Bounds for online bin packing with cardinality constraints*

József Békési †   György Dósa‡   Leah Epstein§

Abstract

We study a bin packing problem in which a bin can contain at most k items of total size at most 1, where $k \geq 2$ is a given parameter. Items are presented one by one in an online fashion. We analyze the best absolute competitive ratio of the problem and prove tight bounds of 2 for any $k \geq 4$. Additionally, we present bounds for relatively small values of k with respect to the asymptotic competitive ratio and the absolute competitive ratio. In particular, we provide tight bounds on the absolute competitive ratio of First Fit for $k = 2, 3, 4$, and improve the known lower bounds on asymptotic competitive ratios for multiple values of k. Our method for obtaining a lower bound on the asymptotic competitive ratio using a certain type of an input is general, and we also use it to obtain an alternative proof of the known lower bound on the asymptotic competitive ratio of standard online bin packing.

1 Introduction

We study a variant of bin packing called bin packing with cardinality constraints (BPCC). In this problem, the input consists of items, denoted by $1, 2, \ldots, n$, such that item i has a size $s_i > 0$ associated with it, and there is a global parameter $k \geq 2$, called the cardinality constraint. The goal is to partition the input items into subsets, called bins, such that the total size of items of every bin is at most 1, and the number of items packed into each bin does not exceed k. We believe that bounding the number of items as well as their total size provides a more accurate model for packing problems; for example, a data center can usually only store a constant number of files. BPCC is a well-studied variant in the offline and online environments [17, 18, 16, 5, 1, 8, 10, 11].

In this paper, we study online algorithms that receive and pack input items one by one, without any information on future input items. A fixed optimal offline algorithm that receives the complete list of items before packing it is denoted by OPT. For an input I and algorithm ALG, we let $ALG(I)$ denote the number of bins that A uses to pack I. We also use $OPT(I)$ to denote the

*Békési was partly supported by the Chinese–Hungarian bilateral project TÉT_12_CN-1-2012-0028, and by the European Union and the European Social Fund through project Telemedicine-focused research activities on the field of Mathematics, Informatics and Medical sciences, project no. TÁMOP-4.2.2.A-11/1/KONV-2012-0073. Dósa was partly supported by VKSZ_12-1-2013-0088 and Hungarian Scientific Research Fund NKFIH/OTKA grant SNN 116095.

†Department of Applied Informatics, Gyula Juhász Faculty of Education, University of Szeged, Hungary. bekesi@jgypk.u-szeged.hu

‡Department of Mathematics, University of Pannonia, Veszprem, Hungary, dosagy@almos.vein.hu.

§Department of Mathematics, University of Haifa, Haifa, Israel. lea@math.haifa.ac.il.

number of bins that OPT uses for a given input I. The absolute competitive ratio of an algorithm ALG is the supremum ratio over all inputs I between the number of bins $ALG(I)$ that it uses and the number of bins $OPT(I)$ that OPT uses. The asymptotic competitive ratio is the limit of absolute competitive ratios R_K when K tends to infinity and R_K takes into account only inputs for which OPT uses at least K bins. Note that (by definition), for a given algorithm (for some online bin packing problem), its asymptotic competitive ratio never exceeds its absolute competitive ratio. If the algorithm is offline, the standard terms are *approximation ratio* and *asymptotic approximation ratio*. For an algorithm whose competitive ratio (or approximation ratio) does not exceed R, we say that it is an R-competitive (or R-approximation). We see a bin as a set of items, and for a bin B, we let $s(B) = \sum_{i \in B} s_i$ be its level or load.

Bin packing problems are often studied with respect to the asymptotic measures. Approximation algorithms were designed for the offline version of BPCC (that is strongly NP-hard for $k \geq 3$) [17, 16, 5, 10], and the problem has an asymptotic fully polynomial approximation scheme (AFPTAS) [5, 10]. Using elementary bounds, it was shown by Krause, Shen, and Schwetman [17] that the cardinality constrained variant of First Fit (FF), that packs an item i into a minimum indexed bin where it fits both with respect to size and cardinality (i.e., the target bin must have at most $k-1$ items and its current level must be at most $1 - s_i$), has an asymptotic competitive ratio of at most $2.7 - \frac{2.4}{k}$. For $k \to \infty$, the asymptotic competitive ratio of FF is at most 2.7, which follows from the result of [17] and also from that of [12], since this is a special case of vector bin packing (with two dimensions).

Next, we survey other known results for BPCC. The case $k = 2$ is solvable using matching techniques in the offline scenario, but it is not completely resolved in the online scenario. Liang [20] showed a lower bound of $\frac{4}{3}$ on the asymptotic competitive ratio for this case, Babel et al. [1] improved the lower bound to $\sqrt{2} \approx 1.41421$, and designed an algorithm whose asymptotic competitive ratio is at most $1 + \frac{1}{\sqrt{5}} \approx 1.44721$ (improving over the previous bound, which was proved for FF). Recently, Fujiwara and Kobayashi [11] improved the lower bound to 1.42764. For larger k, there is a 2-competitive algorithm [1], and improved algorithms are known for $k = 3, 4, 5, 6$ (whose competitive ratios are at most 1.75, 1.86842, 1.93719, and 1.99306, respectively) [8].

Note that the upper bound of [17] for FF and $k = 3$ is 1.9, and an algorithm whose competitive ratio is at most 1.8 was proposed by [1]. A full analysis of the cardinality constrained variant of the Harmonic algorithm [19] is given in [8], and its competitive ratios for $k = 2$ and $k = 3$ are 1.5 and $\frac{11}{6}$, respectively (its competitive ratio is in $[2, 2.69103]$ for $k \geq 4$). As for lower bounds, until recently, except for the case $k = 3$ for which a lower bound of 1.5 on the competitive ratio was proved in [1], most of the known lower bounds followed from the analysis of lower bounds for standard bin packing [28, 26, 3].

New lower bounds for many values of k were given by Fujiwara and Kobayashi in [11], and in particular, they proved lower bounds of 1.5 and $\frac{25}{17} \approx 1.47058$ for $k = 4$ and $k = 5$, respectively. For $6 \leq k \leq 9$, the current best lower bound remained 1.5, which was implied by the lower bound of Yao [28], and for $k = 10$ and $k = 11$, lower bounds of $\frac{80}{53} \approx 1.50943$ and $\frac{44}{29} \approx 1.51724$, respectively, were proved in [11] (see [11] for the lower bounds of other values of k). In this paper we provide improved lower bounds on the asymptotic competitive ratio of arbitrary online algorithms for

value of k	previous asymptotic LB	new LB	
5	1.47058 [11]	3/2	= 1.5
7	1.5 [28]	217/143	≈ 1.51748
8	1.5 [28]	32/21	≈ 1.52380
9	1.5 [28]	189/124	≈ 1.52419
10	1.50943 [11]	235/154	≈ 1.52597
11	1.51724 [11]	209/137	≈ 1.52554
14	1.52595 [11]	315/206	≈ 1.52912
15	1.52912 [11]	75/49	≈ 1.53061
16	1.52567 [11]	72/47	≈ 1.53191
17	1.52312 [11]	765/499	≈ 1.53306
18	1.52459 [11]	135/88	≈ 1.53409
19	1.52678 [11]	30799/20072	≈ 1.53442
20	1.52912 [11]	2365/1541	≈ 1.53471
21	1.52941 [11]	13251/8633	≈ 1.53492
22	1.52914 [11]	10417/6786	≈ 1.53507
23	1.53004 [11]	49795/32434	≈ 1.53527
24	1.53086 [11]	152/99	≈ 1.53535
25	1.53162 [11]	54175/32284	≈ 1.53539
26	1.53231 [11]	3523/2294	≈ 1.53574
27	1.53296 [11]	2439/1588	≈ 1.53589
28	1.53356 [11]	1897/1235	≈ 1.53603
29	1.53412 [11]	70789/46079	≈ 1.53625
30	1.53465 [11]	6105/3974	≈ 1.53623
31	1.53514 [11]	84103/54742	≈ 1.53635
32	1.53560 [11]	39104/25449	≈ 1.53656
33	1.53603 [11]	23925/15568	≈ 1.53680
34	1.53644 [11]	289/188	≈ 1.53723
35	1.53682 [11]	76195/49569	≈ 1.53715

Table 1: New lower bounds on the asymptotic competitive ratio. The second column contains the previously known bounds and the third column contains our improved lower bounds.

$k = 5, 7, 8, 9, 10, 11$. The values of these lower bounds are 1.5 for $k = 5$, and approximately 1.51748, 1.5238, 1.5242, 1.526, 1.5255, for $k = 7, 8, 9, 10, 11$, respectively. We also provide improved lower bounds for larger values of k, see Table 1.

The methods used in this work differ from the ones used before. In the past, the inputs that were used for $k \geq 3$ consisted of sub-inputs with identical items, such that the number of items of each kind is equal (and the input may be stopped after a sub-input was presented). We use this framework too, but the numbers of items will not necessarily be equal. Moreover, unlike the constructions given in the previous work, in some cases we do not adapt previously used sequences (those that were used for proving lower bounds for standard online bin packing [26, 3]), but we use a different input sequence. To avoid dealing with packing patterns (subsets of items that can fit into bins), we use weight functions. For that, we provide a general theorem that allows to deal with inputs coming from a class of inputs, and is useful for proving lower bounds on the asymptotic competitive ratio for various bin packing problems (we demonstrate it also on standard online bin packing, getting the lower bound of [3]).

There are few known results for the absolute measures. The asymptotic $(1+\varepsilon)-$approximation algorithm of Caprara, Kellerer, and Pferschy [5] uses $(1+\varepsilon)OPT(I)+1$ bins to pack the items of an input I, and thus, choosing $\varepsilon > 0$ to be small (for example $\varepsilon = \frac{1}{100}$) results in a polynomial time absolute $\frac{3}{2}$-approximation algorithm. This is the best possible unless P=NP. In the online environment, it is not difficult to see that given the absolute upper bound of 1.7 on the competitive ratio of FF for standard bin packing [7], the upper bound of $2.7-2.4/k$ becomes an absolute one (we provide the proof here for completeness). In this paper, we also analyze the absolute competitive ratio, and show a tight bound of 2 on the absolute competitive ratio for any $k \geq 4$, and a tight bound of 1.5 for $k = 2$. The upper bound for $k = 4$ is proved for FF. An upper bound for $k = 5$ is proved using an algorithm that performs FF except for one case. We show that a variant of the algorithm of [1] has an absolute competitive ratio 2 for any $k \geq 3$. In the case $k = 3$, we provide a lower bound of $\frac{7}{4} = 1.75$ on the absolute competitive ratio of any algorithm, and show that the absolute competitive ratio of FF is $\frac{11}{6} \approx 1.8333$. A complete analysis of the *asymptotic* competitive ratio of FF can be found in [6].

For standard bin packing [25, 13, 14, 15, 19, 28, 21], it is known that the asymptotic competitive ratio is in [1.5403,1.58889] [3, 22], and the absolute competitive ratio is $\frac{5}{3}$ [29, 2] (the absolute competitive ratio of FF without cardinality constraints is 1.7 [7]). Another related problem is called *class constrained bin packing* [9, 23, 24, 27]. In that problem each item has a color, and a bin cannot contain items of more than k colors (for a fixed parameter k). BPCC is the special case of that problem where all items have distinct colors (with the same value of k). A lower bound of 2 on the asymptotic competitive ratio was proved for the former problem [24] even for the case where items have equal sizes, whereas the latter problem (BPCC) has an algorithm whose asymptotic competitive ratio is 2 [1] (and BPCC is trivial for the case of equal item sizes). Moreover, for BPCC and $k = 2$, there is an algorithm whose asymptotic competitive ratio is approximately 1.44721 [1], whereas the more general problem has a lower bound of approximately 1.5652 on the asymptotic competitive ratio [9]. Thus, it is possible that class constrained bin packing is more difficult than BPCC.

