

Online Bin Packing with Cardinality Constraints Resolved

János Balogh¹, József Békési², György Dósa³, Leah Epstein⁴, and Asaf Levin⁵

- 1 Department of Applied Informatics, Gyula Juhász Faculty of Education, University of Szeged, Hungary
balogh@jgypk.u-szeged.hu
- 2 Department of Applied Informatics, Gyula Juhász Faculty of Education, University of Szeged, Hungary
bekesi@jgypk.u-szeged.hu
- 3 Department of Mathematics, University of Pannonia, Veszprém, Hungary
dosagy@almos.vein.hu
- 4 Department of Mathematics, University of Haifa, Haifa, Israel
lea@math.haifa.ac.il
- 5 Faculty of Industrial Engineering and Management, The Technion, Haifa, Israel
levinas@ie.technion.ac.il

Abstract

Cardinality constrained bin packing or bin packing with cardinality constraints is a basic bin packing problem. In the online version with the parameter $k \geq 2$, items having sizes in $(0, 1]$ associated with them are presented one by one to be packed into unit capacity bins, such that the capacities of bins are not exceeded, and no bin receives more than k items. We resolve the online problem in the sense that we prove a lower bound of 2 on the overall asymptotic competitive ratio. This closes the long standing open problem of finding the value of the best possible overall asymptotic competitive ratio, since an algorithm of an absolute competitive ratio 2 for any fixed value of k is known. Additionally, we significantly improve the known lower bounds on the asymptotic competitive ratio for every specific value of k . The novelty of our constructions is based on full adaptivity that creates large gaps between item sizes. Thus, our lower bound inputs do not follow the common practice for online bin packing problems of having a known in advance input consisting of batches for which the algorithm needs to be competitive on every prefix of the input. Last, we show a lower bound strictly larger than 2 on the asymptotic competitive ratio of the online 2-dimensional vector packing problem, and thus provide for the first time a lower bound larger than 2 on the asymptotic competitive ratio for the vector packing problem in any fixed dimension.

1998 ACM Subject Classification F.2.2 Sequencing and Scheduling, G.2.1 Combinatorial Algorithms

Keywords and phrases Online algorithms, bin packing, cardinality constraints, lower bounds

Digital Object Identifier 10.4230/LIPIcs.ESA.2017.10

1 Introduction

Bin packing with cardinality constraints (CCBP, also called cardinality constrained bin packing) is a well-known variant of bin packing [18, 19, 17, 9, 10, 11, 15]. In this problem, a parameter k is given. Items of indices $1, 2, \dots, n$, where item i has a size $s_i \in (0, 1]$ are to be



© János Balogh, József Békési, György Dósa, Leah Epstein, and Asaf Levin;
licensed under Creative Commons License CC-BY

25th Annual European Symposium on Algorithms (ESA 2017).

Editors: Kirk Pruhs and Christian Sohler; Article No. 10; pp. 10:1–10:14

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

split into subsets called bins, such that the total size of items packed into each bin is at most 1, and no bin has more than k items. In the standard bin packing problem, only the first condition is required.

CCBP is a special case of vector packing (VP) [14]. In VP with dimension $d \geq 2$, a set of items, where every item is a non-zero d -dimensional vector whose components are rational numbers in $[0, 1]$, are to be split into subsets (called bins in this case as well) such that the vector sum of every subset does not exceed 1 in any component. Given an input for CCBP, an input for VP is created as follows. For every item, let the first component be $\frac{1}{k}$, the second component is s_i , and the remaining components are equal to zero (or to $\frac{1}{k}$).

