kept small to operate sustainably. Roughly spoken, a low number of fire-ups relates to continuous operation of the servers or a scenario where servers can be switched off without being required again later.

Typically, a weighted sum method (scaling the number of fire-ups by some parameter $\gamma > 0$) is used to address both goals together in one objective function, see [1]. Although there is a quite strong relation to the TBPP, research has shown that important properties are lost as a consequence of the generalized problem statement. In the literature, the two main differences are given by:

- An optimal solution to the TBPP-FU typically uses more bins than required in an optimal configuration without considering fire-ups, see [1, Example 2.2]. Hence, solving the TBPP does not necessarily lead to an upper bound on the number of bins required in the TBPP-FU.
- In general, temporal decompositions cannot be applied to the TBPP-FU, see [22, Theorem 3], meaning that an instance typically cannot be split into independent subinstances of smaller size.

So, not only the solution sets of the two problems may be completely disjoint, but also some fundamental techniques exploiting the structural properties of an instance cannot be used to obtain these solutions in case of the TBPP-FU, in general. As a consequence of that, research has mainly dealt with improving the ILP formulations (called M1 and M2) proposed in [1] by various aspects like symmetry breaking conditions and valid cuts, see [19], as well as clique-based reduction methods or the use or heuristic information, see [20]. For the latter, the *constructive look-ahead heuristic (CLH)* introduced in [1] was used, since it is the only approximation algorithm, known in the literature, specifically addressing the fire-up term in the objective function. While empirically it was shown that lots of variables and constraints can be removed based on the heuristic solution, its theoretical properties have not been dealt with at all so far.

In this article, we would therefore like to focus on the approximation guarantee of heuristic approaches

[1] An online algorithm has to make its decision just with the information available when placing the current item. In particular, there is no further knowledge of which items will arrive next.

[2] Note that similar ideas are also discussed in the field of thermal units, see [14].

for the TBPP-FU. As already alluded to earlier, we will start with a short repetition of standard online algorithms known from the classic TBPP, collect their main theoretical properties (→ Sect. 2), and show that their worst-case performance ratio is no longer bounded, when fire-ups have to be respected (→ Sect. 3). As a main contribution, we prove the same result for CLH thus closing an open theoretical question for the only TBPP-FU heuristic known in the literature (→ Sect. 3). Moreover, our investigations are not only the first to cover theoretical properties of heuristic approaches for the TBPP-FU, but they do also establish a third fundamental difference between the problem under consideration and the underlying TBPP, that is, the hardness of finding reasonably good approximate solutions by (common) heuristics.

2. Heuristics for the TBPP: An Overview and Important Results

Let us start with the following definition:

Definition 1. *A tuple $E = (n, C, \boldsymbol{c}, \boldsymbol{s}, \boldsymbol{e})$, where $\boldsymbol{c}$, $\boldsymbol{s}$, and $\boldsymbol{e}$ are n-dimensional vectors collecting the input-data of the items, is called an instance (of the TBPP).*

Without loss of generality, we assume the items to be sorted with respect to non-decreasing starting times (breaking ties in an arbitrary way) and to satisfy $c_i \leq C$ to ensure solvability. For any given algorithm ALG, we define the *(worst case) performance ratio* $\sigma := \sigma(ALG)$ by

$$\sigma(ALG) := \sup_{E} \frac{ALG(E)}{OPT(E)},$$

with $ALG(E)$ and $OPT(E)$ denoting the heuristic and the optimal value (of E), respectively. Although there is a certain body of work dealing with heuristics for the TBPP, as mentioned earlier, the fact that they were all published with respect to a completely different terminology might be the reason why there is no link between the most recent literature dealing with exact approaches (partly requiring and benefiting from heuristic information) and the former theoretical results related to what was called dynamic bin packing. To close this gap, let us briefly repeat the most important results obtained at that time.

The first heuristic proposed in the literature is of first-fit type, see [6], and an interval for the performance ratio is given by the following result.

Theorem 1 (see Theorem 2 and Theorem 6 in [6])**.** *We have*

$$2.389 \approx \frac{43}{18} \leq \sigma(FF) \leq \frac{5}{2} + \frac{3}{2} \log\left(\frac{\sqrt{13}-1}{2}\right) \approx 2.897.$$

The lower bound also holds for any arbitrary online algorithm.