We start with lower bounds in Section 2, where both the absolute competitive ratio and the asymptotic competitive ratio are studied. We consider algorithms afterwards, in Section 3, where we analyze FF for small k, and design algorithms whose absolute competitive ratios are at most 2.

2 Lower bounds

In this section we present lower bounds for the two measures. As the two measures are quite different, the lower bound constructions are different as well.

2.1 Lower bounds on the absolute competitive ratio

We show that the absolute competitive ratio is at least 2 for $k \geq 4$. Together with the analysis of Section 3.1, we will find that this is the best possible competitive ratio.

Proposition 1 *The absolute competitive ratio of any deterministic online algorithm for $k \geq 4$ is at least 2.*

Proof. Let $0 < \varepsilon < \frac{1}{10k}$. The input starts with k items, each of size ε, called tiny items. Since an optimal solution packs them into one bin, if an online algorithm uses two bins, then we are done. Otherwise the algorithm packs them into one bin, and no further items can be combined into this bin, since it already has k items. The next two items have sizes of $\frac{1}{3} + \varepsilon$. If the algorithm packs them into two new bins, then the next item has size $\frac{2}{3}$ and it requires a new bin. An optimal solution packs the last item with $k-1$ tiny items, and the remaining three items into another bin, while the algorithm uses four bins. Otherwise, the algorithm packs the two items of sizes $\frac{1}{3}+\varepsilon$ into one bin. In this case the last two items have sizes of $\frac{1}{2} + \varepsilon$. The algorithm now has four bins, while an optimal solution has two bins, each with an item of size $\frac{1}{2} + \varepsilon$, an item of size $\frac{1}{3} + \varepsilon$, and $\lfloor \frac{k}{2} \rfloor$ or $\lceil \frac{k}{2} \rceil$ items of size ε each (which is possible since at most $k-2$ items of size ε are added; for $k = 4$, $\frac{k}{2} = k - 2 = 2$ holds, and $\lceil \frac{k}{2} \rceil \leq \frac{k+1}{2} \leq k - 2$ holds for $k \geq 5$). ■

In the case $k = 2$, a lower bound of $\frac{3}{2}$ on the absolute competitive ratio follows from an input that consists of two tiny items, each of size ε, possibly followed by two larger items, each of size $1 - \varepsilon$ (for $0 < \varepsilon < \frac{1}{2}$). If an algorithm packs the first two items into separate bins, then its absolute competitive ratio is 2, as these items can be packed into one bin. Otherwise, the larger two items require two new bins, while an optimal solution for the four items consists of two bins, each containing a tiny item and a larger item.

Next, we present a lower bound of $\frac{7}{4}$ on the absolute competitive ratio of any algorithm for $k = 3$. Recall that the best asymptotic competitive ratio for $k = 3$ is in $[\frac{3}{2}, \frac{7}{4}]$. The upper bound of [17] for the asymptotic competitive ratio of FF is $2.7 - 2.4/3 = 1.9$, and we will show a tight bound of $\frac{11}{6} \approx 1.83333$ on the absolute competitive ratio of FF.

Proposition 2 *The absolute competitive ratio of any deterministic online algorithm for $k = 3$ is at least $\frac{7}{4} = 1.75$. The absolute competitive ratio of FF algorithm for $k = 3$ is at least $\frac{11}{6} \approx 1.8333$.*

Proof. Let $0 < \varepsilon < \frac{1}{24}$. The input starts with three tiny jobs of size ε each. Since an optimal solution can pack them into one bin, to avoid a competitive ratio of at least 2, the algorithm must do the same. Note that the bin containing these items cannot receive any additional items. Next, two items of sizes $\frac{1}{3} + \varepsilon$ arrive. If the last two items are packed into separate bins, an item of size $\frac{2}{3}$ is presented. The last three items are packed into three bins, and the algorithm uses four bins for this input. An optimal solution can pack the items into two bins; the last item is combined with two tiny items, and the remaining three items are packed into a second bin, which is possible given the value of ε.

Otherwise, the algorithm has two bins, where the second one can still receive one item of size at most $\frac{1}{3} - 2\varepsilon$. The remaining items will be larger, and thus they will be packed into new bins.

Now, two items of sizes $\frac{1}{3}+3\varepsilon$ arrive. If the algorithm uses two new bins to pack them, then two items whose sizes are equal to $\frac{2}{3}-2\varepsilon$ arrive, and the algorithm is forced to use two new bins for them, for a total of six bins. An optimal solution uses three bins; two bins contain (each) an item of size $\frac{2}{3}-2\varepsilon$, an item of size $\frac{1}{3}+\varepsilon$, and a tiny item. The remaining three items have total size $\frac{2}{3}+7\varepsilon$ and can be packed into a third bin by an optimal solution. Thus, the competitive ratio is 2 in this case.

Otherwise, the algorithm has three bins, where the second and third bins can still receive one item each, but no item of size at least $\frac{1}{3}$ can be packed there. The remaining items will be larger than $\frac{1}{3}$, and thus they will be packed into new bins. Specifically, there are four items whose sizes are equal to $\frac{2}{3}-4\varepsilon > \frac{1}{2}$. Each such item must be packed into a separate new bin, for a total of seven bins. An optimal solution can combine each such item with an item of size $\frac{1}{3}+\varepsilon$ or an item of size $\frac{1}{3}+3\varepsilon$, and three such bins also receive a tiny item.

Consider the following input for FF (see [6] for an example showing an asymptotic lower bound of the same value). There are six tiny items of size ε, six medium items of size $\frac{1}{3}+\varepsilon$, and six huge items of size $\frac{1}{2}+\varepsilon$. An optimal solution packs six bins containing one item of each size. FF packs two bins with three tiny items each, three bins with two medium items each, and six bins, each containing one huge item. ■

2.2 Lower bounds on the asymptotic competitive ratio

In this section we present a new method for proving lower bounds on the asymptotic competitive ratio of bin packing problems. This method can be applied for different variants, and we use it for standard online bin packing (getting the best known lower bound [3] in a simple way), and for bin packing with cardinality constraints, where it allows us to improve the lower bounds known for relatively small values of k. The main advantage of this method is that it does not require the explicit construction of all possible patterns or dominant patterns (patterns are multi-sets of input items that can be packed into bins). The approach of [3] is to use weights as well, but their usage is slightly different. Our approach resembles that of proving upper bounds on the asymptotic competitive ratio of algorithms [25, 15, 22].

Consider an input of the following form for a given bin packing problem Π. Let $\theta \geq 2$ be a fixed positive integer. There are θ lists of items, where the list L_i (for $1 \leq i \leq \theta$) has identical items, each of size s_i, where $s_1 < s_2 < \cdots < s_\theta$. For a large integer $N > 0$, the list L_i has $\alpha_i \cdot N$ items, where $0 < \alpha_i \leq 1$ is a rational parameter for $i = 1, \ldots, \theta$ (N will be selected such that $\alpha_i \cdot N$ is an integer). The items are presented to an online algorithm sorted by non-decreasing sizes (that is, items are presented sorted by non-decreasing indices of their lists) and the input may be stopped after all the items of some list were presented. That is, there are θ possible inputs, and we will examine the behavior of the algorithm for all possible inputs.

Given a set of values $w_i > 0$ for $i = 1, \ldots, \theta$, w_i is called the *weight* of an item of list L_i, and the weight of a bin is equal to the total weight of its items. For $i = 1, \ldots, \theta+1$, let W_i denote the maximum weight of any bin that contains only items of lists $i, i+1, \ldots, \theta$, and no items of earlier lists (so $W_{\theta+1} = 0$), assuming that an unlimited number of items of any size is available. Obviously, the algorithm has no such bins if the input stops after list L_j, for some $j < i$. By definition, $W_i \geq W_{i+1}$ for $i = 1, \ldots, \theta$. For $i = 0, \ldots, \theta$, let OPT_i denote the cost of an optimal solution for the input that consists of the items of lists $L_1, \ldots, L_i$ (thus $OPT_0 = 0$), and denote this input by $\mathcal{L}_i$. We have $OPT_i = \Theta(N)$, as $OPT_i \geq \alpha_i \cdot N \cdot s_i$ (because this is the total size of items of list i) and $OPT_i \leq i \cdot N$ (as this is an upper bound on the number of input items in $\mathcal{L}_i$). Let O_i be an upper bound on $\frac{OPT_i}{N}$ for $1 \leq i \leq \theta$ and let $O_0 = 0$ (these values are constants depending on the input parameters).

Theorem 3 *The asymptotic competitive ratio of any online (deterministic or randomized) algorithm the bin packing problem* Π *is at least*

$$\frac{\sum_{i=1}^{\theta} \alpha_i \cdot w_i}{\sum_{i=1}^{\theta} (O_i - O_{i-1}) \cdot W_i} \ .$$

Proof. Consider a deterministic or randomized algorithm A for Π. Algorithm A will receive one of the inputs $\mathcal{L}_i$ (for some $i \in \{1, 2, \ldots, \theta\}$). Let X_i be the number of new bins (or expected number of bins) that A opens (uses for the first time) while packing the items of list L_i. Assume that the competitive ratio is R. Let f be a monotonically non-decreasing function where $f(n) = o(n)$ such that for any input I, $A(I) \leq R \cdot OPT(I) + f(OPT(I))$. We have $A(\mathcal{L}_i) \leq R \cdot OPT_i + f(OPT_i)$. Note that $A(\mathcal{L}_i) = \sum_{j=1}^{i} X_j$. For any set of θ parameters $\gamma_i \geq 0$ for $i = 1, \ldots, \theta$ (constants that are independent of N), we multiply the constraint of index i and take the sum of the resulting inequalities to find $\sum_{i=1}^{\theta} \gamma_i A(\mathcal{L}_i) \leq \sum_{i=1}^{\theta} \gamma_i (R \cdot OPT_i + f(OPT_i))$. Letting $\beta_i = \sum_{j=i}^{\theta} \gamma_i$ we find $\sum_{i=1}^{\theta} \gamma_i A(\mathcal{L}_i) = \sum_{i=1}^{\theta} \gamma_i \sum_{j=1}^{i} X_j = \sum_{j=1}^{\theta} X_j \sum_{i=j}^{\theta} \gamma_i = \sum_{j=1}^{\theta} \beta_j \cdot X_j$. Thus, $\sum_{j=1}^{\theta} \beta_j \cdot X_j \leq R \sum_{i=1}^{\theta} \gamma_i \cdot OPT_i + \sum_{i=1}^{\theta} \gamma_i f(OPT_i)$.

