Noname manuscript No.
(will be inserted by the editor)

A Two-Phase Algorithm for Bin Stretching with Stretching Factor 1.5

Martin Böhm · Jiří Sgall · Rob van Stee · Pavel Veselý

the date of receipt and acceptance should be inserted later

Abstract ONLINE BIN STRETCHING is a semi-online variant of bin packing in which the algorithm has to use the same number of bins as an optimal packing, but is allowed to slightly overpack the bins. The goal is to minimize the amount of overpacking, i.e., the maximum size packed into any bin.

We give an algorithm for ONLINE BIN STRETCHING with a stretching factor of 1.5 for any number of bins. We build on previous algorithms and use a two-phase approach. However, our analysis is technically more complicated and uses amortization over the bins with the help of two weight functions.

Keywords online algorithms · bin packing · scheduling

1 Introduction

The most famous algorithmic problem dealing with online assignment is arguably ONLINE BIN PACKING. In this problem, known since the 1970s, items of size between 0 and 1 arrive in a sequence and the goal is to pack these items into the least number of unit-sized bins, packing each item as soon as it arrives.

ONLINE BIN STRETCHING, which has been introduced by Azar and Regev in 1998 [3,4], deals with a similar online scenario. Again, items of size between 0 and 1 arrive in a sequence, and the algorithm needs to pack each item as soon as it arrives, but there are the following differences: (i) The packing algorithm knows m, the number of bins that an optimal offline algorithm would use, and must also use only

Supported by the project 14-10003S of GA ČR and by the GAUK project 548214.

M. Böhm, J. Sgall, P. Veselý
Computer Science Institute of Charles University, Prague, Czech Republic. E-mail: {bohm,sgall,vesely}@iuuk.mff.cuni.cz.

R. van Stee
Department of Computer Science, University of Leicester, Leicester, UK. E-mail: rob.vanstee@leicester.ac.uk.

at most m bins, and (ii) the packing algorithm can use bins of capacity R for some $R \geq 1$. The goal is to minimize the stretching factor R.

In general, the term "semi-online" refers to algorithms that, compared to online algorithms, have some additional global information about the instance of the problem or have another advantage. We have formulated ONLINE BIN STRETCHING as a semi-online bin packing variant, where the algorithm has the additional information that the optimal number of bins is m and at the same time the algorithm has the advantage of using bins of larger capacity.

Taking another view, ONLINE BIN STRETCHING can also be thought of as a semi-online scheduling problem, in which we schedule jobs arriving one by one in an online manner on exactly m machines, with the objective to minimize the makespan, i.e., the length of the resulting schedule. Here the additional information is that the optimal offline algorithm could schedule all jobs with makespan 1. Our task is to present an algorithm with makespan being at most R.

History. ONLINE BIN STRETCHING has been proposed by Azar and Regev [3,4]. The original lower bound of $4/3$ for three bins has appeared even before that, in [16], for two bins together with a matching algorithm. Azar and Regev extended the same lower bound to any number of bins and gave an online algorithm with a stretching factor 1.625.

The problem has been revisited recently, with both lower bound improvements and new efficient algorithms. On the algorithmic side, Kellerer and Kotov [15] have achieved a stretching factor $11/7 \approx 1.57$ and Gabay et al. [12] have achieved $26/17 \approx 1.53$. There is still a considerable gap to the best known general lower bound of $4/3$, shown by a simple argument in the original paper of Azar and Regev [3,4].

While the algorithms and the lower bound from the last paragraph work for any $m \geq 2$, ONLINE BIN STRETCHING has also been studied in the special case of a specific constant number of bins m. In some cases better algorithms and lower bounds are possible. Obviously, for $m = 1$, the problem is trivial, as the only possible algorithm has a stretching factor 1. For $m = 2$, it is easy to achieve a stretching factor $4/3$, which is optimal. Thus the first interesting case is $m = 3$; the currently best algorithm given by the authors of this paper [7] has a stretching factor of $11/8 = 1.375$.

Interestingly, the setting with a small fixed number of bins allows better lower bounds on the stretching factor. The lower bounds cannot be easily translated into a lower bound for a larger m; for example, if we modify the instance by adding new bins and a corresponding number of items of size 1 (that must use exactly the new bins in the optimum), the semi-online algorithm still could use the additional capacity of $R - 1$ in the new bins to its advantage. The paper of Gabay et al. [10] showed a lower bound of $19/14 \approx 1.357$ for $m = 3$ using a computer search. Extending their methods, the authors of this paper were able to reach a lower bound of $15/11 = 1.\overline{36}$ for $m = 3$ in [7] as well as a bound of $19/14$ for $m = 4$ and $m = 5$. The preprint [10] was updated in 2015 [11] to include a lower bound of $19/14$ for $m = 4$ bins. For $m > 5$, the general lower bound of $4/3$ is still the best currently known.

Our contributions. We present a new algorithm for ONLINE BIN STRETCHING with a stretching factor of 1.5. We build on the two-phase approach which appeared previously in [15,12]. In this approach, the first phase tries to fill some bins close to $R - 1$

and achieve a fixed ratio between these bins and empty bins, while the second phase uses the bins in blocks of fixed size and analyzes each block separately. To reach 1.5, we needed to significantly improve the analysis using amortization techniques (represented by a weight function in our presentation) to amortize among blocks and bins of different types.

A preliminary version of this work appeared in [6].

Related work. The NP-hard problem BIN PACKING was originally proposed by Ullman [18] and Johnson [14] in the 1970s. Since then it has seen major interest and progress, see the survey of Coffman et al. [8] for many results on classical BIN PACKING and its variants. While our problem can be seen as a variant of BIN PACKING, note that the algorithms cannot open more bins than the optimum and thus general results for BIN PACKING do not translate to our setting.

As noted, ONLINE BIN STRETCHING can be formulated as online scheduling on m identical machines with known optimal makespan. Such algorithms were studied and are important in designing constant-competitive algorithms without the additional knowledge, e.g., for scheduling in the more general model of uniformly related machines [2,5,9].

For scheduling, also other types of semi-online algorithms are studied. Historically first is the study of ordered sequences with non-decreasing processing times [13]. Most closely related is the variant with known sum of all processing times studied in [16]. If we know the optimal makespan, we can always pad the instance by small items at the end so that the optimal makespan remains the same and the sum of processing time equals m times the optimal makespan. Thus one could expect that these two quantities are interchangeable. However, when the sum of all processing times is known, the currently best results are a lower bound of 1.585 and an algorithm with ratio 1.6, both from [1]. This shows, somewhat surprisingly, that knowing the actual optimum gives a significantly bigger advantage to the semi-online algorithm over knowing just the sum of the processing times. See Pruhs et al. [17] for a survey of other results on (semi-)online scheduling.

1.1 Definitions and notation

Our main problem, ONLINE BIN STRETCHING, can be described as follows:
Input: an integer m and a sequence of items $I = i_1, i_2, \ldots$ given online one by one. Each item has a *size* $s(i) \in [0,1]$ and must be packed immediately and irrevocably.
Parameter: The *stretching factor* $R \geq 1$.
Output: Partitioning (packing) of I into bins $B_1, \ldots, B_m$ so that $\sum_{i \in B_j} s(i) \leq R$ for all $j = 1, \ldots, m$.
Guarantee: there exists a packing of all items in I into m bins of capacity 1.
Goal: Design an online algorithm with the stretching factor R as small as possible which packs all input sequences satisfying the guarantee.

For a bin B, we define the *size of the bin* $s(B) = \sum_{i \in B} s(i)$. Unlike $s(i)$, $s(B)$ can change during the course of the algorithm, as we pack more and more items into the bin. To easily differentiate between items, bins and lists of bins, we use lowercase letters for items (i, b, x), uppercase letters for bins and other sets of items (A, B, X), and calligraphic letters for lists of bins ($\mathscr{A}$, $\mathscr{C}$, $\mathscr{L}$).