The proofs related to these bounds are very technical and shall therefore be omitted. Among others, establishing the lower bound requires an instance construction containing eleven steps (partly with several subcases). Note that, even if the true value of $\sigma(FF)$ was not identified in that early publication, the results obtained are nevertheless quite remarkable:

- First of all, we see that adding a temporal dimension to the classical BPP makes it much harder to find a feasible solution of good quality with reasonable numerical efforts. By that, we particularly mean that the known approximation factor of FF for the BPP (that is, 1.7, see [11]) is possibly raised by up to more than one unit.
- Secondly, already this very first article dealing with heuristics for the TBPP was able to establish a lower bound for a wide variety of approximation algorithms.

In [4], the authors present new results for all the simple heuristics mentioned in the above list, see Sect. 1, but some of their considerations are limited to *unit fraction* item sizes (meaning that the bin capacity C is an integer multiple of any c_i, $i \in I$). In the cutting and packing literature, such a scenario is sometimes also referred to as the *divisible case*, see [7, 17, 18].

Remark 2. *Note that an upper bound for $\sigma(AF)$ also holds for FF, BF, and WF, since the latter are, in a sense, "special cases" that can occur in the random bin selection process of AF.*

In particular, the following results are obtained:

Theorem 3 (see Theorem 7 and Theorem 8 in [4])**.** *We have $\sigma(WF) \geq 3$ and $\sigma(BF) \geq 3$.*

Interestingly, in [4, Theorem 6] the tightness of these approximation factors (of BF and WF) for instances with unit fraction item sizes was shown. In fact, the corresponding proof can be easily extended to arbitrary instances, so that even the following result holds:

Theorem 4. *We have $\sigma(AF) \leq 3$.*

In the light of Remark 2, we now know that BF and WF do possess a worst-case performance ratio of 3. This is a remarkable qualitative difference to the situation for the classic BPP due to two reasons:

- Firstly, we see that there is no difference between BF and WF in terms of the approximation guarantee. For the BPP, BF is known to perform better than WF, see [12, 16].
- Secondly, FF is better than BF for the TBPP, whereas from a worst-case perspective both of them were equivalent for the BPP, see [11, 12].

As a last point, we mention that for FF the following two improvements of the lower bounds from [6] can be obtained:

Theorem 5 (see Theorem 1 and Theorem 5 in [4])**.** *We have $\sigma(FF) \geq 2.45$. Moreover, for the subclass of unit fraction item sizes the performance ratio of FF is bounded above by a constant less than* 2.5*.*

Theorem 6 (see Theorem 1 in [5])**.** *For any online algorithm, we have $\sigma(ALG) \geq \frac{5}{2}$.*

Both together imply that approximating the TBPP is harder for arbitrary item sizes than for unit fractions. However, the exact performance guarantee of FF is still not known in either case.

3. Heuristics for the TBPP-FU: A Worst-Case Analysis

Let us define a family of TBPP-FU instances $E(\alpha, \beta)$ parametrized by $\alpha, \beta \in \mathbb{Z}_+$ with $\alpha \geq 1$. More precisely, any such instance uses the bin capacity $C = 2$, some scaling parameter $\gamma > 0$, and is given by the following items:

- two items (Type 'A') with $[s_A, e_A) = [1, 2\alpha)$ and $c_A = 1$,
- β items (Type 'B') $[s_B, e_B) = [1, 2\alpha)$ and $c_B = 2$,
- α items (Type 'C', labelled from 1 to α) with $[s_i, e_i) = [2i - 1, 2i)$ and $c_i = 1$, $i = 1, \ldots, \alpha$.

Note that γ does not affect the shape of the items, so that we do not have to specify this value when illustrating an instance. Some exemplary configurations are given in Fig. 1 and Fig. 2. With the help of these instances, we will show that any heuristic from the literature proposed for the TBPP and the TBPP-FU can be arbitrarily bad. To this end, let us first construct an optimal solution of $E(\alpha, \beta)$.

Theorem 7. *For any feasible choice of (α, β) and any scaling parameter $\gamma > 0$ we have*

$$OPT(E(\alpha, \beta)) = (1 + \gamma) \cdot (\beta + 2).$$

Proof. Given the items available at $t = 1$, at least $\beta + 2$ bins are required in an optimal solution. Hence, it suffices to find a feasible solution using precisely this number of bins in continuous operation (i.e., with one fire-up per bin). To achieve this, we consider the following assignment:

- Any of the β items of type 'B' requires a seperate bin, leaving no space for any other item to be added.