As the weight of an item of list L_i is w_i, the total weight of all items, denoted by W, satisfies $W = \sum_{i=1}^{\theta} w_i \cdot \alpha_i \cdot N$. By definition, any bin that is opened starting the time when the first item of list L_i arrived has total weight no larger than W_i. The total weight of all items is equal to the total weight of bins used by the algorithm, which is no larger than $\sum_{i=1}^{\theta} W_i \cdot X_i$, as X_i bins are opened by the algorithm when items of L_i arrive (these bins may be used also for packing items of later lists). Letting $\gamma_i = W_i - W_{i+1}$ for $i = 1, \ldots, \theta$, we get $\beta_i = \sum_{j=i}^{\theta} \gamma_i = W_i$. We find $N \sum_{i=1}^{\theta} w_i \cdot \alpha_i = W \leq \sum_{i=1}^{\theta} \beta_i \cdot X_i$. This gives $N \sum_{i=1}^{\theta} w_i \cdot \alpha_i \leq R \sum_{i=1}^{\theta} \gamma_i \cdot OPT_i + \sum_{i=1}^{\theta} \gamma_i f(OPT_i)$. Since $OPT_i = \Theta(N)$, we divide by N and neglect the lower order term to get

$$\sum_{i=1}^{\theta} w_i \cdot \alpha_i \leq R \sum_{i=1}^{\theta} \gamma_i \cdot \frac{OPT_i}{N} \leq R \sum_{i=1}^{\theta} (W_i - W_{i+1}) \cdot O_i = R \sum_{i=1}^{\theta} W_i \cdot (O_i - O_{i-1}) \ .$$

This proves the theorem. ∎

We start with proving the current best known lower bound for standard bin packing using Theorem 3. Given $\theta \geq 4$ and $0 < \delta < \frac{1}{7^{4\theta}}$, define sizes as follows. For $i = 2, \ldots, \theta - 2$, $s_i = \frac{1+\delta}{7^{\theta-i-1}}$, $s_{\theta-1} = \frac{1+\delta}{3}$, $s_\theta = \frac{1+\delta}{2}$, and $s_1 = \frac{1}{6 \cdot 7^{\theta-3}} - \delta$. We also let $\alpha_i = 1$, i.e., there are N items in each list. The value δ was chosen such that $s_1 > \frac{1}{6 \cdot 7^{\theta-3}+1}$. The total size of i items of different lists $L_1, \ldots, L_i$, for $1 \leq i \leq \theta - 2$ is at most $\frac{1}{6 \cdot 7^{\theta-i-2}} - \frac{5\delta}{6}$, as $\sum_{j=2}^{i} \frac{1}{7^{\theta-j-1}} = \frac{7^{i-1}-1}{6 \cdot 7^{\theta-3}}$, and $s_1 + \sum_{j=2}^{i} s_i \leq \frac{1}{6 \cdot 7^{\theta-3}} - \delta + (1+\delta)\frac{7^{i-1}-1}{6 \cdot 7^{\theta-3}} \leq \frac{1}{6 \cdot 7^{\theta-i-2}} + \delta \frac{7^{\theta-3}-1-6 \cdot 7^{\theta-3}}{6 \cdot 7^{\theta-3}} < \frac{1}{6 \cdot 7^{\theta-i-2}} - \frac{5\delta}{6}$, as $i \leq \theta - 2$. The total size of $\theta - 1$ items of different lists $L_1, \ldots, L_{\theta-1}$ is at most $\frac{1-\delta}{2}$ as $\sum_{j=1}^{\theta-1} s_j = \sum_{j=1}^{\theta-2} s_j + \frac{1+\delta}{3} < \frac{1}{6} - \frac{5\delta}{6} + \frac{1}{3} + \frac{\delta}{3} = \frac{1}{2} - \frac{\delta}{2}$, and by the last calculation, the total size of θ items of different lists $L_1, \ldots, L_\theta$ is at most 1. Thus, we let $O_1 = \frac{1}{6 \cdot 7^{\theta-3}}$, $O_i = \frac{1}{6 \cdot 7^{\theta-i-2}}$ for $2 \le i \le \theta - 2$, $O_{\theta-1} = \frac{1}{2}$, and $O_\theta = 1$.

We also let $w_i = 6 \cdot 7^{i-2}$ for $i = 2, \ldots, \theta - 2$, $w_1 = 1$, and $w_{\theta-1} = w_\theta = 12 \cdot 7^{\theta-4}$. The weights of items are approximately proportional to their sizes, with the exception of the largest two sizes whose weights are equal. We have defined $w_{\theta-1} = 2w_{\theta-2}$, as (by $s_{\theta-1} = \frac{1+\delta}{3}$ and $s_{\theta-2} = \frac{1+\delta}{7}$), $2s_{\theta-2} < s_{\theta-1} < 3s_{\theta-2}$.

Claim 4 *We have $W_i = (7^{\theta-i-1} - 1) \cdot 6 \cdot 7^{i-2} = 6 \cdot 7^{\theta-3} - 6 \cdot 7^{i-2}$ for $i = 2, \ldots, \theta - 2$, $W_1 = 6 \cdot 7^{\theta-3}$, $W_{\theta-1} = 24 \cdot 7^{\theta-4}$, and $W_\theta = 12 \cdot 7^{\theta-4}$.*

Proof. Note that for $i = 2, \ldots, \theta-2$, $(7^{\theta-i-1} - 1)s_i = \frac{(1+\delta)(7^{\theta-i-1}-1)}{7^{\theta-i-1}} = (1+\delta)(1 - \frac{1}{7^{\theta-i-1}}) \le 1 + \delta - \frac{1}{7^{\theta-i-1}} \le 1 + \frac{1}{7^{4\theta}} - \frac{1}{7^\theta} < 1$. Therefore, for these values of i, $W_i \ge (7^{\theta-i-1} - 1)w_i = 7^{\theta-i-1} - 1) \cdot 6 \cdot 7^{i-2}$. For $i = 1$, $6 \cdot 7^{\theta-3} s_1 < 1$, and therefore $W_1 \ge 6 \cdot 7^{\theta-3}$.

Since $s_\theta > \frac{1}{2}$ and $s_{\theta-1} > \frac{1}{3}$, at most one item of size s_θ can be packed into a bin, and at most two items of size at least $s_{\theta-1}$, thus $W_\theta = w_\theta$ and $W_{\theta-1} = w_{\theta-1} + w_\theta$ (the equality holds since these combinations of items can actually be packed).

Consider a bin B that contains items of lists $L_i, \ldots, L_\theta$ for some $1 \le i \le \theta - 2$. To calculate an upper bound on the total weight, we will apply a process in which we will repeatedly replace some items with other items such that the total size does not increase, keeping the bin valid, and such that the total weight does not decrease. The process has two steps, and eventually all the items of B will be replaced with items of size s_i. Since during the process of replacement the total weight does not decrease, we will be able to calculate the maximum total weight of such items that are packed into one bin by the weight of a maximum number of items of size s_i that can be packed into a bin.

If B contains an item of size $s_{\theta-1}$ or an item of size s_θ, replace every such item with two items of size $s_{\theta-2} = \frac{1+\delta}{7}$. Since $s_\theta > s_{\theta-1} = \frac{1+\delta}{3} > 2 \cdot \frac{1+\delta}{7}$, the bin remains valid. As $w_\theta = w_{\theta-1} = 2w_{\theta-2}$, the total weight does not change. If $i \ge 2$, for any item of size s_j for $j > i$, $s_j = 7^{j-i}s_i$ and $w_j = 7^{j-i}w_i$, thus replacing an item of size s_j with 7^{j-i} items of size s_i does not change the total size of items, and it does not change their total weight. If $i = 1$, $s_j > 6 \cdot 7^{j-2} s_1$ while $w_j = 6 \cdot 7^{j-2} w_1$, and an item of size s_j can be replaced with $6 \cdot 7^{j-2}$ items of size s_1.

Now, we can thus assume that B only has items of size s_i, and moreover, due to size constraints, it can contain at most $\lfloor \frac{1}{s_i} \rfloor$ items of size s_i. For $i = 1$, this value is at most $6 \cdot 7^{\theta-3}$. For $2 \le i \le \theta - 2$, this value is at most $7^{\theta-i-1} - 1$. Thus, for $i = 1$, the total weight that can be packed into a bin is no larger than $6 \cdot 7^{\theta-3}$, and for $2 \le i \le \theta - 2$ it is at most $\lfloor \frac{1}{s_i} \rfloor \cdot w_i \le (7^{\theta-i-1} - 1) \cdot 6 \cdot 7^{i-2} = 6 \cdot 7^{\theta-3} - 6 \cdot 7^{i-2}$. Since these numbers of items can actually be packed, this implies the equality in the values of W_i. ■

Corollary 5 *The parameters chosen here result in a lower bound of $\frac{248}{161}$ on the asymptotic competitive ratio for any algorithm for standard online bin packing.*

Proof. We have

$$\sum_{i=1}^{\theta} \alpha_i \cdot w_i = \sum_{i=1}^{\theta} w_i = 1 + 6\sum_{i=2}^{\theta-2} 7^{i-2} + 24 \cdot 7^{\theta-4} = 31 \cdot 7^{\theta-4} \ .$$

We use $W_i - W_{i+1} = 12 \cdot 7^{\theta-4}$ for $i = \theta-2, \theta-1, \theta$, $W_i - W_{i+1} = 36 \cdot 7^{i-2}$ for $2 \leq i \leq \theta-3$, and $W_1 - W_2 = 6$. Recall that $O_i = \frac{1}{6 \cdot 7^{\theta-i-2}}$, $O_{\theta-1} = \frac{1}{2}$, and $O_{\theta=1}$.

We use $\sum_{i=1}^{\theta}(O_i - O_{i-1}) \cdot W_i = \sum_{i=1}^{\theta}(W_i - W_{i+1}) \cdot O_i$ and get,

$$\sum_{i=1}^{\theta}(W_i - W_{i+1}) \cdot O_i = \frac{1}{7^{\theta-3}} + \sum_{i=2}^{\theta-3} 6 \cdot 7^{2i-\theta} + (\frac{1}{6} + \frac{1}{2} + 1)12 \cdot 7^{\theta-4} = \frac{1}{7^{\theta-3}} + \frac{49^{\theta-4}-1}{8 \cdot 7^{\theta-4}} + 20 \cdot 7^{\theta-4} \ .$$

By Theorem 3, we find a lower bound of

$$\frac{31 \cdot 7^{\theta-4}}{\frac{1}{7^{\theta-3}} + \frac{49^{\theta-4}-1}{8 \cdot 7^{\theta-4}} + 20 \cdot 7^{\theta-4}} = \frac{31}{\frac{1}{7^{2\theta-7}} + \frac{1}{8} + \frac{1}{8 \cdot 7^{2\theta-8}} + 20} \ .$$

Letting θ tend to infinity, we get a lower bound of $\frac{31}{20.125} = \frac{248}{161} \approx 1.54037$. ■

Our method of proving new lower bounds for BPCC is to use inputs where the numbers of items in the lists are not necessarily equal. Consider the case $k=5$. Let $\theta = 4$, $\alpha_1 = \frac{1}{2}$, $\alpha_i = 1$ for $i = 2, 3, 4$. Let $0 < \delta < \frac{1}{2000}$, $s_1 = \frac{1}{42} - \delta > 0$, $s_2 = \frac{1+\delta}{7}$, $s_3 = \frac{1+\delta}{3}$, and $s_4 = \frac{1+\delta}{2}$. In this case, there are less items of size s_1 (than the numbers of other items), since a bin that contains two items of size s_2 and two items of size s_3 can only contain one additional item. We use $O_1 = \frac{1}{10}$, $O_2 = \frac{3}{10}$ (as any five items of sizes at most s_2 can be packed into a bin), $O_3 = \frac{1}{2}$ (as a set of five items, consisting of two items of size s_3, two items of size s_2, and one item of size s_1 can be packed into a bin), and $O_4 = 1$ (as a set of items consisting of one item of each size can be packed into a bin, while only half of the bins will contain an item of size s_1).

Consider the cases $k = 7, 8, \ldots, 11$. Let $\theta = 4$, the item sizes are the same as the case $k = 5$, $\alpha_1 = \frac{k-6}{6}$ (where $\alpha_1 < 1$), $\alpha_i = 1$ for $i = 2, 3, 4$. The motivation for the value α_1 is that a bin that has six items of size s_2 can contain only $k-6$ additional items. We use $O_1 = \frac{k-6}{6k}$, $O_2 = \frac{1}{6}$ (as a set of six items of size s_2 and $k-6$ items of size s_1 can be packed into a bin), $O_3 = \frac{1}{2}$ (as a set consisting of at most two items of each one of the three sizes s_1, s_2, s_3 can be packed into a bin), and $O_4 = 1$. Let $w_2 = 1$, $w_3 = w_4 = 2$ for all cases. For $k = 5$, let $w_1 = 2$, for $k = 7, 8$, let $w_1 = 1$, for $k = 9$, let $w_1 = \frac{1}{2}$, and for $k = 10, 11$, let $w_1 = \frac{1}{3}$.