2 Algorithm

We rescale the sizes of items and the capacities of bins so that the optimal bins have capacity 12 and the bins of the algorithm have capacity 18.

We follow the general two-phase scheme of recent results [15, 12] which we sketch now. In the first phase of the algorithm we try to fill the bins so that their size is at most 6, as this leaves space for an arbitrary item in each bin. Of course, if items larger than 6 arrive, we need to pack them differently, preferably in bins of size at least 12 (since that is the size of the optimal bins). We stop the first phase when the number of non-empty bins of size at most 6 is three times the number of empty bins. In the second phase, we work in blocks consisting of three non-empty bins and one empty bin. The goal is to show that we are able to fill the bins so that the average size is at least 12, which guarantees we are able to pack the total size of $12m$ which is the upper bound on the size of all items.

The limitation of the previous results using this scheme was that the volume achieved in a typical block of four bins is slightly less than four times the size of the optimal bin, which then leads to bounds strictly above $3/2$. This is also the case in our algorithm: A typical block may have three bins with items of size just above 4 from the first phase plus one item of size 7 from the second phase, while the last bin contains two items, each of size 7, from the second phase—a total of 47 instead of desired $4 \cdot 12$. However, we notice that such a block contains five items of size 7 which the optimum cannot fit into four bins. To take an advantage of this, we cannot analyze each block separately as in [15, 12]. Instead, the rough idea of our improved method is to show that a bin with no item of size more than 6 typically has size at least 13 and amortize among the blocks of different types. Technically this is done using a weight function w that takes into account both the total size of items and the number of items larger than 6. This is the main new technical idea of our proof.

There are other complications. We would like to ensure that a typical bin of size at most 6 has size at least 4 after the first phase. However, this is impossible to guarantee if the items packed there have size between 3 and 4. Larger items are fine, as one per bin is sufficient, and the smaller ones are fine as well as we can always fit at least two of them. It is crucial to consider the items with sizes between 3 and 4 very carefully. This motivates our classification of items: Only the *regular items* of size in $(0,3] \cup (4,6]$ are packed in the bins filled up to size 6. The *medium items* of size in $(3,4]$ are packed in their own bins (four or five per bin). Similarly, *large items* of size in $(6,9]$ are packed in pairs in their own bins. Finally, the *huge items* of size larger than 9 are handled similarly as in the previous papers: If possible, they are packed with the regular items, otherwise each in their own bin.

The introduction of medium size items implies that we need to revisit the analysis of the first phase and also of the case when the first phase ends with no empty bin. These parts of the proof are similar to the previous works, but due to the new item type we need to carefully revisit it; it is now convenient to introduce another function

v that counts the items according to their type; we call it a value, to avoid confusion with the main weight function w. The analysis of the second phase when empty bins are present is more complicated, as we need to take care of various degenerate cases, and it is also here where the novel amortization is used.

Now we are ready to proceed with the formal definitions, statement of the algorithm, and the proof our main result.

Theorem 2.1 *There exists an algorithm for* ONLINE BIN STRETCHING *with a stretching factor of* 1.5 *for an arbitrary number of bins.*

We take an instance with an optimal packing into m bins of size at most 12 and, assuming that our algorithm fails, we derive a contradiction. One way to get a contradiction is to show that the size of all items is larger than $12m$. As stated above, we also use two bounds in the spirit of weight functions: weight $w(i)$ and value $v(i)$. The weight $w(i)$ is a slightly modified size to account for items of size larger than 6. The value $v(i)$ only counts the number of items with relatively large sizes. For our calculations, it is convenient to normalize the weight $w(i)$ and value $v(i)$ so that they are at most 0 for bins in the optimal packing (see Lemma 2.3). To get a contradiction, it is then sufficient to prove that the total weight or value of all bins is positive.

We classify the items based on their size $s(i)$ and define their value $v(i)$ as follows.

$s(i)$	$(9,12]$	$(6,9]$	$(3,4]$	$(0,3] \cup (4,6]$
type	huge	large	medium	regular
$v(i)$	3	2	1	0

Definition 2.2 For a set of items A, we define the value $v(A) = (\sum_{i \in A} v(i)) - 3$.

Furthermore we define weight $w(A)$ as follows. Let $k(A)$ be the number of large and huge items in A. Then $w(A) = s(A) + k(A) - 13$.

For a set of bins $\mathscr{A}$ we define $v(\mathscr{A}) = \sum_{A \in \mathscr{A}} v(A)$, $w(\mathscr{A}) = \sum_{A \in \mathscr{A}} w(A)$ and $k(\mathscr{A}) = \sum_{A \in \mathscr{A}} k(A)$.

Lemma 2.3 *For any packing $\mathscr{A}$ of a valid input instance into m bins of an arbitrary capacity, we have $w(\mathscr{A}) \leq 0$ and $v(\mathscr{A}) \leq 0$.*

Proof For the value $v(\mathscr{A})$, no optimal bin can contain items with the sum of their values larger than 3. The bound follows by summing over all bins and the fact that the number of bins is the same for the optimum and the considered packing.

For $w(\mathscr{A})$, we have $s(\mathscr{A}) \leq 12m$ and $k(\mathscr{A}) \leq m$, as the optimum packs all items in m bins of volume 12 and no bin can contain two items larger than 6. Thus $w(\mathscr{A}) = s(\mathscr{A}) + k(\mathscr{A}) - 13m \leq 12m + m - 13m = 0$. □

First phase. During the first phase, our algorithm maintains the invariant that only bins of the following types exist. See Figure 1 for an illustration of the bin types.

Definition 2.4 Given a bin A, we define the following bin types and introduce letters that typically denote those bins:

- **Empty bins (E):** bins that have no item.

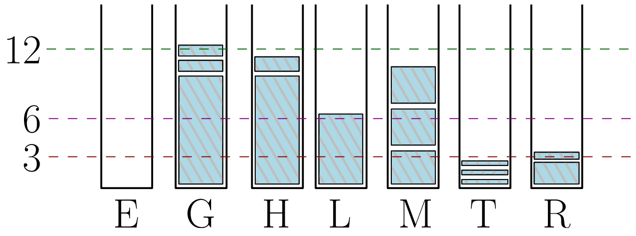

Fig. 1 An illustration of bin types during the first phase.

- **Complete bins (G):** all bins that have $w(A) \geq 0$ and $s(A) \geq 12$;
- **Huge-item bins (H):** all bins that contain a huge item (plus possibly some other items) and have $s(A) < 12$;
- **One large-item bin (L):** a bin containing only a single large item;
- **One medium-item bin (M):** a non-empty bin with $s(A) < 13$ and only medium items;
- **One tiny bin (T):** a non-empty bin with $s(A) \leq 3$;
- **Regular bins (R):** all other bins with $s(A) \in (3,6]$;

First-phase algorithm:
During the algorithm, let e be the current number of empty bins and r the current number of regular bins.

```
(1)  While $r < 3e$, consider the next item $i$ and pack it as follows, using bins of
     capacity 18; if more bins satisfy a condition, choose among them arbitrarily:
(2)      If $i$ is regular:
(3)          If there is a huge-item bin, pack $i$ there.
(4)          Else, if there is a regular bin $A$ with $s(A)+s(i) \leq 6$, pack it there.
(5)          Else, if there is a tiny bin $A$ with $s(A)+s(i) \leq 6$, pack it there.
(6)      If $i$ is medium and there is a medium-item bin where $i$ fits, pack it there.
(7)      If $i$ is large and there is a large-item bin where $i$ fits, pack it there.
(8)      If $i$ is huge:
(9)          If there is a regular bin, pack $i$ there.
(10)         Else, if there is a tiny bin, pack $i$ there.
(11)     If $i$ is still not packed, pack it in an empty bin.
```

First we observe that the algorithm described in the box above is properly defined. The stopping condition guarantees that the algorithm stops when no empty bin is available. Thus an empty bin is always available and each item i is packed. We now state the basic properties of the algorithm.