In this paper we study online algorithms, which receive input items one by one, and pack each new item irrevocably before the next item is presented, into an empty (new) bin or non-empty bin. Such algorithms receive an input as a sequence, while offline algorithms receive an input as a set. By the definition of CCBP, an item i can be packed into a non-empty bin B if the packing is feasible both with respect to the total size of items already packed into that bin and with respect to the number of packed items (i.e., the bin contains items of total size at most $1 - s_i$ and it contains at most $k - 1$ items). An optimal offline algorithm, which uses a minimum number of bins for packing the items, is denoted by OPT . For an input L and algorithm A , we let $A(L)$ denote the number of bins that A uses to pack L . We also use $OPT(L)$ to denote the number of bins that OPT uses for a given input L . The absolute competitive ratio of an algorithm A is the supremum ratio over all inputs L between the number of its bins $A(L)$ and the number of the bins of OPT , $OPT(L)$. The asymptotic approximation ratio is the limit of absolute approximation ratios R_K when K tends to infinity and R_K takes into account only inputs for which OPT uses at least K bins, that is the asymptotic competitive ratio of A is

$$\lim_{K \rightarrow \infty} \sup_{OPT(L) \geq K} \frac{A(L)}{OPT(L)}.$$

The term *competitive ratio* is used for online algorithms instead of *approximation ratio* and it is equivalent. In this paper we mostly deal with the asymptotic competitive ratio, and also refer to it by the term competitive ratio. When we discuss the absolute competitive ratio, we use this last term explicitly.

In this paper, we resolve the long standing open problem of online CCBP, in the sense that we find the best overall asymptotic competitive ratio and the best overall absolute competitive ratio. An algorithm with an asymptotic competitive ratios of 2 has been designed by Babel et al. [4], and a similar algorithm was shown to have an absolute competitive ratio of 2 [6] (earlier, it was known that the competitive ratio of a suitable variant of First Fit is below 2.7 for any k [18]). However, prior to this work, all lower bounds were strictly smaller than the best lower bounds for standard bin packing [23, 5]. With the exception of the case $k = 2$ for which simple algorithms have competitive ratios of 1.5 [18, 10], and a more sophisticated algorithm has a competitive ratio of at most 1.44721 [4], all lower bounds on the competitive ratio were implied by partial inputs of ones used to prove lower bounds for standard bin packing [24, 23, 5] (such lower bounds can be used for $k \geq \frac{1}{3}$ when all items have sizes no smaller than δ , for a fixed value $\delta > 0$), and modifications of such inputs [4, 12, 6]. That is, all lower bounds had the form where a number of lists may be presented, each list has a large number of items of a certain size (the sequence of sizes of the different lists is increasing, and the numbers of items in the lists are not necessarily equal). The unknown factor is the number of presented lists, that is, the input can stop after any of the lists. See Table 1 for values of previously known lower bounds (and note that for $k = 3, 4, 5, 6$ algorithms with competitive ratios strictly below 2 are known [10]).

■ **Table 1** Bounds for $2 \leq k \leq 10$. The middle column contains the previously best known asymptotic lower bounds on the asymptotic competitive ratio for CCBP with parameter k . The right column contains our improved lower bounds.

| Value of k | previous lower bound | new lower bound |
|------------------------|--|--------------------------------|
| 2 | 1.42764 [12] | $\frac{10}{7} \approx 1.42857$ |
| 3 | $\frac{3}{2} = 1.5$ [4] | 1.55642 |
| 4 | $\frac{3}{2} = 1.5$ [12] | 1.63330 |
| 5 | $\frac{3}{2} = 1.5$ [6] | 1.69776 |
| 6 | $\frac{3}{2} = 1.5$ [24] | 1.74093 |
| 7 | $\frac{217}{143} \approx 1.51748$ [6] | 1.77223 |
| 8 | $\frac{32}{21} \approx 1.52381$ [6] | 1.79634 |
| 9 | $\frac{189}{124} \approx 1.524194$ [6] | 1.81563 |
| 10 | $\frac{235}{154} \approx 1.52597$ [6] | 1.83148 |
| 200000 | 1.54037 [5] | 1.99999 |
| $k \rightarrow \infty$ | $\frac{248}{161} \approx 1.54037$ [5] | 2 |

In this work, we take a different approach for proving lower bounds, where many of the item sizes are based on the complete and precise action of the algorithm up to the time it is presented. While some ingredients of our approach were used for the very limited special case of $k = 2$ in the past [7, 4, 12], it was unclear how and if it could be used for $k > 2$. In a nutshell, in these lower bound sequences for $k = 2$, sub-inputs were constructed such that items packed in certain ways (for example, as the second item of a bin) had much larger sizes than items of the same sub-input packed in other ways. Here, we generalize the approach for larger values of k by defining careful constructions where *sufficiently large multiplicative gaps* are created. This requires much more delicate procedures where item sizes are defined.