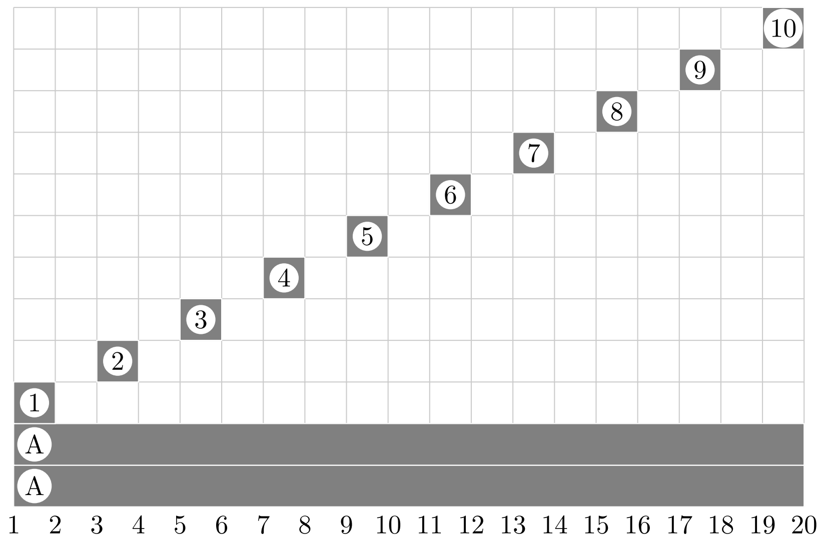

Figure 1: An illustration of $E(\alpha, \beta)$ for $\alpha = 10$ and $\beta = 0$.

- We pack one item of type 'A' together with all the α items of type 'C'.
- The last bin just contains one item of type 'A'.

Altogether, this solution uses $\beta + 2$ bins each having exactly one fire-up (at the very beginning). Hence, the optimal value is given by $(1 + \gamma) \cdot (\beta + 2)$ and the claim is proved. □

As a direct consequence of that, we can state:

Theorem 8. *For any $\gamma > 0$, the worst-case performance ratio of AF, FF, BF, and WF (applied to the TBPP-FU) is unbounded.*

Proof. Let us consider the instance $E(\alpha, 0)$ with some arbitrary $\gamma > 0$, see also Fig. 1. Then, in any of the heuristics mentioned before the items are placed in the same way since there is precisely one possibility in every iteration. Hence, either way, we end up with one bin grouping the two items of type 'A' (that is, one fire-up) and one bin collecting all items of type 'C' (that is, α fire-ups). Altogether, we have $ALG(E(\alpha, 0)) = 2 + \gamma \cdot (1 + \alpha)$, meaning that the ratio $ALG(E)/OPT(E)$ is unbounded when α tends to infinity. □

Obviously, the previous result also holds for any other iterative online algorithm, since the reason for the bad performance is that the algorithm is not allowed to open an additional bin when the existing bins are able to accommodate the currently considered item. In fact, this restriction cannot be relaxed either, because an online algorithm does not know which items will follow in future (if any), and so there is no basis for deciding whether to add an extra bin or not.

Although these critical features have not been identified or reported before, the constructive look-ahead heuristic (CLH) proposed in [1], see Alg. 1, was intuitively equipped with some tailored modifications addressing these issues. First of all, this is not an online algorithm since it uses a certain amount of future items (specified by q) when making the current decision. Moreover, and even more importantly, it allows to open an additional bin at every stage to possibly get around the critical situation observed in the previous example. Indeed, CLH would solve the instance displayed in Fig. 1 correctly.

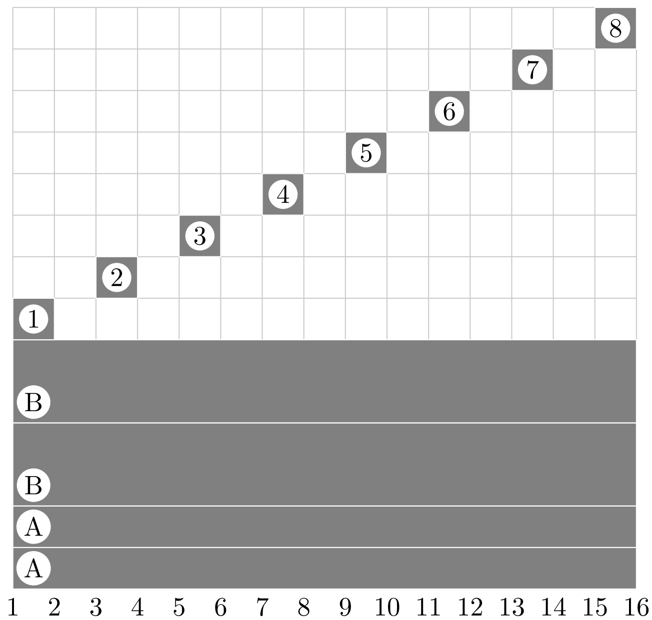

Figure 2: An illustration of $E(\alpha, \beta)$ for $\alpha = 8$ and $\beta = 2$.