Lemma 2.5 *At any time during the first phase the following holds:*
(i) *All bins used by the algorithm are of the types from Definition 2.4.*
(ii) *All complete bins B have $v(B) \geq 0$.*
(iii) *If there is a huge-item bin, then there is no regular and no tiny bin.*
(iv) *There is at most one large-item bin and at most one medium-item bin.*

(v) *There is at most one tiny bin T. If T exists, then for any regular bin, $s(T)+s(R)>6$. There is at most one regular bin R with $s(R)\le 4$.*
(vi) *At the end of the first phase $3e \le r \le 3e+3$.*

Proof (i)-(v): We verify that these invariants are preserved when an item of each type arrives and also that the resulting bin is of the required type; the second part is always trivial when packing in an empty bin.

If a huge item arrives and a regular bin exists, it always fits there, thus no huge-item bin is created and (iii) cannot become violated. Furthermore, the resulting size is more than 12, thus the resulting bin is complete. Otherwise, if a tiny bin exists, the huge item fits there and the resulting bin is either complete or huge. In either case, if the bin is complete, its value is 0 as it contains a huge item.

If a large item arrives, it always fits in a large-item bin if it exists and makes it complete; its value is at least 1, as it contains two large items. Thus a second large-item bin is never created and (iv) is not violated.

If a medium item arrives, it always fits in a medium-item bin if it exists; the bin is then complete if it has size at least 13 and then its value is at least 1, as it contains 4 or 5 medium items; otherwise the bin type is unchanged. Again, a second medium-item bin is never created and (iv) is not violated.

If a regular item arrives and a huge-item bin exists, it always fits there, thus no regular bin is created and (iii) cannot become violated. Furthermore, if the resulting size is at least 12, the bin becomes complete and its value is 0 as it contains a huge item; otherwise the bin type is unchanged.

In the last case, a regular item arrives and no huge-item bin exists. The algorithm guarantees that the resulting bin has size at most 6, thus it is regular or tiny. We now proceed to verify (v). A new tiny bin T can be created only by packing an item of size at most 3 in an empty bin. First, this implies that no other tiny bin exists, as the item would be put there, thus there is always at most one tiny bin. Second, as the item is not put in any existing regular bin R, we have $s(R)+s(T)>6$ and this also holds later when more items are packed into any of these bins. A new regular bin R with $s(R)\le 4$ can be created only from a tiny bin; note that a bin created from an empty bin by a regular item is either tiny or has size in $(4,6]$. If another regular bin with size at most 4 already exists, then both the size of the tiny bin and the size of the new item are larger than 2 and thus the new regular bin has size more than 4. This completes the proof of (v).

(vi): Before an item is packed, the value $3e-r$ is at least 1 by the stopping condition. Packing an item may change e or r (or both) by at most 1. Thus after packing an item we have $3e-r \ge 1-3-1=-3$, i.e., $r \le 3e+3$. If in addition $3e \le r$, the algorithm stops in the next step and (vi) holds. □

If the algorithm packs all items in the first phase, it stops. Otherwise according to Lemma 2.5(iii) we split the algorithm in two very different branches. If there is at least one huge-item bin, follow the second phase with huge-item bins below. If there is no huge-item bin, follow the second phase with regular bins.

Any bin that is complete is not used in the second phase. In addition to complete bins and either huge-item bins, or regular and empty bins, there may exist at most three *special bins* denoted and ordered as follows: the large-item bin L, the medium-item bin M, and the tiny bin T.

Second phase with huge-item bins. In this case, we assume that a huge-item bin exists when the first phase ends. By Lemma 2.5(iii), we know that no regular and tiny bins exist. There are no empty bins either, as we end the first phase with $3e \leq r = 0$. With only a few types of bins remaining, the algorithm for this phase is very simple:

Algorithm for the second phase with huge-item bins:
Let the list of bins $\mathscr{L}$ contain first all the huge-item bins, followed by the special bins L, M, in this order, if they exist.

```
For any incoming item $i$:
    Pack $i$ using First Fit on the list $\mathscr{L}$, with all bins of capacity 18.
```

Suppose that we have an instance that has a packing into bins of capacity 12 and on which our algorithm fails. We may assume that the algorithm fails on the last item f. By considering the total volume, there always exists a bin with size at most 12. Thus $s(f) > 6$ and $v(f) \geq 2$.

If during the second phase an item n with $s(n) \leq 6$ is packed into the last bin in $\mathscr{L}$, we know that all other bins have size more than 12, thus all the remaining items fit into the last bin. Otherwise we consider $v(\mathscr{L})$. Any complete bin B has $v(B) \geq 0$ by Lemma 2.5(ii) and each huge-item bin gets nonnegative value, too. Also $v(L) \geq -1$ if L exists. This shows that M must exist, since otherwise $v(\mathscr{L}) + v(f) \geq -1 + 2 \geq 1$, a contradiction.

Now we know that M exists, furthermore it is the last bin and thus we also know that no regular item is packed in M. Therefore M contains only medium items from the first phase and possibly large and/or huge items from the second phase. We claim that $v(M) + v(f) \geq 2$ using the fact that f does not fit into M and M contains no item a with $v(a) = 0$: If f is huge we have $s(M) > 6$, thus M must contain either two medium items or at least one medium item together with one large or huge item and $v(M) \geq -1$. If f is large, we have $s(M) > 9$; thus M contains either three medium items or one medium and one large or huge item and $v(M) \geq 0$. Thus we always have $v(\mathscr{L}) \geq -1 + v(M) + v(f) \geq 1$, a contradiction.

Second phase with regular bins. Let $\mathscr{E}$ resp. $\mathscr{R}$ be the set of empty resp. regular bins at the beginning of the second phase, and let $e = |\mathscr{E}|$. Let $\lambda \in \{0, 1, 2, 3\}$ be such that $|\mathscr{R}| = 3e + \lambda$; Lemma 2.5(vi) implies that λ exists. Note that it is possible that $\mathscr{R} = \emptyset$, in that case $e = \lambda = 0$.

We organize the remaining non-complete bins into blocks $\mathscr{B}_i$, and then order them into a list $\mathscr{L}$, as follows:

Definition 2.6 Denote the empty bins $E_1, E_2, \ldots, E_e$. The regular bins are denoted by $R_{i,j}$, $i = 1, ..., e+1$, $j = 1, 2, 3$. The ith **block** $\mathscr{B}_i$ consists of bins $R_{i,1}, R_{i,2}, R_{i,3}, E_i$ in this order. There are several modifications to this rule:

(1) The first block $\mathscr{B}_1$ contains only λ regular bins, i.e., it contains $R_{1,1}, \ldots, R_{1,\lambda}, E_1$ in this order; in particular, if $\lambda = 0$ then $\mathscr{B}_1$ contains only E_1.

(2) The last block $\mathscr{B}_{e+1}$ has no empty bin, only exactly 3 regular bins.

(3) If $e = 0$ and $r = \lambda > 0$ we define only a single block $\mathscr{B}_1$ which contains $r = \lambda$ regular bins $R_{1,1}, \ldots, R_{1,\lambda}$.
(4) If $e = r = \lambda = 0$, there is no block, as there are no empty and regular bins.
(5) If $r > 0$, we choose as the first regular bin the one with size at most 4, if there is such a bin.

Denote the first regular bin by R_{first}. If no regular bin exists (i.e., if $r = 0$), R_{first} is undefined.

Note that R_{first} is either the first bin $R_{1,1}$ in $\mathscr{B}_1$ if $\lambda > 0$ or the first bin $R_{2,1}$ in $\mathscr{B}_2$ if $\lambda = 0$. By Lemma 2.5(v) there exists at most one regular bin with size at most 4, thus all the remaining $R_{i,j} \neq R_{\text{first}}$ have $s(R_{i,j}) > 4$.