Additionally, we improve the lower bounds for all values of k , and in particular, prove lower bounds above the best known lower bound on the competitive ratio for standard online bin packing, 1.54037 [5] for $k \geq 3$. Already for $k = 3$ our lower bound is above 1.55, and already for $k = 4$, our lower bound is above the competitive ratio of many algorithms for standard online bin packing (see for example [21, 22]).

Our result for CCBP provides, in particular, a lower bound of 2 for the asymptotic competitive ratio of VP in two dimensions. The previously known lower bounds for VP are as follows. The best results for constant dimensions are fairly low, and tending to 2 as the dimension d grows to infinity [13, 8, 7], while a lower bound of $\Omega(d^{1-\epsilon})$ was given by Azar et al. [2] for the case where both d and the optimal cost are functions of a common parameter n that grow to infinity when n grows to infinity, and thus this result does not give any lower bound on the competitive ratio for constant values of d (see also [1, 3] for results on vectors with small components). In particular, the best lower bound for $d = 2$ prior to this work was 1.67117 [13, 8, 7]. An upper bound of $d + 0.7$ on the competitive ratio is known [14]. We conclude this work by establishing a lower bound strictly larger than 2 on the competitive ratio of 2-dimensional VP, and thus we show here for the first time that the 2-dimensional VP is provably harder for online algorithms than its special case of CCBP.

Note that the offline CCBP problem is NP-hard in the strong sense, and approximation schemes are known for it [9, 11, 15]. We note that for online CCBP, it is sometimes the

case that the competitive ratio for some specific algorithms for CCBP is larger by 1 with comparison to that of the corresponding algorithms for standard bin packing [18, 16, 20, 10]. Interestingly, this is not the case with respect to the results shown in this paper.

1.1 Paper outline

We discuss general properties in Section 2, and we define procedures for constructing sub-inputs in Section 3. Our main result, an overall lower bound of 2 on the competitive ratio of any online algorithm for CCBP is proved in Section 4, and improved lower bounds for fixed values of k are given in Section 5. Our result for VP is established in Section 6. Omitted proofs appear in the full version of this work.

2 Preliminaries

The analysis of the lower bounds on the asymptotic competitive ratio of online algorithms will be based on the following lemma that basically allows us to disregard a constant number of bins in the costs of the optimal solution and the solution returned by the algorithm.

► **Lemma 1.** *Consider an algorithm ALG , such that the asymptotic competitive ratio of the algorithm ALG is at most R , where $R \geq 1$ is a fixed value, and let $f(n)$ denote a positive function such that $f(n) = o(n)$ and for any input, $ALG(I) \leq R \cdot OPT(I) + f(OPT(I))$. Let $C_a \geq 0$, $C_b \geq 0$ be constants. Assume that for a given integer N_0 , for any integer $n \geq N_0$ there is an input I^n for which $OPT(I^n) = \Omega(n)$, then we have*

$$R \geq \limsup_{n \rightarrow \infty} \frac{ALG(I^n) + C_a}{OPT(I^n) - C_b}.$$

Proof. We have

$$\frac{ALG(I^n) + C_a}{n} \leq R \cdot \frac{OPT(I^n) - C_b}{n} + \frac{C_a + R \cdot C_b}{n} + \frac{f(OPT(I^n))}{n}$$

for any $n \geq N_0$.

Since $ALG(I^n) + C_a \geq OPT(I^n) - C_b$ and $OPT(I^n) - C_b = \Omega(n)$ while $C_a + R \cdot C_b + f(OPT(I^n)) = o(n)$, letting n grow to infinity implies that

$$R \geq \limsup_{n \rightarrow \infty} \frac{ALG(I^n) + C_a}{OPT(I^n) - C_b}. \quad \blacktriangleleft$$

In what follows, we will use Lemma 1 as follows. We construct inputs whose size depends on a parameter N , so that the costs of optimal solutions increase with the input size. We will compare the cost of a fixed online algorithm ALG plus a suitable non-negative constant to the optimal cost minus a suitable non-negative constant by considering their ratio.