Algorithm 1 CLH with look-ahead parameter q

Input: Item list ordered by non-decreasing starting times s_i, parameter $q \in \mathbb{N}$.

Initialize the "empty" assignment $\mathcal{A}(0) := \emptyset$.

for $i \in I$ **do**

  Assign item i to any open bin of $\mathcal{A}(i-1)$ that can accommodate it and (as another alternative) also to a new empty bin. By that we obtain the assignments $\mathcal{A}_1(i), \ldots, \mathcal{A}_p(i)$ for some $p := p(i) \leq n$.

  Add the next q items (or less, if $i + q > n$) to each of these allocations in a best-fit fashion and obtain the (updated) assignments $\widetilde{\mathcal{A}}_1(i), \ldots, \widetilde{\mathcal{A}}_p(i)$.

  Choose one assignment from $\widetilde{\mathcal{A}}_1(i), \ldots, \widetilde{\mathcal{A}}_p(i)$ with the lowest objective value, say $\widetilde{\mathcal{A}}_{opt}(i)$, and define $\mathcal{A}_{opt}(i)$ as $\mathcal{A}(i)$, that is, the starting point of the next iteration.

end for

Output: heuristic solution with objective value z_{heu}.

Remark 9. *Note that a deeper look into the future (i.e., a larger value of q) does not necessarily lead to an equivalent or even better result. By way of example, let us consider the instance E defined by $n = 5$, $C = 2$, $\gamma = 1$, and the following item characteristics*

$$\boldsymbol{c} = (1, 1, 1, 1, 2), \quad \boldsymbol{s} = (1, 1, 2, 4, 5), \quad \boldsymbol{e} = (6, 6, 3, 5, 6).$$

For $q = 1$, CLH finds an optimal solution using two bins and three fire-ups by forming the bins $B_1 = \{1, 2\}$ and $B_2 = \{3, 4, 5\}$. Contrary to that, $q = 2$ leads to a worse objective value by requiring the three bins $\widetilde{B}_1 = \{1, 3, 4\}$, $\widetilde{B}_2 = \{2\}$, and $\widetilde{B}_3 = \{5\}$, together with three fire-ups. Hence, there is no strict relation between q and the objective value obtained by Alg. 1.

By inserting sufficiently many items of type 'B', as in Fig. 2, the possibilities to successfully look into the future can be limited, especially for the crucial decision how to deal with the first two items (of type 'A'). Hence, we obtain the following result, in which we refer to Alg. 1 and a fixed choice of q by CLH_q.

Theorem 10. *For any feasible choice of (α, β) and any scaling parameter $\gamma > 0$ we have*

$$CLH_q(E(\alpha, \beta)) = \beta + 2 + \gamma \cdot (\beta + 1 + \alpha),$$

if $q = \beta$ is chosen as the look-ahead parameter in CLH.

Proof. According to the rules given in CLH, see Alg. 1, we have the following observations:

- The first item of type 'A' is scheduled to bin $k = 1$.
- The second item of type 'A' can either enter bin $k = 1$ or open a new bin $k = 2$. The decision between these two possibilities is done by involving the next β items in a best-fit fashion. Since the next β items are all of type 'B', they will consume precisely one bin in any feasible assignment. Hence, in terms of the objective value, CLH will choose to pack both items of type 'A' into the same bin (that is, $k = 1$).
- The next β items (all type 'B' items) will always be assigned to a separate bin, since they cannot be packed in any of the existing bins.
- Due to the same reason, the first item of type 'C' (labelled $i = 1$) has to open a new bin.
- The remaining items of type 'C' can either go to the same bin (as $i = 1$) or open an additional one. However, the latter will lead to a larger objective value for any $\gamma > 0$. Hence, they are all scheduled to the same bin $k = \beta + 2$.

Altogether, at the end of CLH_q we have $\beta + 1$ bins with precisely one fire-up and one bin with α fire-ups, summing up to $\beta + 2$ bins and $(\beta + 2) + (\alpha - 1) = \beta + 1 + \alpha$ fire-ups. This proves the claim. □

Obviously, both observations together ensure that for any possible q the heuristic solution obtained by CLH can be arbitrarily bad.