Definition 2.7 The **list of bins** $\mathscr{L}$ we use in the second phase contains first the special bins and then all the blocks $\mathscr{B}_1, \ldots, \mathscr{B}_{e+1}$. Thus the list $\mathscr{L}$ is (some or all of the first six bins may not exist):

$$L, M, T, R_{1,1}, R_{1,2}, R_{1,3}, E_1, R_{2,1}, R_{2,2}, R_{2,3}, E_2, \ldots, E_e, R_{e+1,1}, R_{e+1,2}, R_{e+1,3}.$$

Whenever we refer to the ordering of the bins, we mean the ordering in the list $\mathscr{L}$. See Figure 2 for an illustration.

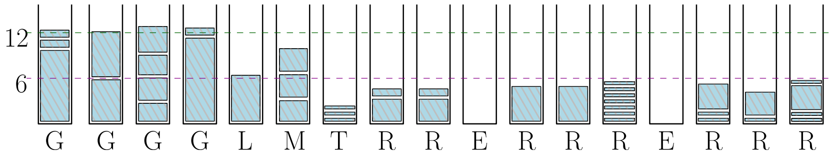

Fig. 2 A typical state of the algorithm after the first phase. The bin labels correspond to the particular bin types. G denotes complete bins, other labels are the initial letters of the bin types. The non-complete bins (other than G) are ordered as in the list $\mathscr{L}$ at the beginning of the second phase with regular bins.

```
Algorithm for the second phase with regular bins:
Let $\mathscr{L}$ be the list of bins as in Definition 2.7, with all bins of capacity 18.
(1) For any incoming item $i$:
(2)     If $i$ is huge, pack it using First Fit on the reverse of the list $\mathscr{L}$.
(3)     In all other cases, pack $i$ using First Fit on the normal list $\mathscr{L}$.
```

Suppose that we have an instance that has a packing into bins of capacity 12 and on which our algorithm fails. We may assume that the algorithm fails on the last item. Let us denote this item by f. We have $s(f) > 6$, as otherwise all bins have size more than 12, contradicting the existence of optimal packing. Call the items that arrived in the second phase *new* (including f), the items from the first phase are *old*. See Figure 3 for an illustration of a typical final situation (and also of notions that we introduce later).

Our overall strategy is to obtain a contradiction by showing that

$$w(\mathscr{L}) + w(f) > 0.$$

In some cases, we instead argue that $v(\mathscr{L})+v(f)>0$ or $s(\mathscr{L})+s(f)>12|\mathscr{L}|$. Any of these is sufficient for a contradiction, as all complete bins have both value and weight nonnegative and size at least 12.

Let $\mathscr{H}$ denote all the bins from $\mathscr{L}$ with a huge item, and let $h=|\mathscr{H}| \bmod 4$. First we show that the average size of bins in $\mathscr{H}$ is large and exclude some degenerate cases; in particular, we exclude the case when no regular bin exists.

Lemma 2.8 *Let ρ be the total size of old items in R_{first} if $\mathscr{R}\neq\emptyset$ and $R_{\mathrm{first}}\in\mathscr{H}$, otherwise set $\rho=4$.*

(i) *The bins $\mathscr{H}$ are a final segment of the list $\mathscr{L}$ and $\mathscr{H}\subsetneq\mathscr{E}\cup\mathscr{R}$. In particular, $\mathscr{R}\neq\emptyset$ and R_{first} is defined.*
(ii) *We have $s(\mathscr{H})\geq 12|\mathscr{H}|+h+\rho-4$.*
(iii) *If $\mathscr{H}$ does not include R_{first}, then $s(\mathscr{H})\geq 12|\mathscr{H}|+h\geq 12|\mathscr{H}|$.*
(iv) *If $\mathscr{H}$ includes R_{first}, then $s(\mathscr{H})\geq 12|\mathscr{H}|+h-1\geq 12|\mathscr{H}|-1$.*

Proof First we make an easy observation used later in the proof. If $\mathscr{E}\cup\mathscr{R}\cup\{T\}$ contains a bin B with no huge item, then no bin preceding B contains a huge item. Indeed, if a huge item does not fit into B, then B must contain a new item i of size at most 9. This item i was packed using First Fit on the normal list $\mathscr{L}$, and therefore it did not fit into any previous bin. Thus the huge item also does not fit into any previous bin, and cannot be packed there.

Let $\mathscr{H}'=\mathscr{H}\cap(\mathscr{E}\cup\mathscr{R})$. We begin proving our lemma for $\mathscr{H}'$ in place of $\mathscr{H}$. That is, we ignore the special bins at this stage. The previous observation shows that $\mathscr{H}'$ is a final segment of the list.

We now prove the claims (ii)–(iv) with $\mathscr{H}'$ in place of $\mathscr{H}$. All bins $R_{i,j}$ with a huge item have size at least $4+9=13$, with a possible exception of R_{first} which has size at least $\rho+9=13+\rho-4$, by the definition of ρ. Each E_i with a huge item has size at least 9. Thus for each i with $E_i\in\mathscr{H}'$, $s(E_i)+s(R_{i+1,1})+s(R_{i+1,2})+s(R_{i+1,3})\geq 4\cdot 12$, with a possible exception of $i=1$ in the case when $\lambda=0$. Summing over all i with $E_i\in\mathscr{H}'$ and the h bins in $\mathscr{R}$ from the first block intersecting $\mathscr{H}'$, and adjusting for R_{first} if $R_{\mathrm{first}}\in\mathscr{H}'$, (ii) for $\mathscr{H}'$ follows. The claims (iii) and (iv) for $\mathscr{H}'$ are an immediate consequence as $\rho>3$ if $R_{\mathrm{first}}\in\mathscr{H}'$ and $\rho=4$ otherwise.

We claim that the lemma for $\mathscr{H}$ follows if $\mathscr{H}'\subsetneq\mathscr{E}\cup\mathscr{R}$. Indeed, following the observation at the beginning of the proof, the existence of a bin in $\mathscr{E}\cup\mathscr{R}$ with no huge item implies that no special bin has a huge item, i.e., $\mathscr{H}'=\mathscr{H}$, and also $\mathscr{H}'=\mathscr{H}$ is a final segment of $\mathscr{L}$. Furthermore, the existence of a bin in $\mathscr{E}\cup\mathscr{R}$ together with $3e\leq r$ implies that there exist at least one regular bin, thus also R_{first} is defined and (i) follows. Claims (ii), (iii), and (iv) follow from $\mathscr{H}'=\mathscr{H}$ and the fact that we have proved them for $\mathscr{H}'$.

Thus it remains to show that $\mathscr{H}'\subsetneq\mathscr{E}\cup\mathscr{R}$. Suppose for a contradiction that $\mathscr{H}'=\mathscr{E}\cup\mathscr{R}$.

If T exists, let o be the total size of old items in T. If also R_{first} exists, Lemma 2.5(v) implies that $o+\rho>6>4$, otherwise $o+\rho>4$ trivially. In either case, summing with (ii) we obtain

$$o+s(\mathscr{H}')>12|\mathscr{H}'|\,. \tag{1}$$

Now we proceed to bound $s(\mathscr{H})$. We have already shown claim (iv) for $\mathscr{H}'$, i.e., $s(\mathscr{H}')\geq 12|\mathscr{H}'|-1$. If L or M has a huge item, the size of the bin is at least 12, as there is an old large or medium item in it. If T has a huge item, then (1) implies $s(T)+s(\mathscr{H}')>9+o+s(\mathscr{H}')>9+12|\mathscr{H}'|$. Summing these bounds we obtain

$$s(\mathscr{H})>12|\mathscr{H}|-3\,. \tag{2}$$

We now derive a contradiction in each of the following four cases.