Lemma 6 *1. For all cases $W_2 \leq 6$, $W_3 \leq 4$, and $W_4 \leq 2$.*

2. For $k = 5$, $W_1 \leq 10$, for $k = 7, 8$, $W_1 \leq k + 2$, for $k = 9$, $W_1 \leq 8$, and for $k = 10, 11$, $W_1 \leq \frac{k}{3} + 4$.

Proof. We start with the first part. The claim regarding W_4 holds since $W_4 = w_4$ must hold (as every bin opened for the last list contains exactly one item). In the cases where there may be items of L_3 packed into a bin, any item of size s_4 can be replaced with an item of size s_3 without increasing the total size or number of items, and without changing the total weight (as $w_3 = w_4$

in all cases). Thus, we assume that no such items are packed into the considered bins. Any bin with items of L_3 can contain at most two items and therefore $W_3 \le 2w_3$. Consider a bin with no items of list L_1. Let y_2 be the number of items of size s_2, and let $y_3 \le 2$ be the number of items of size s_3. Their weight is $y_2 + 2y_3$, and their total size is above $\frac{y_2+2y_3}{7}$ (and no larger than 1). Thus, $W_2 \le y_2 + 2y_3 \le 6$.

Next, we bound W_1 for all values of k considered here. For $k = 5$, since the weight of any item is at most 2, and there are at most five items packed into each bin, $W_1 \le 10$. Otherwise, consider a packed bin, and let y_2 and $y_3 \le 2$ have the same meaning as before, while y_1 is the number of items of size s_1. We have that $y_1 + y_2 + y_3 \le k$ must hold, and by the value of W_2, $y_2 + 2y_3 \le 6$. The total size of items is at least $y_1(\frac{1}{42} - \delta) + y_2(\frac{1+\delta}{7}) + y_3(\frac{1+\delta}{3}) > \frac{y_1+6y_2+14y_3}{42} - y_1\delta$. Since $y_1\delta < \frac{1}{42}$, we have $y_1 + 6y_2 + 14y_3 \le 42$. For $k = 7, 8$, the weight of the bin is $y_1 + y_2 + 2y_3$. Since $y_1 + y_2 + y_3 \le k$ and $y_3 \le 2$, $W_1 \le k + 2$.

For $k = 9$, the weight is $y_1/2 + y_2 + 2y_3$. Adding $y_1 + 6y_2 + 14y_3 \le 42$ to $4(y_1 + y_2 + y_3) \le 36$, we get $5y_1 + 10y_2 + 18y_3 \le 78$, or alternatively, $y_1/2 + y_2 + 2y_3 \le 7.8 + 0.2y_3$. Thus, if $y_3 \le 1$, $W_1 \le 8$. If $y_3 = 2$, then by substituting it into the inequalities we get $y_1 + 6y_2 \le 14$ and $y_1 + y_2 \le 7$, or alternatively, $y_1 + 2y_2 \le 14 - 4y_2$ and $y_1 + 2y_2 \le 7 + y_2$. If $y_2 \ge 2$, then the first inequality implies $y_1/2 + y_2 \le 3$, and if $y_2 \le 1$, then the second inequality implies $y_1/2 + y_2 \le 4$. In both cases, $y_1/2 + y_2 + 2y_3 \le 8$.

For $k = 10, 11$, the weight is $y_1/3 + y_2 + 2y_3$, and we will show $y_1 + 3y_2 + 6y_3 \le k + 12$. As $y_3 \le 2$, we consider three cases. If $y_3 = 0$, then we find $y_2 \le 6$ and $y_1 + y_2 \le k$. Thus, $y_1 + 3y_2 \le k + 12$. If $y_3 = 1$, then we find $y_1 + 6y_2 \le 28$, $y_2 \le 4$, and $y_1 + y_2 \le k - 1$. If $y_2 \le 3$, we get $y_1 + 3y_2 + 6y_3 \le (k-1) + 2 \cdot 3 + 6 < k + 12$. If $y_2 = 4$, then by the first inequality we find $y_1 \le 4$, and $y_1 + 3y_2 + 6y_3 \le 4 + 3 \cdot 4 + 6 = 22 \le k + 12$ as $k \ge 10$. Finally, if $y_3 = 2$, we get $y_1 + 6y_2 \le 14$, $y_2 \le 2$, and $y_1 + y_2 \le k - 2$. If $y_2 \le 1$, we get $y_1 + 3y_2 + 6y_3 \le (k-2) + 2 \cdot 1 + 6 \cdot 2 = k + 12$. If $y_2 = 2$, then by the first inequality we find $y_1 \le 2$, and $y_1 + 3y_2 + 6y_3 \le 2 + 3 \cdot 2 + 6 \cdot 2 = 20 \le k + 12$. ■

We apply Theorem 3 to get the following.

Corollary 7 *The following values are lower bounds on the competitive ratios.*

- $\frac{3}{2} = 1.5$ *for* $k = 5$.
- $\frac{k^2+24k}{k^2+10k+24}$ *for* $k = 7, 8$. *This value is equal to* $217/143 \approx 1.5174825$ *for* $k = 7$ *and to* $\frac{32}{21} \approx 1.5238095$ *for* $k = 8$.
- $\frac{10.5}{62/9} = \frac{189}{124} \approx 1.5241935$, *for* $k = 9$.
- $\frac{k^2+84k}{k^2+48k+36}$ *for* $k = 10, 11$. *This value is equal to* $235/154 \approx 1.525974$ *for* $k = 10$ *and to* $\frac{209}{137} \approx 1.525547$ *for* $k = 11$.

Proof. For $k = 5$, $\sum_{i=1}^{4} \alpha_i \cdot w_i = 6$ and $\sum_{i=1}^{4}(O_i - O_{i-1}) \cdot W_i = 4$.

For $k = 7, 8$, $\sum_{i=1}^{4} \alpha_i \cdot w_i = \frac{k+24}{6}$ and $\sum_{i=1}^{4}(O_i - O_{i-1}) \cdot W_i = \frac{7}{3} + \frac{6}{k} + \frac{k^2-4k-12}{6k}$.

For $k = 9$, $\sum_{i=1}^{4} \alpha_i \cdot w_i = 5.25$ and $\sum_{i=1}^{4}(O_i - O_{i-1}) \cdot W_i = \frac{31}{9}$.

For $k = 10, 11$, $\sum_{i=1}^{4} \alpha_i \cdot w_i = \frac{k+84}{18}$ and $\sum_{i=1}^{4}(O_i - O_{i-1}) \cdot W_i = \frac{7}{3} + \frac{6}{k} + \frac{k^2+6k-72}{18k}$. ■

Note that in the cases $k = 6$ and $k = 12$, our methods do not produce improved lower bounds, and they give exactly the known lower bound.

Next, consider the cases $14 \leq k \leq 18$. Let $\theta = 4$, $\alpha_i = 1$ for $i = 1, 2, 3, 4$. Let $0 < \delta < \frac{1}{2000}$, $s_1 = \frac{1}{18} - 3\delta > 0$, $s_2 = \frac{1+\delta}{9}$, $s_3 = \frac{1+\delta}{3}$, and $s_4 = \frac{1+\delta}{2}$. We use $O_1 = \frac{1}{k}$, $O_2 = \frac{1}{6}$ (as a set of six items of size s_1 and six items of size s_2 can be packed into a bin), $O_3 = \frac{1}{2}$ (as a set of six items, consisting of two items of each size out of s_1, s_2, s_3 can be packed into a bin), and $O_4 = 1$ (as a set of items consisting of one item of each size can be packed into a bin). Let $w_1 = 1$, $w_2 = 2$, and $w_3 = w_4 = 6$.

Lemma 8 *We have $W_4 \leq 6$, $W_3 \leq 12$, $W_2 \leq 16$, and $W_1 \leq 18$.*

Proof. As in the proof of Lemma 6, $W_4 = w_4$ and $W_3 = w_3 + w_4$. We will prove upper bounds on W_1 and W_2 such that the number of packed items is not necessarily bounded by k. This may only increase the bounds.

To prove the bound for W_2, consider a bin with items of sizes above $\frac{1}{9}$. Replace any item of size s_4 or s_3 with three items of size s_2. The total size of items cannot increase while the total weight is unchanged. The bin now contains at most 8 items of size s_2, and therefore its weight is at most 16.

To prove the bound for W_1, consider a bin B. Replace any item of size s_4 or s_3 with six items of size s_1, and any item of size s_2 is replaced with two items of size s_1. The total size of items cannot increase while the total weight is unchanged. The bin now contains at most 18 items of size s_1, and therefore its weight is at most 18. ■

We apply Theorem 3 to get the following.

Corollary 9 *The value $\frac{45k}{29k+6}$ is a lower bound on the asymptotic competitive ratio for k, where $14 \leq k \leq 18$.*

Proof. We have $\sum_{i=1}^{4} \alpha_i \cdot w_i = 15$ and $\sum_{i=1}^{4} (O_i - O_{i-1}) \cdot W_i = 18/k + 16(1/6 - 1/k) + 12/3 + 6/2$. ■

Note that in the cases $k = 12, 13$, our method does not produce improved lower bounds.

Finally, consider the cases $k = 19, 20, \ldots, 35$. Let $\theta = 5$, $\alpha_1 = \frac{k-18}{18}$, $\alpha_i = 1$ for $i = 2, 3, 4, 5$. Let $0 < \delta < \frac{1}{10000}$, $s_1 = \frac{1}{342} - \delta$, $s_2 = \frac{1+\delta}{19}$, $s_3 = \frac{1+\delta}{9}$, $s_4 = \frac{1+\delta}{3}$, and $s_5 = \frac{1+\delta}{2}$. In this case, there are less items of size s_1, since a bin that contains 18 items of size s_2 can only contain $k - 18$ additional items. We use $O_1 = \frac{k-18}{18k}$ (any bin will contain k items), $O_2 = \frac{1}{18}$ (any bin will contain 18 items of size s_2 and $k - 18$ items of size s_1), $O_3 = \frac{1}{6}$ (any bin will contain six items of size s_3, six items of size s_2, and at most six items of size s_1), $O_4 = \frac{1}{2}$ (any bin will contain two items for each one of the sizes s_2, s_3, s_4, and at most two items of size s_1), and $O_5 = 1$ (any bin will contain one item of each of the sizes of s_2, s_3, s_4, s_5, and at most one item of size s_1). Let $w_2 = 1$, $w_3 = 2$, and $w_4 = w_5 = 6$. The value of w_1 will differ for the different values of k, and we denote it by ρ_k, where $0 < \rho_k < 1$.

Lemma 10 *We have $W_2 \leq 18$, $W_3 \leq 16$, $W_4 \leq 12$, and $W_5 \leq 6$.*

Proof. As in previous cases, $W_5 = w_5$ and $W_4 = 2 \cdot w_4$. We will prove upper bounds on W_2 and W_3 for bins where the number of packed items is not necessarily bounded by k. This may only increase the bounds. Given a bin with items of sizes in $\{s_3, s_4, s_5\}$, replace each item of size s_4 or s_5 with three items of size s_3. As a result, the total size does not increase, and the total weight is unchanged. Since at most eight items of size s_3 can be packed into a bin, $W_3 \le 16$. Given a bin with items of sizes in $\{s_2, s_3, s_4, s_5\}$, replace each item of size s_4 or s_5 with six items of size s_2, and each item of size s_3 is replaced with two items of size s_2. As a result, the total size does not increase, and the total weight is unchanged. Since at most 18 items of size s_3 can be packed into a bin, $W_3 \le 18$. ■

Consider a bin that possibly contains items of all lists, where the total weight of items of lists L_2, L_3, L_4, L_5 is exactly 18. Let λ_k denote the maximum number of items of size s_1 that the bin can contain under this condition. Similarly, consider a bin where the total weight of items of lists L_2, L_3, L_4, L_5 is exactly 17. Let ψ_k denote the maximum number of items of size s_1 that the bin can contain under this condition. Let $\rho_k = \frac{1}{\psi_k - \lambda_k}$ (which is well defined, as we will show that $\psi_k > \lambda_k$ for all k). Recall that for any k, the value w_1 is defined by ρ_k. We define an additional parameter, $\phi_k = 18 + \rho_k \cdot \lambda_k$ for all values of k considered here. These values are displayed in Table 2.