Theorem 11. *For any look-ahead parameter $q \in \mathbb{N}$ we have*

$$\sigma\left(CLH_q\right) = \sup_E \frac{CLH_q(E)}{OPT(E)} = \infty.$$

In particular, this result is independent of the choice of $\gamma > 0$.

Proof. For any fixed $\gamma > 0$, we can use the instances $E(\alpha, \beta)$ with $\beta = q$ and consider the case $\alpha \to \infty$. □

Altogether, we see that the approximation guarantee of CLH is unbounded for every choice of q, too. This is a quite remarkable result since CLH was able to use information of future items to keep the objective value low, e.g., by already opening an additional bin for the item to be scheduled right now.

4. Conclusions

In this article, we considered heuristic approaches from the literature for two neighboring optimization problems, the TBPP and the TBPP-FU. For the former, we briefly repeated some important results obtained from publications on dynamic bin packing, because according to our impression these contributions have not yet made their way into the most recent scientific discussion on the TBPP. For the TBPP-FU, in a first step we were able to show that the worst-case performance ratio of the former TBPP heuristics is unbounded when fire-ups have to be considered. Having identified the critical features of these algorithms, we observed that they are not present in the only heuristic specifically designed for the TBPP-FU in the literature, that is, CLH from [1]. However, despite the promising modifications contained in CLH, the approximation guarantee of this heuristic is unbounded, too. Remarkably, this result is independent of the scaling parameter γ and the look-ahead parameter q. Altogether, this work represents the first systematic framework to study theoretical properties of heuristic approaches for the TBPP-FU, showing some previously unknown differences between the two temporal problems under consideration. On the other hand, our contributions inevitably lead to a new open challenge for future research in cutting and packing, that is: How to construct a (preferably simple) heuristic for the TBPP-FU, the performance ratio of which can be shown to be bounded?

References

[1] 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)

[2] 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)

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

[4] Chan, J.W.-T., Lam, T.-W., Wong, P.W.H.: Dynamic bin packing of unit fraction items. Theoretical Computer Science 409, 521–529 (2008)

[5] Chan, J.W.-T., Wong, P.W.H., Yung, F.C.C.: On Dynamic Bin Packing: An Improved Lower Bound and Resource Augmentation Analysis. Algorithmica 53, 172–206 (2009)

[6] Coffman Jr., E.G., Garey, M.R., Johnson, D.S.: Dynamic Bin Packing. SIAM Journal on Computing 12(2), 227–258 (1983)

[7] Coffman Jr., E.G., Garey, M.R., Johnson, D.S.: Bin packing with divisible item sizes. Journal of Complexity 3(4), 406 – 428 (1987)

[8] 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)

[9] 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)

[10] 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)

[11] Dósa, G., Sgall, J.: First Fit bin packing: A tight analysis. 30th International Symposium on Theoretical Aspects of Computer Science (STACS 2013), 538–549 (2013)

[12] Dósa, G., Sgall, J.: Optimal Analysis of Best Fit Bin Packing. Lecture Notes in Computer Science 8572, 429–441 (2014)

[13] 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)

[14] Frangioni, A., Gentile, C.: Solving Nonlinear Single-Unit Commitment Problems with Ramping Constraints. Operations Research 54(4), 767–775 (2006)

[15] Gschwind, T., Irnich, S.: Stabilized column generation for the temporal knapsack problem using dual-optimal inequalities. OR Spectrum 39, 541–556 (2017)

[16] Johnson, D.S.: Near-optimal bin packing algorithms. PhD dissertation, Massachusetts Institute of Technology (1973)

[17] Marcotte, O.: Topics in combinatorial packing and covering. Technical Report No. 568, Cornell University (1983)

[18] Martinovic, J.: A Note on the Integrality Gap of Cutting and Skiving Stock Instances: Why 4/3 is an Upper Bound for the Divisible Case? 4OR 20, 85–104 (2022)

[19] 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)

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

[21] Martinovic, J., Strasdat, N., Valério de Carvalho, J.M., Furini, F.: A Combinatorial Flow-based Formulation for Temporal Bin Packing Problems. Preprint MATH-NM-02-2022, Technische Universität Dresden (2022)

[22] Martinovic, J., Strasdat, N.: Theoretical Insights and a New Class of Valid Inequalities for the Temporal Bin Packing Problem with Fire-Ups. Preprint MATH-NM-01-2022, Technische Universität Dresden (2022) (`http://www.optimization-online.org/DB_HTML/2022/02/8791.html`)

[23] 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)