Case 1: All special bins have a huge item. Then $\mathscr{L}=\mathscr{H}$ and (2) together with $s(f)>6$ implies $s(\mathscr{L})+s(f)>12|\mathscr{L}|+3$, a contradiction.

Case 2: There is one special bin with no huge item. Then its size together with f is more than 18, thus (2) together with $s(f)>6$ implies $s(f)+s(\mathscr{L})>18+12|\mathscr{H}|-3>12|\mathscr{L}|$, a contradiction.

Case 3: There are two special bins with no huge item and these bins are L and M.

Suppose first that M contains a new item n. Then $s(L)+s(n)>18$ by the First Fit packing rule. The bin M contains at least one old medium item. Thus, using (2), we get $s(f)+s(\mathscr{L})>6+s(L)+s(n)+3+12|\mathscr{H}|-3>24+12|\mathscr{H}|=12|\mathscr{L}|$, a contradiction.

If M has no new item, then either f is huge and M has at least two medium items, or f is large and M has at least three items. In both cases $v(M)+v(f)\geq 2$. Also $v(L)\geq -1$ since L has a large item, and $v(\mathscr{H})\geq 0$ as each bin has a huge item. Altogether we get that the total value $v(\mathscr{L})>0$, a contradiction.

Case 4: There are two or three special bins with no huge item, one of them is T. Observe that the bin T always contains a new item n, as the total size of all old items in it is at most 3.

If there are two special bins with no huge item, denote the first one by B. As we observed at the beginning of the proof, if T exists and has no huge item, no special bin can contain a huge item, thus the third special bin cannot exist. We have $s(B)+s(n)>18$, summing with (1) we obtain $s(f)+s(\mathscr{L})\geq s(f)+s(B)+s(n)+o+s(\mathscr{H}')>6+18+12|\mathscr{H}'|=12|\mathscr{L}|$, a contradiction.

If there are three special bins with no huge item, we have $s(f)+s(L)>18$ and $s(M)+s(n)>18$. Summing with (1) we obtain $s(f)+s(\mathscr{L})>18+18+o+s(\mathscr{H}')>36+12|\mathscr{H}'|=12|\mathscr{L}|$, a contradiction. □

Having proven Lemma 2.8, we can infer existence of the following two important bins, which (as we will later see) split the instance into three logical blocks:

Definition 2.9

- Let F, the **final bin** be the last bin in $\mathscr{L}$ before $\mathscr{H}$, or the last bin if $\mathscr{H}=\emptyset$.
- Let C, the **critical bin**, be the first bin in $\mathscr{L}$ of size at most 12.

First note that both F and C must exist: F exists by Lemma 2.8(i), which also shows that $F\in\mathscr{E}\cup\mathscr{R}$. C exists, as otherwise the total size is more than $12m$.

To make our calculations easier, we modify the packing so that f is put into F, even though it exceeds the capacity of F. Thus $s(F)>18$ and f (a new item) as well as all the other new items packed in F or in some bin before F satisfy the property that they do not fit into any previous bin. See Figure 3 for an illustration of the definitions.

We start by some easy observations. Each bin, possibly with the exception of L and M, contains a new item, as it enters the phase with size at most 6, and the algorithm failed. Only items of size at most 9 are packed in bins before F; in F itself only the item f can be huge. The bin F always has at least two new items, one that did fit into it and f. All the new items in the bins after C are large, except for the new huge items in $\mathscr{H}$ and f which can be large or huge. (Note that at this point of the proof it is possible that C is after F; we will exclude this possibility soon.)

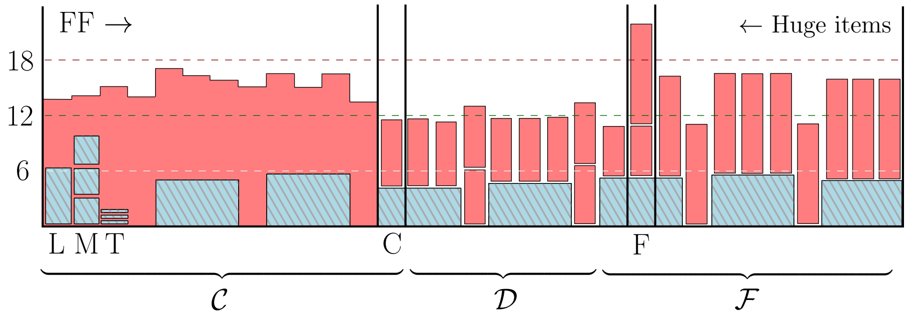

Fig. 3 A typical state of the algorithm after the second phase with regular bins. The gray (hatched) areas denote the old items (i.e., packed in the first phase), the red (solid) regions and rectangles denote the new items (i.e., packed in the second phase). The bins that are complete at the end of the first phase are not shown. The item f on which the algorithm fails is shown as packed into the final bin F and exceeding the capacity 18, following the convention introduced after Definition 2.9.

More observations are given in the next two lemmata.

Lemma 2.10 *(i) Let B be any bin before F. Then $s(B) > 9$. Furthermore, if $B \in \mathscr{E}$ then B contains at least two new items.*

(ii) Let B, B', B'' be any three bins in this order before than or equal to F and let B'' contain at least two new items. Then $s(B) + s(B') + s(B'') > 36 + o$, where o is the size of old items in B''.

(iii) Let B be arbitrary and let $B' \in \mathscr{R}$ be an arbitrary bin after B and before than or equal to F in $\mathscr{L}$.
If $B' \neq R_{\text{first}}$ then $s(B) + s(B') > 22$, in particular $s(B) > 11$ or $s(B') > 11$.
If $B' = R_{\text{first}}$ then $s(B) + s(B') > 21$.

Proof F contains a new item n different from f. To prove (i), note that $s(n) \leq 9$, and n does not fit into B. It follows that if $B \in \mathscr{E}$, then B must contain at least two new items, as only items with size smaller than 9 are packed before F.

To prove (ii), let n, n' be two new items in B'' and note that $s(B) + s(n) > 18$ and $s(B') + s(n') > 18$.

To prove (iii), observe that B' has a new item of size larger than $18 - s(B)$, and it also has old items of size at least 3 or even 4 if $B' \neq R_{\text{first}}$. □

Lemma 2.11 *The critical bin C is before F, there are at least two bins between C and F and C is not in the same block as F.*

Proof All bins before C have size larger than 12. Using Lemma 2.8 we have

$$s(F) + s(\mathscr{H}) > 18 + 12|\mathscr{H}| - 1 = 12(|\mathscr{H}| + 1) + 5\,.$$

It remains to bound the sizes of the other bins. Note that $F \neq C$ as $s(F) > 18$. If C is after F, all bins before F have size more than 12, so all together $s(\mathscr{L}) > 12|\mathscr{L}| + 5$, a contradiction. If C is just before F, then by Lemma 2.10(i), $s(C) > 9 = 12 - 3$ and the total size of bins in $s(\mathscr{L}) > 12|\mathscr{L}| + 5 - 3 > 12|\mathscr{L}|$, a contradiction.

If there is a single bin B between C and F, then $s(C) + s(B)$ plus the size of two new items in F is more than 36 by Lemma 2.10(ii). If $F \in \mathscr{E}$ then $\mathscr{H}$ starts with three bins in $\mathscr{R}$, thus $s(\mathscr{H}) \geq 12|\mathscr{H}| + 2$ using Lemma 2.8 with $h = 3$, and we get a contradiction. If $F \in \mathscr{R}$ then $R_{\text{first}} \notin \mathscr{H}$, thus Lemma 2.8 gives $s(\mathscr{H}) \geq 12|\mathscr{H}|$, and we get a contradiction as well.