Lemma 11 *Let $k \in \{19, 20, \ldots, 35\}$. The values λ_k and ψ_k are as in Table 2, and $W_1 \le \phi_k$.*

Proof. We start with proving that the values of λ_k and ψ_k given in Table 2 correspond to our definition of these values. For a given bin, let y_2, y_3, and y_4, be the numbers of items of sizes s_2, s_3, and s_4 packed into the bin (we replace items of size s_5, if such items exist, with items of size s_4, as they are smaller and have the same weight).

If the total weight of the items of size at least s_2 is 18, then $y_2 + 2y_3 + 6y_4 = 18$, and the total size of items is $(1+\delta)(\frac{y_2}{19} + \frac{y_3}{9} + \frac{y_4}{3}) = \frac{1+\delta}{18}(y_2 + 2y_3 + 6y_4) - \frac{1+\delta}{342}y_2 = 1 + \delta - \frac{1+\delta}{342}y_2$. The bin can still contain items of size s_1 of total size no larger than $\frac{1+\delta}{342}y_2 - \delta$. Let y_1 be the number of such items. We will show $y_1 = \min\{k - y_2 - y_3 - y_4, y_2\}$ by proving that the total size of $y_2 + 1$ items of size s_1 exceeds $\frac{1+\delta}{342}y_2 - \delta$, while the total size of y_2 such items does not exceed this value (and obviously the bin cannot contain more than $k - y_2 - y_3 - y_4$ additional items).

We find that $y_1 \le y_2$, as the total size of $y_2 + 1$ items of size s_1 is $(y_2 + 1)(\frac{1}{342} - \delta)$, and $(y_2 + 1)(\frac{1}{342} - \delta) > \frac{1+\delta}{342}y_2 - \delta$ is equivalent to $343\delta \cdot y_2 < 1$, which holds as $y_2 \le 18$ and $\delta < \frac{1}{10000}$. On the other hand, $y_2(\frac{1}{342} - \delta) \le \frac{1+\delta}{342}y_2 - \delta$ is equivalent to $\frac{343y_2}{342} \ge 1$ (which holds for $y_2 \ge 1$), and therefore if $y_2 \ge 1$, then y_2 items of size s_1 can be packed into the bin (in terms of total size).

For any valid triple (y_2, y_3, y_4) of numbers of items, the maximum value of y_1 is therefore $\min\{k - y_2 - y_3 - y_4, y_2\}$ items. To find the possible triples (y_2, y_3, y_4), we take into account that $y_4 \le 2$ (as no bin can contain more than two items of sizes above $\frac{1}{3}$) and $y_3 + 3y_4 \le 8$ (as the total weight of these items is at most $W_3 \le 16$, and it is equal to $w_3 \cdot y_3 + w_4 \cdot y_4 = 2y_3 + 6y_4$). These patterns are: $(2,2,2)$, $(4,1,2)$, $(6,0,2)$, $(2,5,1)$, $(4,4,1)$, $(6,3,1)$, $(8,2,1)$, $(10,1,1)$, $(12,0,1)$, $(2,8,0)$, $(4,7,0)$, $(6,6,0)$, $(8,5,0)$, $(10,4,0)$, $(12,3,0)$, $(14,2,0)$, $(16,1,0)$, and $(18,0,0)$. Thus,

$$\lambda_k = \max\{\min\{2, k-6\}, \min\{4, k-7\}, \min\{6, k-8\}, \min\{8, k-11\},$$

$$\min\{10, k-12\}, \min\{12, k-13\}, \min\{14, k-16\}, \min\{16, k-17\}, \min\{18, k-18\}\} \ .$$

The last bound was computed by considering each pattern separately, and computing $\min\{k - y_2 - y_3 - y_4, y_2\}$, then removing any entry that is dominated by another entry, for example, $\min\{2, k-8\}$

that results from the fourth triple is dominated by $\min\{2, k-6\}$ resulting from the first triple. The values in the table are with accordance to this calculation.

If the total weight of the items of sizes above $\frac{1}{19}$ is 17, then $y_2 + 2y_3 + 6y_4 = 17$, and the total size of items is $(1+\delta)(\frac{y_2}{19} + \frac{y_3}{9} + \frac{y_4}{3}) = \frac{1+\delta}{18}(y_2 + 2y_3 + 6y_4) - \frac{1+\delta}{342}y_2 = \frac{17(1+\delta)}{18} - \frac{1+\delta}{342}y_2$. The bin can contain items of size s_1 of total size no larger than $\frac{1+\delta}{18} + \frac{1+\delta}{342}y_2 - \delta$. Let y_1 be this number. In this case we show that $y_1 = \min\{y_2+19, k-y_2-y_3-y_4\}$. To show $y_1 \le y_2+19$, we prove that the total size of y_2+20 items of size s_1 exceeds $\frac{1+\delta}{18} + \frac{1+\delta}{342}y_2 - \delta$. Indeed, $(y_2+20)(\frac{1}{342} - \delta) > \frac{1+\delta}{342}y_2 - \delta + \frac{1+\delta}{18}$ is equivalent to $\frac{1}{342} > \frac{343y_2\delta}{342} + \frac{343\delta}{18}$, which holds as $y_2 \le 18$ and $\delta < \frac{1}{10000}$. On the other hand, it is possible to pack y_2+19 items of size s_1 (in terms of total size) as the empty space is $\frac{1+\delta}{342}y_2 - \delta + \frac{1+\delta}{18}$, we saw that y_2 items can be packed into a space of $\frac{1+\delta}{342}y_2 - \delta$, and since $s_1 < \frac{1}{342}$, 19 items of size s_1 can be packed into a space of $\frac{1+\delta}{18}$. Thus, for any triple (y_2, y_3, y_4), it is possible to pack $\min\{k - y_2 - y_3 - y_4, y_2 + 19\}$ items. The possible triples are: $(1,2,2)$, $(3,1,2)$, $(5,0,2)$, $(1,5,1)$, $(3,4,1)$, $(5,3,1)$, $(7,2,1)$, $(9,1,1)$, $(11,0,1)$, $(1,8,0)$, $(3,7,0)$, $(5,6,0)$, $(7,5,0)$, $(9,4,0)$, $(11,3,0)$, $(13,2,0)$, $(15,1,0)$, and $(17,0,0)$. Thus,

$$\psi_k = \max\{\min\{20, k-5\}, \min\{22, k-6\}, \min\{24, k-7\}, \min\{26, k-10\}, \min\{28, k-11\},$$
$$\min\{30, k-12\}, \min\{32, k-15\}, \min\{34, k-16\}, \min\{36, k-17\}\} .$$

The calculation is similar to the one for λ_k, and the values for ψ_k in the table are deduced from this calculation.

Consider a bin B, and let X denote the total weight of items that do not belong to list L_1 that are packed into B. We have $X \le 18$, as $W_2 \le 18$.

If $X = 18$, then the total weight of items is at most $X + \rho_k \cdot y_1 \le 18 + \rho_k \cdot \lambda_k = \phi_k$. If $X = 17$, then the total weight of items is at most $17 + \rho_k \cdot \psi_k = 17 + \rho_k \cdot (\psi_k - \lambda_k) + \rho_k \cdot \lambda_k \le 18 + \rho_k \cdot \lambda_k = \phi_k$, by the definitions of ρ_k and ϕ_k.

We claim that otherwise (if $X \le 16$), the total weight of items is no larger than $X + \rho_k \cdot k$. For $k \le 32$, $(k - \lambda_k)\rho_k \le 2$, and therefore $X + k \cdot \rho_k \le 16 + 2 + \lambda_k\rho_k = \phi_k$. If $k \ge 33$, and $X \le 15$, since $(k - \lambda_k)\rho_k \le 3$, we also find $X + k \cdot \rho_k \le 15 + 3 + \lambda_k\rho_k = \phi_k$. In the case $X = 16$, there must be at least four items of lists L_2, L_3, L_4, L_5 packed into the bin, as the total weight of three items is at most 14 (there can be at most two items of weight 6 as their sizes exceed $\frac{1}{3}$). Thus, there are at most $k - 4$ items whose weights are ρ_k. We have that $(k - 4 - \lambda_k)\rho_k < 2$ for $k \in \{33, 34, 35\}$, and $X + (k-4) \cdot \rho_k \le 16 + 2 + \lambda_k\rho_k = \psi_k$ in this case as well. ■

Corollary 12 *The values stated in Table 1 are lower bounds on the competitive ratios for $k = 19, 20, \ldots, 35$.*

Proof. Recall that $\alpha_1 = \frac{k-18}{18}$, and $\alpha_i = 1$ for $i = 2, 3, 4, 5$. Thus, $\sum_{i=1}^{\theta} \alpha_i \cdot w_i = \frac{k-18}{18} \cdot \rho_k + w_2 + w_3 + w_4 + w_5 = \frac{k-18}{18} \cdot \rho_k + 15$.

We have $O_1 - O_0 = \frac{k-18}{18k}$, $O_2 - O_1 = \frac{1}{k}$, $O_3 - O_2 = \frac{1}{9}$, $O_4 - O_3 = \frac{1}{3}$, and $O_5 - O_4 = \frac{1}{2}$. Thus, $\sum_{i=1}^{\theta}(O_i - O_{i-1})W_i = \frac{k-18}{18k} \cdot (18 + \rho_k \cdot \lambda_k) + \frac{18}{k} + \frac{16}{9} + \frac{12}{3} + \frac{6}{2} = 1 - \frac{18}{k} + \frac{\rho_k \cdot \lambda_k}{18} - \frac{\rho_k \cdot \lambda_k}{k} + \frac{18}{k} + \frac{79}{9} = \frac{88}{9} + \rho_k \cdot \lambda_k(\frac{1}{18} - \frac{1}{k})$.

Therefore, using Theorem 3 we find a lower bound of $\frac{15+(k-18)\rho_k/18}{88/9+\rho_k\lambda_k(1/18-1/k)}$ on the asymptotic competitive ratio for $k = 19, 20, \ldots, 35$. ■

value of k	λ_k	ψ_k	ρ_k	$(k-\lambda_k)\rho_k$	$\phi_k - 18 = \rho_k\lambda_k$
19	8	14	1/6	11/6	4/3
20	8	15	1/7	12/7	8/7
21	9	16	1/7	12/7	9/7
22	10	17	1/7	12/7	10/7
23	10	18	1/8	13/8	5/4
24	11	19	1/8	13/8	11/8
25	12	20	1/8	13/8	3/2
26	12	20	1/8	14/8	3/2
27	12	21	1/9	15/9	4/3
28	12	22	1/10	8/5	6/5
29	13	22	1/9	16/9	13/9
30	14	23	1/9	16/9	14/9
31	14	24	1/10	17/10	7/5
32	15	24	1/9	17/9	5/3
33	16	24	1/8	17/8	2
34	16	24	1/8	18/8	2
35	17	25	1/8	18/8	17/8

Table 2: Auxiliary variables for the analysis of lower bounds for $k = 19, 20, \ldots, 35$.