The last case is when C and F are in the same block with two bins between them. Then $F \in \mathscr{E}$, so $h = 3$, and C is the first bin of the three other bins from the same block, so $R_{\text{first}} \notin \mathscr{H}$. Then $s(C) > 9$, the remaining two bins together with F have size more than 36 by Lemma 2.10(ii) and we use $s(\mathscr{H}) \geq 12|\mathscr{H}| + 3$ from Lemma 2.8 to get a contradiction. □

We now partition $\mathscr{L}$ into several parts (see Figure 3 for an illustration):

Definition 2.12
- Let $\mathscr{F} = \mathscr{B}_i \cup \mathscr{H}$, where $F \in \mathscr{B}_i$.
- Let $\mathscr{D}$ be the set of all bins after C and before $\mathscr{F}$.
- Let $\mathscr{C}$ be the set of all bins before and including C.

Lemma 2.11 shows that the parts are non-overlapping. We analyze the weight of the parts separately, essentially block by block. Recall that a weight of a bin is defined as $w(A) = s(A) + k(A) - 13$, where $k(A)$ is the number of large and huge items packed in A. The proof is relatively straightforward if C is not special (and thus also $F \notin \mathscr{B}_1$), which is the most important case driving our choices for w. A typical block has nonnegative weight, we gain more weight in the block of F which exactly compensates the loss of weight in $\mathscr{C}$, which occurs mainly in C itself.

Let us formalize and prove the intuition stated in the previous paragraph in a series of three lemmata.

Lemma 2.13 *If F is not in the first block then $w(\mathscr{F}) > 5$, otherwise $w(\mathscr{F}) > 4$.*

Proof All the new items in bins of $\mathscr{F}$ are large or huge. Each bin has a new item and the bin F has two new items. Thus $k(\mathscr{F}) \geq |\mathscr{F}| + 1$. All that remains is to show that $s(\mathscr{F}) > 12|\mathscr{F}| + 3$, and $s(\mathscr{F}) > 12|\mathscr{F}| + 4$ if F is not in the first block.

If F is the first bin in a block, the lemma follows as $s(F) > 18$ and $s(\mathscr{H}) \geq 12|\mathscr{H}| - 1$, thus $s(\mathscr{F}) = s(F) + s(\mathscr{H}) > 12|\mathscr{F}| + 5$.

In the remaining cases there is a bin in $\mathscr{R} \cap \mathscr{F}$ before F. Lemma 2.8 gives $s(\mathscr{H}) \geq 12|\mathscr{H}|$; moreover, if $F \in \mathscr{E}$, then $s(\mathscr{H}) \geq 12|\mathscr{H}| + 3$.

If F is preceded by three bins from $\mathscr{F} \cap \mathscr{R}$, then $F \in \mathscr{E}$ and thus $s(\mathscr{H}) \geq 12|\mathscr{H}| + 3$. Using Lemma 2.10(iii) twice, two of the bins in $\mathscr{F} \cap \mathscr{R}$ before F have size at least 11 and using Lemma 2.10(i) the remaining one has size 9. Thus the size of these four bins is more than $11 + 11 + 9 + 18 = 4 \cdot 12 + 1$, summing with the bound for $\mathscr{H}$ we get $s(\mathscr{F}) > 12|\mathscr{F}| + 4$.

If F is preceded by two bins from $\mathscr{F} \cap \mathscr{R}$, then by Lemma 2.10(i) the total size of these two bins and two new items in F is more than 36. If $F \in \mathscr{R}$, the size of old items in F is at least 4 and with $s(\mathscr{H}) \geq 12|\mathscr{H}|$ we get $s(\mathscr{F}) > 12|\mathscr{F}| + 4$. If $F \in \mathscr{E}$, which also implies that F is in the first block, then $s(\mathscr{H}) \geq 12|\mathscr{H}| + 3$, thus $s(\mathscr{F}) > 12|\mathscr{F}| + 3$.

If F is preceded by one bin R from $\mathscr{F} \cap \mathscr{R}$, then let n be a new item in F different from f. We have $s(R) + s(n) > 18$ and $s(f) > 6$. We conclude the proof as in the previous case. □

Lemma 2.14
If $C \in \mathscr{R}$ then $w(\mathscr{C}) \geq -6$.
If $C \in \mathscr{E}$ then $w(\mathscr{C}) \geq -5$.
If C is a special bin then $w(\mathscr{C}) \geq -4$.

Proof For every bin B before C, $s(B) > 12$ and thus $w(B) > -1$ by the definition of C. Let $\mathscr{C}'$ be the set of all bins B before C with $w(B) \leq 0$. This implies that for $B \in \mathscr{C}'$, $s(B) \in (12, 13]$ and B has no large item. It follows that any new item in any bin after the first bin in $\mathscr{C}'$ has size more than 5. We have

$$w(\mathscr{C}) \geq w(\mathscr{C}') + w(C) \geq -|\mathscr{C}'| + w(C)\,. \tag{3}$$

First we argue that either $|\mathscr{C}'| \leq 1$ or $\mathscr{C}' = \{M, T\}$. Suppose that $|\mathscr{C}'| > 1$, choose $B, B' \in \mathscr{C}'$ so that B is before B'. If $B' \in \mathscr{E}$, either B' has at most two (new) non-large items and $s(B') \leq 6 + 6 = 12$, or it has at least three items and $s(B') > 5 + 5 + 5 = 15$; both options are impossible for $B' \in \mathscr{C}'$. If $B' \in \mathscr{R}$, it has old items of total size in $(3, 6]$. Either B' has a single new item and $s(B') \leq 6 + 6 = 12$, or it has at least two new items and $s(B') > 3 + 5 + 5 = 13$; both options are impossible for $B' \in \mathscr{C}'$. The only remaining option is that B' is a special bin. Since L has a large item, $L \notin \mathscr{C}'$ and $\mathscr{C}' = \{M, T\}$.

By Lemma 2.10(i), we have $w(C) \geq -4$. The lemma follows by summing with (3) in the following three cases: (i) $C \in \mathscr{R}$, (ii) if $\mathscr{C}' = \emptyset$ and also (iii) if both $C \in \mathscr{E}$ and $|\mathscr{C}'| = 1$.

For the remaining cases, (3) implies that it is sufficient to show $w(C) \geq -3$. If $C \in \mathscr{E}$ and $\mathscr{C}' = \{M, T\}$ then C contains two new items of size at least 5, thus $w(C) \geq -3$. If $C = T$ and $\mathscr{C}' = \{M\}$ then C either has a large item, or it has two new items: otherwise it would have size at most 3 of old items plus at most 6 from a single new item, total of at most 9, contradicting Lemma 2.10(i). Thus $w(C) \geq -3$ in this case as well. □

Lemma 2.15 (i) *For every block $\mathscr{B}_i \subseteq \mathscr{D}$ we have $w(\mathscr{B}_i) \geq 0$.*
(ii) *If there is no special bin in $\mathscr{D}$, then $w(\mathscr{D}) \geq 0$. If also $C \in \mathscr{R}$ then $w(\mathscr{D}) \geq 1$.*

Proof First we claim that for each block $\mathscr{B}_i \subseteq \mathscr{D}$ with three bins in $\mathscr{R}$, we have

$$w(\mathscr{B}_i) \geq 0\,. \tag{4}$$

By Lemma 2.10(iii), one of the bins in $\mathscr{R} \cap \mathscr{B}_i$ has size at least 11. By Lemma 2.10(ii), the remaining three bins have size at least 36. We get (4) by observing that $k(\mathscr{B}_i) \geq 5$, as all the new items placed after C and before F are large, each bin contains a new item and E_i contains two new items.