The construction that was used for $k = 19, \ldots, 35$ can be used for $k = 36$, but the resulting lower bound is lower than the known lower bound [11]. It is possible, however, to prove improved bounds for larger values of k. Consider, for example, the cases $k = 43, 44, 45$. Let $s_1 = \frac{1}{1806} - \delta$, $s_2 = \frac{1+\delta}{43}$, $s_3 = \frac{1+\delta}{7}$, $s_4 = \frac{1+\delta}{3}$, and $s_5 = \frac{1+\delta}{2}$; $\alpha_1 = \frac{k-42}{42}$, and $\alpha_i = 1$ for $i = 2, 3, 4, 5$. It can be verified that using the weights $w_1 = \rho_k$, $w_2 = 1$, $w_3 = 6$, and $w_4 = w_5 = 12$, where $\rho_{43} = \frac{1}{14}$, $\rho_{44} = \frac{1}{15}$, and $\rho_{45} = \frac{1}{16}$, gives $W_1 = 42 + \rho_k\lambda_k$, $W_2 = 42$, $W_3 = 36$, $W_4 = 24$, and $W_5 = 12$. This gives lower bound of approximately 1.53903, 1.53906, and 1.53909 on the asymptotic competitive ratios for $k = 43, 44, 45$, respectively. This slightly improves the previously known lower bound of approximately 1.53900 [26] mentioned in [11].

3 Algorithms

In this section we define and analyze algorithms for BPCC, and analyze FF as well.

3.1 A 2-competitive algorithm for all $k \geq 3$

In the case $k = 2$, an algorithm that packs each item into a different bin is 2-competitive in the absolute sense, as if it uses x bins, no packing can use less than $\lceil \frac{x}{2} \rceil$ bins. We later show that FF has a smaller absolute competitive ratio, and focus on the case $k \geq 3$ in this section. Kotov et al. [1] designed an algorithm that is 2-competitive in the asymptotic sense. We present a simplified version of that algorithm and prove that it is 2-competitive in the absolute sense. Our algorithm Thin and Fat (TF) has three kinds of bins.

1. **Paired bins.** Those are bins partitioned into pairs such that the total size of items for each pair is strictly above 1, and the total number of items packed into the two bins is at least k. Those bins will not be used for packing further items.

2. **Fat bins.** Those are bins (which are not paired) containing exactly $k-1$ items.

3. **Thin bins.** Those are non-empty bins (which are not paired) containing at most $k-2$ items.

After we define TF, we will prove that if it has at least one fat bin, then it has at most one thin bin. The algorithm acts as follows. Initially all three sets of bins are empty. Let $i \geq 1$ be a new item. The following steps are processed for i until it is packed.

1. If there is a fat bin B such that $s(B) + s_i > 1$, pack i into a new bin, match B and the new bin, these bins become paired.

2. If there are no thin bins, pack i into a new bin.

3. If there exists a thin bin B such that $s(B) + s_i \leq 1$, then pack i into B. If B becomes fat and there is a thin bin $B' \neq B$, match B and B', these bins become paired.

4. If there are no fat bins, pack i into a new bin.

5. Pack i into a fat bin B, match B with a thin bin B', these bins become paired.

Lemma 13 *1. In all cases, the actions described above can be performed, and all items are packed.*

2. For any two thin bins, the total size of items packed into them is above 1*.*

3. Every two bins that are matched have a sufficient total size of items (a total size above 1*) and a sufficient number of items (at least k items).*

4. If there is at least one fat bin, then there is at most one thin bin.

Proof. We start with proving part 1, i.e., we show that any item i can be packed into the bin that it is assigned to. In steps 1,2, and 4, the item is packed into a new bin. Step 3 is applied provided that B exists. Such a bin has at most $k-2$ items, and has sufficient space. Assume that step 5 is reached. Since i is not packed in step 1, every fat bin can receive i since it has $k-1$ items and sufficient space. Since step 4 was not applied, a fat bin must exist. Since in step 5 the packing is unconditional, all items will be packed. The only other action that is performed unconditionally is matching bins in step 5. The thin bin must exist as i was not packed in step 2.

Next we consider part 2. Note that a thin bin can become fat, but a fat bin cannot become thin. Thus, a new thin bin is created only by packing an item into a new bin. The bin remains thin as long as it is not paired, and it has at most $k-2$ items. Obviously, the total size packed into a bin cannot decrease over time. Thus, to prove this part, it is sufficient to prove that a pair consisting of a thin bin and another thin bin that was just created (and was not paired immediately) have a total size of items above 1. A new bin B' that it not paired immediately can be created in steps 2 and 4. In step 2 it becomes the only thin bin. In step 4, it is created since the item was not packed in step 3, thus for any existing thin bin B, $s(B)+s(B')=s(B)+s_i>1$.

Consider part 3. Bins are matched only in steps 1,3,5. In step 1, the two bins will have k items and a total size of items above 1. In step 3, the pair is created only if a thin bin B becomes fat, i.e., B now has $k-1$ items and B' has at least one item. Moreover, since B and B' were thin, their total size of items is above 1. In step 5, since step 3 was not applied, we have $s(B')+s_i>1$, and the total number of items is at least $k+1$.

To verify the last property (part 4), consider the cases where i is packed into a bin that is not paired immediately. In step 2, there will be a unique thin bin. In step 3, if B becomes fat and it is not paired, then no thin bins remain. Otherwise, there is no change in the numbers of fat and thin bins. In step 4, there will be no fat bins after i is packed, as a bin with a single item is thin. ■

Theorem 14 *For any $k\geq 3$ and for any input L, $TF(L)\leq 2\cdot OPT(L)$.*

Proof. Let p, f, and t be the numbers of paired bins, fat bins, and thin bins when the algorithm terminates. Let S be the total size of items, and n their number.

Assume first that $f=0$. In this case $TF(L)=p+t$. The total size of items of every two paired bins is above 1, and moreover, the total size of items of every pair of thin bins is above 1. If $p=0$ and $t\leq 1$, the solution is optimal (containing no bins if the input is empty and one bin otherwise). If $p=0$ and $t\geq 2$, $OPT(L)\geq S>\frac{t}{2}$, while $TF(L)=t$. We are left with the case $p>0$ (and therefore $p\geq 2$). If $t\geq 2$, $OPT(L)\geq S>\frac{p}{2}+\frac{t}{2}$, while $TF(L)=p+t$. Otherwise, $t\leq 1$ and $TF(L)\leq p+1$ hold, while $OPT(L)\geq S>\frac{p}{2}$, and by integrality of $OPT(L)$, $OPT(L)\geq\frac{p+1}{2}$. Thus, in both cases, $p+f+t=p+t\leq 2OPT(L)$.

Otherwise, $f\geq 1$. In this case $t\leq 1$. We have $n\geq\frac{p}{2}\cdot k+(k-1)f+t$ (as every two paired bins have at least k items, a fat bin has $k-1$ items, and a thin bin has at least one item). If $t=0$, then we get $n\geq\frac{k}{2}(p+f)=\frac{k}{2}\cdot TF(L)$, since $k-1>\frac{k}{2}$. Otherwise, $t=1$. We have $n\geq\frac{p}{2}\cdot k+(k-1)f+1=\frac{k}{2}(p+f+1)+(\frac{k}{2}-1)f-\frac{k}{2}+1=\frac{k}{2}(p+f+1)+(\frac{k}{2}-1)(f-1)\geq\frac{k}{2}(p+f+1)=\frac{k}{2}\cdot TF(L)$. Since $OPT(L)\geq\frac{n}{k}$, we get $TF(L)\leq 2OPT(L)$. ■

3.2 Analysis of the absolute competitive ratio of FF for several cases

In our analysis, any bin created by an algorithm (FF or another algorithm) that has j items for $j\leq k$ is called a j-bin, and a bin whose number of items is in $[j,k-1]$ for some $1\leq j<k$ is called a j^+-bin.

In this section analyze FF. We focus on cases of small k, for which FF has a small absolute competitive ratio. In particular, for $k=2$ it has the best possible absolute competitive ratio, and for $k=3$, its absolute competitive ratio is strictly below 2.

We will use the next two claims.

Claim 15 *Every bin of OPT has at most one item of a 1-bin of FF.*

Proof. Assume by contradiction that this is not the case, and items i, j, such that $i < j$, of one bin of OPT are packed into 1-bins by FF. When j arrives, since $s_i + s_j \leq 1$, FF does not open a new bin for j, as there is at least one existing bin where it can be packed, a contradiction. ■

Claim 16 *Let $1 \leq j \leq k-1$. Every j-bin except for at most one bin has level above $\frac{j}{j+1}$. Moreover, every j^+-bin except for at most one bin has level above $\frac{j}{j+1}$.*

Proof. Assume that there exists a j-bin (or j^+-bin) whose level is at most $\frac{j}{j+1}$. All further j^+-bins (that appear later in the ordering of FF) only have items of sizes above $\frac{1}{j+1}$, and each such bin has at least j items, so their levels are above $\frac{j}{j+1}$. ■

We start with the simple case $k = 2$. A simple upper bound of $\frac{3}{2}$ is achieved by a greedy matching algorithm, which is a generalization of FF. It is folklore that this algorithm matches at least half of the edges that an optimal solution can match and therefore it translates into a $\frac{3}{2}$-competitive algorithm for bin packing (where an edge between two items exists if they can be packed together into a bin). Moreover, for this case the upper bound $2.7 - 2.4/k$ is equal to 1.5. For completeness, and as an introductory case for analysis using weights, we show how FF can be analyzed using weights for the case $k = 2$.

The usage of weights is slightly different from their usage for proving lower bounds. We usually use a weight function w, that is applied on sizes of items. Thus, we define $w(a)$ for $a \in (0, 1]$, where the variable a denotes the size of an item. For a set of items A and a set of bins $\mathcal{A}$, let $w(A)$ and $w(\mathcal{A})$ denote the total weight of all items of A or $\mathcal{A}$. Furthermore, let $W = w(I)$ be the total weight of all items of the input I. In this kind of analysis, the weights of bins of the algorithm and of OPT are compared, using the property that for a fixed input, the total weight of items is equal for all algorithms. The weights will not necessarily be based on sizes, but they may be based on the packing of an algorithm or of OPT, and in particular, for $k = 2$, w will be a function of the items rather than of their sizes. For $k = 2$, an item of an i-bin of FF is assigned a weight of $\frac{1}{i}$ (for $i = 1, 2$). Obviously, any bin of FF has weight 1, and we analyze the total weight of bins of OPT. A bin of OPT cannot have two items of 1-bins, and therefore its weight cannot exceed $\frac{3}{2}$. We find that for any input L, the total weight satisfies $FF(L) = W \leq 1.5 \cdot OPT(L)$.

We continue with a simple proof that the upper bound $2.7 - 2.4/k$ on the competitive ratio of FF holds in the absolute sense for any $k \geq 2$, that is based on the proof of [17].

Proposition 17 *The absolute competitive ratio of FF is at most $2.7 - 2.4/k$.*

Proof. Let L be a non-empty input, and partition it into two subsequences, L_1 that consists of all items that are packed into bins eventually having k items, and $L_2 = L \setminus L_1$. By the definition of FF, running it on L_1 results in the same bins for these items as in the run on L, and the same is true for L_2, even if FF is applied without taking the cardinality constraint into account. Let $M_1 = FF(L_1)$ and $M_2 = FF(L_2)$ be the resulting numbers of bins, where $FF(L) = M_1 + M_2$. We will use $OPT(L) \geq \frac{|L|}{k}$ and $|L| - |L_2| = |L_1| = kM_1$. Since the output for L_2 is valid without cardinality constraints, we have $FF(L_2) \leq 1.7 OPT(L_2)$.

First, consider the case $M_2 \leq OPT(L)$. Since every bin of FF has at least one item, we have $|L_2| \geq M_2$, and therefore $M_1 + M_2 = M_1 + M_2/k + (1-1/k)M_2 \leq \frac{|L_1|}{k} + \frac{|L_2|}{k} + (1-1/k)M_2 = \frac{|L|}{k} + (1-1/k)M_2 \leq (2-1/k)OPT(L) \leq (2.7-2.4/k)OPT(L)$, where the last inequality holds for any $k \geq 2$.

Otherwise, as the number of 1-bins is at most $OPT(L)$, and the remaining bins for L_2 have at least two items, thus, $|L_2| \geq M_2 + (M_2 - OPT(L))$, and we get $M_1 + M_2 \leq M_1 + 2M_2/k + (k-2)M_2/k \leq \frac{|L_1|}{k} + |L_2|/k + OPT(L)/k + (k-2)M_2/k = |L|/k + OPT(L)/k + (k-2)/k \cdot 1.7OPT(L_2) \leq OPT(L) + OPT(L)/k + (1.7k - 3.4)OPT(L)/k = (2.7-2.4/k)OPT(L)$. ■

3.2.1 The case $k = 3$

In this section we show that the absolute competitive ratio of FF for $k = 3$ is exactly $\frac{11}{6} < 2$ (note that the asymptotic competitive ratio of FF is also equal to this value).

Theorem 18 *The absolute competitive ratio of FF for $k = 3$ is exactly $\frac{11}{6}$.*

Proof. The lower bound follows from Proposition 2. Next, we prove the upper bound. Let I be an input sequence of items. Let $OPT = OPT(I)$.

Restricting our attention to the 2-bins and 1-bins created by FF, we can see that these bins would have been created by running FF only on the subsequence of the items packed into them, even if the cardinality constraint is not taken into account. Thus, as in [4], it can be assumed that there does not exist a 2-bin generated by FF that contains two items that are packed together in an optimal solution. The last assumption is valid as merging two items that are packed into the same 2-bin of FF into one item, where the resulting merged item arrives instead of the first item of the two, would result in the same packing with the exception that the 2-bin of FF becomes a 1-bin of FF (the property that the output of FF is almost unchanged is valid both for the execution of FF on the original input and for the execution of FF on the items of 2-bins and 1-bins). Thus, by merging every such pair of items into one item, we get an input of the required form (and we assume that I already satisfies this property).

Moreover, if the number of 1-bins is $OPT(I)$, then no bin of the optimal solution contains two items that are packed into 2-bins of FF (as in [4]). For completeness, we prove the last property. Assume by contradiction that a bin B^* of OPT (for the input I) has two items, denoted by i_2 and i_3, of 2-bins of FF. These items are packed into distinct bins of FF, as we assume that there are no two items of one 2-bin of FF are packed into the same bin of OPT. Let their 2-bins packed by FF be B and B', such that B appears before B' in the ordering of FF. Without loss of generality, we assume that FF packed i_2 into B and it packed i_3 into B'. Let i_1 be the other item of B, and let $\tilde{B^*}$ be its bin in the packing of OPT (where $\tilde{B^*} \neq B^*$). Since no bin of OPT has more than one item of a 1-bin of FF, while the number of 1-bins is $OPT(I)$, every bin of OPT has such an item that FF packs in a 1-bin. Let i_4 be such an item of B^*, and let i_5 be such an item of $\tilde{B^*}$.

We find $s_{i_3} + s_{i_1} + s_{i_2} > 1$ and $s_{i_4} + s_{i_5} > 1$, as i_3 was not packed into B, and i_4, i_5 are packed into 1-bins (the item out of i_4 and i_5 that arrives later was not packed with the other item out of these two items). On the other hand, $s_{i_3} + s_{i_2} + s_{i_4} \leq 1$ and $s_{i_1} + s_{i_5} \leq 1$. We have $2 < s_{i_1} + s_{i_2} + s_{i_3} + s_{i_4} + s_{i_5} \leq 2$, a contradiction.

We split the analysis into cases.

Case 1. The number of 1-bins is $OPT(I)$. Consider the items of these bins. An optimal solution has at most one such item packed into each bin, and thus every bin of OPT contains such an item. Additionally it can contain at most one item packed into a 2-bin by FF. We define a weight function based on the packing of FF. An item packed into an i-bin has weight $\frac{1}{i}$. We find that any bin of the optimal solution has weight of at most $1 + \frac{1}{2} + \frac{1}{3} = \frac{11}{6}$.

Case 2. The number of 1-bins is at most $OPT(I) - 1$. In this case we define weights that are based on item sizes. An item of size above $\frac{1}{2}$ (called a big item) has weight $\frac{2}{3}$, an item of size in $(\frac{1}{4}, \frac{1}{2}]$ (called a medium item) has weight $\frac{1}{2}$, and an item of size in $(0, \frac{1}{4}]$ (called a small item) has weight $\frac{1}{3}$. Note that there is at most one item packed into a 1-bin whose size does not exceed $\frac{1}{2}$ (and moreover, there is at most one bin whose level does not exceed $\frac{1}{2}$). If such an item exists, then we call it the special item. The special item can be small or medium.

Let W denote the total weight. A bin of OPT has weight at most $\frac{3}{2}$ if it has no big item. If it has a big item, it can have at most one medium item, and therefore its weight is at most $\frac{2}{3} + \frac{1}{2} + \frac{1}{3} = \frac{3}{2}$. Thus, $W \leq \frac{3}{2} \cdot OPT(I)$.

Given the output of FF, let X_1, X_2, and X_3 denote the numbers of 1-bins, 2-bins, and 3^+-bins, respectively (so $FF(I) = X_1 + X_2 + X_3$). Recall that in this case we assume $X_1 \leq OPT(I) - 1$.

Claim 19 *If there is a special item and it is small, then any* 2*-bin of FF has a level above* $\frac{3}{4}$. *There is at most of* 2*-bin of FF that has at least one small item.*

Proof. Assume by contradiction that there exist two bins of FF, none of which is a 3-bin, where each of them has a level no larger than $\frac{3}{4}$ at termination, and each of them has at least one small item. Let such two bins be denoted by B_1 and B_2, where B_2 has a larger index in the ordering of FF. Then, when a small item of B_2 arrives, FF can pack it into B_1, contradicting the action of FF.

This proves that at most one 2-bin can have both a level of at most $\frac{3}{4}$ and at least one small item. If there is a special item and it is small, its bin satisfies the condition that its level is at most $\frac{3}{4}$ and it has a small item, and therefore any 2-bin has either a level above $\frac{3}{4}$ or no small items (or both). ■

Claim 20 *We have* $W \geq \frac{2}{3}X_1 + X_2 + X_3 - \frac{1}{3}$.

Proof. We analyze the weights of bins of FF. Any item has a weight of at least $\frac{1}{3}$, thus any 3^+-bin has a total weight no smaller than 1.

Every 1-bin has a weight of $\frac{2}{3}$, except for that of the special item, if it exists. If a 2-bin has a big item (and its second item is medium or small), or it has two medium items, its total weight is at least 1. Otherwise, it has at least one small item and its second item is not big, so its level is no larger than $\frac{3}{4}$. By the previous claim, there is at most one such 2-bin. If such a 2-bin exists, its total weight is at least $\frac{2}{3}$ (since the weight of any item is at least $\frac{1}{3}$), and if its level is above $\frac{1}{2}$, it cannot have two small items, and its total weight is at least $\frac{5}{6}$.

If there is no special item, we find that the total weight of 1-bins and 2-bin is at least $\frac{2}{3}X_1 + X_2 - \frac{1}{3}$, and at most one 2-bin has a weight of at least $\frac{2}{3}$, and every other 1-bin or 2-bin has a weight of at least 1. Similarly, if there is a special item, and every 2-bin has a weight of at least 1, we are done since the weight of the special item is at least $\frac{1}{3}$. Assume that there exists a 2-bin of weight below 1 and there exists a special item. The special item must be medium since if it is small, there is no 2-bin whose level does not exceed $\frac{3}{4}$, and therefore there is no 2-bin whose weight is below 1.

If the special item is medium, its weight is $\frac{1}{2}$. For the 2-bin whose weight is below 1, its level is above $\frac{1}{2}$ (as there cannot be two bins of levels no larger than 1), and the total weight of 1-bins and 2-bin is at least $\frac{2}{3}X_1 - \frac{1}{6} + X_2 - \frac{1}{6}$. ■

Therefore, in this case we get $W \geq FF(I) - \frac{1}{3} - \frac{X_1}{3} \geq FF(I) - \frac{1}{3} - \frac{OPT(I)-1}{3} = FF(I) - \frac{OPT(I)}{3}$. Together with $W \geq \frac{3}{2}OPT(I)$ we find that $FF(I) \leq \frac{11}{6}OPT(I)$. ■

3.2.2 The case $k = 4$

We prove that FF is 2-competitive in the absolute sense for $k = 4$.

Theorem 21 *The absolute competitive ratio of FF for $k = 4$ is exactly 2, which is the best possible.*

The lower bound follows from Proposition 1. We define weights as follows. A big item, i.e. any item whose size exceeds $\frac{1}{2}$ has weight 1. A medium item, i.e., an item of size in $(\frac{1}{4}, \frac{1}{2}]$ has weight $\frac{1}{2}$. A small item, i.e., an item of size at most $\frac{1}{4}$ has weight $\frac{1}{4}$. Recall that the total weight of the items is denoted by W.

Lemma 22 *The weight of any bin of OPT is at most 2.*

Proof. Consider a bin B of OPT. Bin B can contain at most one big item. If B does not contain a big item, then the weight of any item is at most $\frac{1}{2}$, and since $|B| \leq 4$, the total weight is at most 2. Suppose now that B contains a big item. Out of the remaining (at most) three items, at most one item can be medium, and the total weight is at most $1 + \frac{1}{2} + 2 \cdot \frac{1}{4} = 2$, and the claim follows. ■

Claim 23 *Every bin that has a big item has total weight of at least 1. Every 4-bin has total weight of at least 1. Every 2^+-bin that does not have any small items has total weight of at least 1. Every bin that has items of total size above $\frac{3}{4}$ has total weight of at least 1.*

Proof. The first property holds as the weight of a big item is 1. The second property holds as the weight of any item is at least $\frac{1}{4}$. The third claim holds as two medium items have total weight of 1. Finally, we prove the last claim. Given the previous claims, it is sufficient to consider a 2-bin without a big item. If the bin has two medium items, then their total size is 1. There are no other cases as the total size of a small item and another item that is not big is at most $\frac{1}{4} + \frac{1}{2} = \frac{3}{4}$. If a 3-bin only has small items, then the total size of its items is at most $\frac{3}{4}$. A 3-bin that has at most two small items has an item of weight at least $\frac{1}{2}$, so its total weight is at least 1. ■

Claim 24 *Given an input L, the total weight of the input items is at least $FF(L) - \frac{3}{4}$.*

Proof. We calculate the total weight based on the packing of FF. Every 1-bin, except for possibly one such bin, has a big item, thus there is at most one 1-bin whose weight is below 1 (and this weight is at least $\frac{1}{4}$). If all 2^+-bins have weights of at least 1, then we are done as all 2^+-bins, all 4-bins, and all 1-bins except for possibly one bin (that has weight of at least $\frac{1}{4}$) have total weights of at least 1.