Next, we consider an incomplete block, that is, a set of bins $\mathscr{B}$ with at most two bins from $\mathscr{R} \cap \mathscr{D}$ followed by a bin $E \in \mathscr{E} \cap \mathscr{D}$. We claim

$$w(\mathscr{B}) \geq 1\,. \tag{5}$$

The bin E contains two large items, since it is after C. In particular, $w(E) \geq 1$ and (5) follows if $|\mathscr{B}| = 1$. If $|\mathscr{B}| = 2$, the size of one item from E plus the previous bin is more than 18, the size of the other item is more than 6, thus $s(\mathscr{B}) \geq 24$; since $k(B) \geq 3$, (5) follows. If $|\mathscr{B}| = 3$, by Lemma 2.10(ii) we have $s(\mathscr{B}) \geq 36$; $k(\mathscr{B}) \geq 4$ and (5) follows as well.

By definition, $\mathscr{D}$ ends by a bin in $\mathscr{E}$ (if nonempty). Thus the lemma follows by using (5) for the incomplete block, i.e., for $C \in \mathscr{R}$ or for $\mathscr{B}_1$ if it does not have three bins in $\mathscr{R}$, and adding (4) for all the remaining blocks. Note that $C \in \mathscr{R}$ implies $\mathscr{D} \neq \emptyset$. □

We are now ready to derive the final contradiction.

If $\mathscr{D}$ does not contain a special bin, we add the appropriate bounds from Lemmata 2.14, 2.15 and 2.13. If $C \in \mathscr{R}$ then F is not in the first block and $w(\mathscr{L}) = w(\mathscr{C}) + w(\mathscr{D}) + w(\mathscr{F}) > -6 + 1 + 5 = 0$. If $C \in \mathscr{E}$ then F is not in the first block and $w(\mathscr{L}) = w(\mathscr{C}) + w(\mathscr{D}) + w(\mathscr{F}) > -5 + 0 + 5 = 0$. If C is the last special bin then $w(\mathscr{L}) = w(\mathscr{C}) + w(\mathscr{D}) + w(\mathscr{F}) > -4 + 0 + 4 = 0$. In all subcases $w(\mathscr{L}) > 0$, a contradiction.

The rest of the proof deals with the remaining case when $\mathscr{D}$ does contain a special bin. This implies there are at least two special bins and C is not the last special bin. Since T is always the last special bin (if it exists), it must be the case that $C \neq T$ and thus $C = L$ or $C = M$. We analyze the special bins together with the first block, up to F if F belongs to it. First observe that the only bin possibly before C is L and in that case $w(L) \geq 0$, so $w(\mathscr{C}) \geq w(C)$.

Let A denote F if $F \in \mathscr{B}_1$ or E_1 if $F \notin \mathscr{B}_1$. As $A = F$ or $A \in \mathscr{E}$, we know that A contains at least two new items; denote two of these new items by n and n'. Since A is after C, we know that both n and n' are large or huge.

Let $\mathscr{A}$ be the set containing C and all bins between C and A, not including A. Thus $\mathscr{A}$ contains two or three special bins followed by at most three bins from $\mathscr{R}$. We have $k(\mathscr{A}) \geq |\mathscr{A}| - 1$ as each bin in $\mathscr{A}$ contains a large item, with a possible exception of C (if $C = M$). Furthermore $k(A) \geq 2$. The bound on $k(\mathscr{A})$ and $k(A)$ imply that

$$w(\mathscr{A}) + w(A) \geq s(\mathscr{A}) + s(A) - 12|\mathscr{A}| - 12 \tag{6}$$

and thus it is sufficient to bound $s(\mathscr{A}) + s(A)$.

The precise bound we need depends on what bin A is. In each case, we first determine a sufficient bound on $s(\mathscr{A}) + s(A)$ and argue that it implies contradiction. Afterwards we prove the bound. Typically, we bound the size by creating pairs of bins of size 21 or 22 by Lemma 2.10(iii). We also use that $s(B) > 9$ for any $B \in \mathscr{A}$ by Lemma 2.10(i) and that n, n' together with any two bins in $\mathscr{A}$ have size at least 36 by Lemma 2.10(ii).

Case $A \neq F$**:** Then $F \notin \mathscr{B}_1$ and $A = E_1$. We claim that

$$s(\mathscr{A}) + s(A) \geq 12|\mathscr{A}| + 7\,. \tag{7}$$

First we show that (7) implies a contradiction. Indeed, (7) together with (6) yields $w(\mathscr{A})+w(A)\geq -5$ and summing this with all the other bounds, namely $w(\mathscr{F})>5$ from Lemma 2.13 and $w(\mathscr{B}_i)\geq 0$ for whole blocks $\mathscr{B}_i\in\mathscr{D}$ from Lemma 2.15, leads to $w(\mathscr{L})>0$, which is a contradiction.

Now we prove (7). The items n and n' from A together with the first two special bins in $\mathscr{A}$ have size more than 36. Let $\mathscr{A}'$ be the set of the remaining bins; it contains possibly T and at most three bins from $\mathscr{R}$. It remains to show $s(\mathscr{A}')\geq 12|\mathscr{A}'|-5$.

For $|\mathscr{A}'|=0$ it holds trivially.

If $|\mathscr{A}'|=1$, the only bin in $\mathscr{A}'$ has size more than 9 and this is sufficient.

For $|\mathscr{A}'|>1$ we apply Lemma 2.10(iii) and pair as many bins from $\mathscr{A}'$ as possible; note that all the bins in $\mathscr{A}'$ except possibly T are in $\mathscr{R}$, so the assumptions of the lemma hold. If $|\mathscr{A}'|=2$, then $s(\mathscr{A}')>21=2\cdot 12-3$. For $|\mathscr{A}'|=3$ we get $s(\mathscr{A}')>22+9=3\cdot 12-5$, since we can create a pair without R_{first}. Finally, if $|\mathscr{A}'|=4$ then $s(\mathscr{A}')>22+21=4\cdot 12-5$.

Case $A=F$: We claim that it is sufficient to prove

$$s(\mathscr{A})+s(n)+s(n')>12|\mathscr{A}|+\begin{cases}8 & \text{if } F\in\mathscr{R} \text{ and } R_{\text{first}}\in\mathscr{A},\\ 9 & \text{if either } F\in\mathscr{R} \text{ or } R_{\text{first}}\in\mathscr{A},\\ 10 & \text{in all cases.}\end{cases} \tag{8}$$

First we show that (7) implies a contradiction.

If $F=E_1$ we note that $h=3$ (as $\mathscr{H}$ starts with 3 bins in $\mathscr{R}$). Thus Lemma 2.8, items (iii) and (iv), together with $w(\mathscr{H})\geq s(\mathscr{H})-12|\mathscr{H}|$ yields $w(\mathscr{H})\geq 3$ for $R_{\text{first}}\in\mathscr{A}$ or $w(\mathscr{H})\geq 2$ for $R_{\text{first}}\notin\mathscr{A}$. Summing this with $w(\mathscr{A})+w(A)>-3$ or $w(\mathscr{A})+w(A)>-2$, that are obtained from (6) and (8) in the respective cases, we obtain $w(\mathscr{L})>0$, a contradiction.

If $F\in\mathscr{R}$ then we know that F also contains old items of size at least 3 if $R_{\text{first}}\notin\mathscr{A}$ or even 4 if $R_{\text{first}}\in\mathscr{A}$ (and thus $F\neq R_{\text{first}}$). Summing this with the respective bound from (8) we obtain $s(\mathscr{A})+s(F)>12|\mathscr{A}|+12$. Summing this with $s(\mathscr{H})\geq 12|\mathscr{H}|$ from Lemma 2.8(iii) now yields $s(\mathscr{L})>12|\mathscr{L}|$, a contradiction.

Thus (7) always leads to a contradiction.

We now distinguish subcases depending on $|\mathscr{A}|$ and in each case we either prove (8) or obtain a contradiction directly. Note that $R_{\text{first}}\in\mathscr{A}$ whenever $|\mathscr{A}|\geq 4$.

Case $|\mathscr{A}|=2$: The two bins together with n and n' have size more than 36. Thus $s(\mathscr{A})+s(n)+s(n')>36=12\cdot 2+12$, which implies (8).

Case $|\mathscr{A}|=3$: We have $s(C)>9$ and the remaining two bins together with n and n' have size more than 36. Thus $s(\mathscr{A})+s(n)+s(n')>12|\mathscr{A}|+9$, which implies (8) in all cases except if $F=E_1$ and $R_{\text{first}}\notin\mathscr{A}$.

In the remaining case, $\mathscr{A}=\{L,M,T\}$ and $C=L$, as $\mathscr{A}$ contains no bin from $\mathscr{R}$ and $|\mathscr{A}|=3$. We prove a contradiction directly. Let o be the size of old items in T. We apply Lemma 2.8(ii), using the fact that $o+\rho>6$ by Lemma 2.5(v), where ρ is the total size of old items in $R_{\text{first}}\in\mathscr{H}$, and $h=3$. We get $o+s(\mathscr{H})\geq o+12|\mathscr{H}|+h+\rho-4>12|\mathscr{H}|+5$. Let n'' be a new item in T. Since n'' does not fit into M, $s(M)+s(n'')>18$; also $s(L)>9$ and $s(F)>18$. Summing all the bounds, we have $s(\mathscr{L}) \geq o + s(\mathscr{H}) + s(M) + s(n'') + s(L) + s(F) > 12|\mathscr{H}| + 5 + 18 + 9 + 18 = 12|\mathscr{L}| + 2$, a contradiction.

Case $|\mathscr{A}| = 4$: The last bin $R \in \mathscr{A}$ is in $\mathscr{R}$. Together with any previous bin it has size more than 21, the remaining two bins together with n and n' have size more than 36 by Lemma 2.10(ii). Thus $s(\mathscr{A}) + s(n) + s(n') > 21 + 36 = 4 \cdot 12 + 9$ which implies (8), since $R_{\text{first}} \in \mathscr{A}$.

Case $|\mathscr{A}| = 5$: First consider the case $F = E_1$. The last two bins of $\mathscr{A}$ are in $\mathscr{R}$, we pair them with two previous bins to form pairs of size more than The remaining bin has size at least 9, since n does not fit into it and $s(n) < 9$. We also have $s(F) > 18$. Thus $s(\mathscr{A}) + s(A) > 21 + 22 + 9 + 18 = 5 \cdot 12 + 10$, which implies (8).

If $F \in \mathscr{R}$ then one of the last two bins of $\mathscr{A}$ has size more than 11 and the other forms a pair of size more than 21 with one special bin. The remaining two bins together with n and n' have size more than 36 by Lemma 2.10(ii). Thus $s(\mathscr{A}) + s(n) + s(n') > 11 + 21 + 36 = 5 \cdot 12 + 8$ which implies (8), because $R_{\text{first}} \in \mathscr{A}$.

Case $|\mathscr{A}| = 6$: Then $\mathscr{A}$ contains all three special bins and three bins from $\mathscr{R}$, therefore also $F = E_1$. We form three pairs of a special bin with a bin from $\mathscr{R}$ of total size more than $21 + 22 + 22$. Since $s(F) > 18$, we have $s(\mathscr{A}) + s(F) > 21 + 22 + 22 + 18 = 6 \cdot 12 + 11$. Since in this case $A = F = E_1$, we have $s(\mathscr{H}) \geq 12|\mathscr{H}| + 3$ and $s(\mathscr{L}) > 12|\mathscr{L}|$, a contradiction.

In all of the cases we can derive a contradiction, which implies that our algorithm cannot fail. This concludes the proof of Theorem 2.1. □

3 Conclusions

We note that the analysis of our algorithm is tight, i.e., if we reduce the capacity of the bins below 18, the algorithm fails. Consider the following instance. Send two items of size 6 which are in the first phase packed separately into two bins. Then send $m-1$ items of size 12. One of them must be put into a bin with an item of size 6, i.e., one bin receives items of size 18, while all the items can be packed into m bins of size 12.

To decrease the upper bound below 1.5 seems challenging. In particular, the instance above and its modifications with more items of size 6 or slightly smaller items at the beginning shows that these items need to be packed in pairs. This in turns creates difficulties that, in the current approach, lead to new item and bin types; at this point we do not know if such an approach is feasible.

Another possible research direction is to improve the lower bounds, in particular for large m. It is quite surprising that there are no lower bounds for $m > 5$ larger than the easy bound of $4/3$.

Acknowledgment. The authors thank Emese Bittner for useful discussions during her visit to Charles University. We also thank to referees for many useful comments.

References

1. S. Albers and M. Hellwig. Semi-online scheduling revisited. *Theor. Comput. Sci.*, 443:1–9, 2012.

2. J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts. On-line load balancing with applications to machine scheduling and virtual circuit routing. *J. ACM*, 44:486–504, 1997.
3. Y. Azar and O. Regev. On-line bin-stretching. In *Proc. of Randomization and Approximation Techniques in Computer Science (RANDOM)*, pages 71–81. Springer, 1998.
4. Y. Azar and O. Regev. On-line bin-stretching. *Theor. Comput. Sci.* 268(1):17–41, 2001.
5. P. Berman, M. Charikar, and M. Karpinski. On-line load balancing for related machines. *J. Algorithms*, 35:108–121, 2000.
6. M. Böhm, J. Sgall, R. van Stee, and P. Veselý. Better algorithms for online bin stretching. In *Proc. of the 12th Workshop on Approximation and Online Algorithms (WAOA 2014)*, Lecture Notes in Comput. Sci. 8952, pages 23-34, Springer, 2015.
7. M. Böhm, J. Sgall, R. van Stee, and P. Veselý. Online Bin Stretching with Three Bins. arXiv preprint arXiv:1404.5569v3. 2016.
8. E. Coffman Jr., J. Csirik, G. Galambos, S. Martello, and D. Vigo. Bin Packing Approximation Algorithms: Survey and Classification, In P. M. Pardalos, D.-Z. Du, and R. L. Graham, editors, *Handbook of Combinatorial Optimization*, pages 455–531. Springer New York, 2013.
9. T. Ebenlendr, W. Jawor, and J. Sgall. Preemptive online scheduling: Optimal algorithms for all speeds. *Algorithmica*, 53:504–522, 2009.
10. M. Gabay, N. Brauner, V. Kotov. Computing lower bounds for semi-online optimization problems: Application to the bin stretching problem. HAL preprint hal-00921663, version 2, 2013.
11. M. Gabay, N. Brauner, V. Kotov. Improved Lower Bounds for the Online Bin Stretching Problem. HAL preprint hal-00921663, version 3, 2015.
12. M. Gabay, V. Kotov, N. Brauner. Semi-online bin stretching with bunch techniques. HAL preprint hal-00869858, 2013.
13. R. L. Graham. Bounds on multiprocessing timing anomalies. *SIAM J. Appl. Math.*, 17:263–269, 1969.
14. D. Johnson. *Near-optimal Bin Packing Algorithms*. Massachusetts Institute of Technology, project MAC. Massachusetts Institute of Technology, 1973.
15. H. Kellerer and V. Kotov. An efficient algorithm for bin stretching. *Operations Research Letters*, 41(4):343–346, 2013.
16. H. Kellerer, V. Kotov, M. G. Speranza, and Z. Tuza. Semi on-line algorithms for the partition problem. *Oper. Res. Lett.*, 21:235–242, 1997.
17. K. Pruhs, J. Sgall, and E. Torng. Online scheduling. In J. Y.-T. Leung, editor, *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, chapter 15, pages 15–1 – 15–41. CRC Press, 2004.
18. J. Ullman. The Performance of a Memory Allocation Algorithm. *Technical Report 100*, 1971.