Otherwise, there must be a 2^+-bin of level of at most $\frac{3}{4}$, that does not have a big item, and its weight is strictly below 1. Consider the first such bin according to the ordering of FF, and call it B. This bin must have a small item, as the weight of any item that is not small is at least $\frac{1}{2}$, and the total weight of at least two such items is at least 1. All items that are packed into bins that appear later in the ordering of FF can only be medium or big. Thus, all further 2^+-bins have total weights of at least 1, and if there is a 1-bin that does not have a big item, then it must have a medium item. Moreover, if B has level of at most $\frac{1}{2}$, then all 1-bins have big items. In this last case, all bins except for B have weights of at least 1, while B has weight of at least $\frac{1}{2}$. Otherwise, if the level of B is in $(\frac{1}{2}, \frac{3}{4}]$, then we have the following cases. If B is a 2-bin, then it has one medium item and one small item, and its weight is $\frac{3}{4}$. If B is a 3-bin, then it has weight of $\frac{3}{4}$ as well (as its weight is below 1, and a 3-bin has weight of at least $\frac{3}{4}$). Thus, all bins except for B and possibly one 1-bin with a medium item have total weights of at least 1, while these two bins have weights of at least $\frac{3}{4}$ and $\frac{1}{2}$, respectively, and the claim follows. ■

We have $W \leq 2 \cdot OPT(L)$ and $W \geq FF(L) - \frac{3}{4}$. Thus, $FF(L) - 2 \cdot OPT(L) \leq \frac{3}{4}$, which implies (by integrality) $FF(L) \leq 2 \cdot OPT(L)$.

3.3 A simple algorithm with an absolute competitive ratio 2 for $k = 5$

In the cases $k \geq 5$, FF has a strictly higher absolute competitive ratio (than 2), as can be seen from the following simple example. Consider an input consisting of $2k$ items of size $\frac{1}{2} + \varepsilon$, $2k$ items of size $\frac{1}{3} + \varepsilon$, and $2k(k-2)$ items of size ε, where $0 < \varepsilon < 1/(6k)$. An optimal solution has $2k$ bins, each containing one item of size $\frac{1}{2} + \varepsilon$, one item of size $\frac{1}{3} + \varepsilon$, and $k - 2$ items of size ε. If the items are given sorted by non-decreasing size, FF creates $2(k-2)$ bins, each packed with k items of size ε, k bins, each packed with two items of size $\frac{1}{3} + \varepsilon$, and $2k$ bins, each packed with one item of size $\frac{1}{2} + \varepsilon$. The absolute competitive ratio is $\frac{5k-4}{2k} = 2.5 - \frac{2}{k} > 2$ for $k \geq 5$.

For $k = 5$, we present a different algorithm that is based on an adaptation of FF, and its absolute competitive ratio is 2. For this algorithm, as for FF, the ordering (or indexing) of bins is by the time of opening, that is, a bin appears earlier in the ordering if it received its first item earlier. The algorithm ALG acts as follows. A new item i is assigned into a minimum indexed bin B that satisfies the following conditions. The first condition that its level is at most $1 - s_i$, the second condition is that its current number of items is at most 4, and the third condition is that if B currently contains four items, then after assigning i, its level will be at least $\frac{1}{2}$. If there is no such bin, then the item is packed into an empty bin.

A regular bin is a bin of ALG that is a 2-bin or a 3-bin. We treat 1-bins, 4-bins, and 5-bins separately. A large 1-bin is a 1-bin containing an item of size above $\frac{1}{2}$.

Lemma 25 • *The level of any 5-bin is at least $\frac{1}{2}$.*

- *The level of any regular bin, except for at most one bin is at least $\frac{2}{3}$.*

- *For a pair of bins B, B', the total size of items packed into B and B' is above* 1 *in the following two cases.*
    1. *None of the bins contains more than three items.*
    2. *One of the bins is a large* 1*-bin, and the other contains four items.*

Proof. By the third condition, ALG never creates a 5-bin whose level is below $\frac{1}{2}$. For regular bins, ALG simply applies FF (without cardinality constraints) on the subsequence of items of these bins, and thus the claim follows from Claim 16.

Assume without loss of generality that B' appears after B in the ordering of ALG. If B contains at most three items, then when the first item of B' is packed, the third condition is irrelevant, the second condition holds, and thus the first condition does not hold, and the total size of items packed into the two bins exceeds 1.

We are left with the case that B contains four items when the unique item i of B' arrives, and $s_i > \frac{1}{2}$ (the case that B' has four items but B has one item was already considered). Thus, if i is packed into B, the third condition must hold, and therefore the first condition does not hold, and the total size of items packed into the two bins exceeds 1. ■

Consider an input L. For the output of ALG applied on L, let f denote the number of 4-bins, let d_1 be the number of 1-bins whose items have sizes above $\frac{1}{2}$ (such bins are called large 1-bins), and let d_0 be the number of 1-bins whose items have sizes no larger than $\frac{1}{2}$ (such bins are called small 1-bins). By Lemma 25, $d_0 \leq 1$ must hold.

Theorem 26 *The absolute competitive ratio of ALG is exactly* 2.

Proof. By Proposition 1, it is sufficient to prove an upper bound. We distinguish two cases as follows.
Case 1. $f < d_1$. We match 4-bins and large 1-bins into pairs arbitrarily, leaving at least one large 1-bin unmatched. The remaining bins that are not 5-bins (regular bins, and a small 1-bin, if it exists) are also matched into pairs, and if the number of these bins is odd, one of them is matched to an unmatched large 1-bin. The total size of items of any matched pair is above than 1, the level of every remaining large 1-bin is above $\frac{1}{2}$, and the level of any 5-bin is at least $\frac{1}{2}$, by Lemma 25. We find that the total size of items S satisfies $S \geq \frac{ALG}{2}$, thus $OPT(L) \geq S \geq \frac{ALG}{2}$.
Case 2. $f \geq d_1$. For an item of size x, we define its weight to be $w(x) = 1 + 3x$. Let W denote the total weight of all items of L. For any bin, the total weight of its items is at most 8, as it has at most five items of a total size of at most 1. Match every large 1-bin to a 4-bin. For each such pair, the total size is above 1, and the number of items is 5, thus the total weight of the items of every such pair of bins is at least 8. Every remaining 4-bin has four items, and their total weight is at least 4. Similarly, every 5-bin has a total weight above 5. Every regular bin, except for at most one such bin, has a total size of items of at least $\frac{2}{3}$, and at least two items, so its weight is at least $2 + \frac{2}{3} \cdot 3 = 4$. Thus, on average, all bins have weights of at least 4, except for possibly a small 1-bin, if it exists, and a regular bin of level below $\frac{2}{3}$, if it exists. If none of those exists, we find $8 \cdot OPT(L) \geq W \geq 4 \cdot ALG(L)$, and we are done. If both such bins exist, then the total size of

items in those bins is above 1 by Lemma 25, and there are at least three items, so the total weight is at least 6. We find $W \geq 4 \cdot (ALG(L) - 2) + 6$. If $d_0 = 1$ but no regular bin of level below $\frac{2}{3}$ exists, then the weight of the small 1-bin is at least 1, and $W \geq 4 \cdot (ALG(L) - 1) + 1$. If $d_0 = 0$, but there is a regular bin of level below $\frac{2}{3}$, then the weight of this regular bin is at least 2, and $W \geq 4 \cdot (ALG(L) - 1) + 2$. In all three last cases, $8OPT(L) \geq W \geq 4 \cdot ALG(L) - 3$, or alternatively, $ALG(L) \leq 2 \cdot OPT(L) + 3/4$. By integrality, we get $ALG(L) \leq 2 \cdot OPT(L)$. ■

References

[1] L. Babel, B. Chen, H. Kellerer, and V. Kotov. Algorithms for on-line bin-packing problems with cardinality constraints. *Discrete Applied Mathematics*, 143(1-3):238–251, 2004.

[2] J. Balogh, J. Békési, Gy. Dósa, J. Sgall, and R. van Stee. The optimal absolute ratio for online bin packing. In *Proc. of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA2015)*, pages 1425–1438.

[3] J. Balogh, J. Békési, and G. Galambos. New lower bounds for certain classes of bin packing algorithms. *Theoretical Computer Science*, 440-441:1–13, 2012.

[4] J. Boyar, Gy. Dósa, and L. Epstein. On the absolute approximation ratio for first fit and related results. *Discrete Applied Mathematics*, 160(13-14):1914–1923, 2012.

[5] A. Caprara, H. Kellerer, and U. Pferschy. Approximation schemes for ordered vector packing problems. *Naval Research Logistics*, 50(1):58–69, 2003.

[6] Gy. Dósa and L. Epstein. The tight asymptotic approximation ratio of First Fit for bin packing with cardinality constraints. Submitted.

[7] Gy. Dósa and J. Sgall. First fit bin packing: A tight analysis. In *Proc. of the 30th International Symposium on Theoretical Aspects of Computer Science (STACS2013)*, pages 538–549, 2013.

[8] L. Epstein. Online bin packing with cardinality constraints. *SIAM Journal on Discrete Mathematics*, 20(4):1015–1030, 2006.

[9] L. Epstein, C. Imreh, and A. Levin. Class constrained bin packing revisited. *Theoretical Computer Science*, 411(34-36):3073–3089, 2010.

[10] L. Epstein and A. Levin. AFPTAS results for common variants of bin packing: A new method for handling the small items. *SIAM Journal on Optimization*, 20(6):3121–3145, 2010.

[11] H. Fujiwara and K. Kobayashi. Improved lower bounds for the online bin packing problem with cardinality constraints. *Journal of Combinatorial Optimization*, 29(1):67–87, 2015.

[12] M. R. Garey, R. L. Graham, D. S. Johnson, and A. C.-C. Yao. Resource constrained scheduling as generalized bin packing. *Journal of Combinatorial Theory Series A*, 21(3):257–298, 1976.

[13] D. S. Johnson. *Near-optimal bin packing algorithms.* PhD thesis, MIT, Cambridge, MA, 1973.

[14] D. S. Johnson. Fast algorithms for bin packing. *Journal of Computer and System Sciences*, 8:272–314, 1974.

[15] D. S. Johnson, A. Demers, J. D. Ullman, M. R. Garey, and R. L. Graham. Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM Journal on Computing*, 3:256–278, 1974.

[16] H. Kellerer and U. Pferschy. Cardinality constrained bin-packing problems. *Annals of Operations Research*, 92:335–348, 1999.

[17] K. L. Krause, V. Y. Shen, and H. D. Schwetman. Analysis of several task-scheduling algorithms for a model of multiprogramming computer systems. *Journal of the ACM*, 22(4):522–550, 1975.

[18] K. L. Krause, V. Y. Shen, and H. D. Schwetman. Errata: "Analysis of several task-scheduling algorithms for a model of multiprogramming computer systems". *Journal of the ACM*, 24(3):527–527, 1977.

[19] C. C. Lee and D. T. Lee. A simple online bin packing algorithm. *Journal of the ACM*, 32(3):562–572, 1985.

[20] F. M. Liang. A lower bound for on-line bin packing. *Information Processing Letters*, 10(2):76–79, 1980.

[21] P. Ramanan, D. J. Brown, C. C. Lee, and D. T. Lee. Online bin packing in linear time. *Journal of Algorithms*, 10:305–326, 1989.

[22] S. S. Seiden. On the online bin packing problem. *Journal of the ACM*, 49(5):640–671, 2002.

[23] H. Shachnai and T. Tamir. Polynomial time approximation schemes for class-constrained packing problems. *Journal of Scheduling*, 4(6):313–338, 2001.

[24] H. Shachnai and T. Tamir. Tight bounds for online class-constrained packing. *Theoretical Computer Science*, 321(1):103–123, 2004.

[25] J. D. Ullman. The performance of a memory allocation algorithm. Technical Report 100, Princeton University, Princeton, NJ, 1971.

[26] A. van Vliet. An improved lower bound for online bin packing algorithms. *Information Processing Letters*, 43(5):277–284, 1992.

[27] E. C. Xavier and F. K. Miyazawa. The class constrained bin packing problem with applications to video-on-demand. *Theoretical Computer Science*, 393(1-3):240–259, 2008.

[28] A. C. C. Yao. New algorithms for bin packing. *Journal of the ACM*, 27:207–227, 1980.

[29] G. Zhang. Private communication.