3 Constructions of sub-inputs

In this section we introduce the core of our lower bound constructions. In such constructions, we *adaptively* present inputs that are based on the behavior of the algorithm. More specifically, we define several procedures that construct sub-inputs according to certain conditions. Similarly to [4, 12, 7] (and other work on online problems), a new input item is presented at each time, where its size is based on the action of the algorithm on previous items. For example, if the previous item was packed into an empty bin, the size of the next item is

different from the size that would be used if the previous item is added to a non-empty bin. In order to ensure that the properties are satisfied, we will define invariants, and we will prove the specific properties that we need in the sequel via induction. The constructions use k as a parameter since they are defined to be used for CCBP. However, they can be used for any packing problem of items into bins and the property that k is the cardinality constraint is not used in the constructions of sub-inputs (it is used later in the analysis of inputs constructed using these sub-inputs). Thus, if the constructions are used for other problems like we do for VP, the parameter k should be specified.

In the first procedure, the most important property is that there will be a gap between two types of items constructed by applying the procedure, in the sense that the procedure creates items that will be called small and items that will be called large, any large item is larger than any small item, and there is a requirement on the size ratio that will be satisfied (a multiplicative gap between the size of the smallest large item and the largest small item). Such constructions differ from previous work [4, 12, 7] where only an additive gap was created. The gap was always positive, but it could be arbitrarily small. In particular, one limitation was that it was unknown how such an approach could be used for CCBP with parameter $k > 2$.

We will also use this method to construct sub-inputs with large items, such that there is a multiplicative gap in the differences between 1 and the items sizes. This new method will allow us to provide a tight overall result for CCBP, new and significantly improved lower bounds on the asymptotic competitive ratio for CCBP with fixed values of k , and our improved lower bound for VP.

3.1 Procedure SMALL

In this first procedure called SMALL, a rational value $0 < \varepsilon \leq 1$, and an integer upper bound N on the number of items to be presented are given. The goal is to present (at most) N items of sizes in $(0, \varepsilon]$, such that every item will be seen as either a small item or a large item, and such that any large item is more than k times larger than any small item. In fact, a stronger requirement on the item sizes will hold. Moreover, all item sizes will be rational. Given two logical conditions, C_1 and C_2 specified for each construction (such that for every packed item, exactly one of them holds), a new item will be defined as *small* if C_1 holds and it will be defined as *large* if C_2 holds. There is a third condition C_3 that is based on the packing of the prefix of items introduced so far, and the sub-input is stopped if C_3 holds.

Let N be an upper bound on the number of items that will be created by the procedure. Let $N' \leq N$ be the actual number of items (where N is known in advance and used for the sequence construction, while N' is not necessarily known in advance and it becomes known when C_3 holds for the first time). The item sizes $a_1, a_2, \dots, a_{N'}$ will be defined based on another sequence $x_1, x_2, \dots, x_{N'}$, such that $a_i = \varepsilon \cdot k^{-x_i}$ for $1 \leq i \leq N'$. The values x_i will be integral in order to ensure that the values a_i will be rational. There will also be two sequences of values $\tau_1, \dots, \tau_{N'}$ and $\rho_1, \dots, \rho_{N'}$, representing thresholds on item sizes of further items.

Let $\tau_0 = 2^{N+2}$, $\rho_0 = 2^{N+3}$, and $i = 1$. The process is defined as follows for any given value of i (such that $1 \leq i \leq N'$). Let $x_i = \frac{\tau_{i-1} + \rho_{i-1}}{2}$ (we will show that these values are integers). After the algorithm packs item i , if C_1 holds, let $\tau_i = \tau_{i-1}$ and $\rho_i = x_i$ and if C_2 holds, let $\tau_i = x_i$ and $\rho_i = \rho_{i-1}$. If C_3 holds or $i = N$, stop and otherwise increase i by 1.

Intuitively, the process is as follows. The interval (τ_i, ρ_i) contains the x_j values of all further items (with $j > i$), and for $j \leq i$, all items satisfying C_1 have x_j values in $[\rho_i, \rho_0)$ and all items satisfying C_2 have x_j values in $(\tau_0, \tau_i]$. In each iteration i , the new values τ_i, ρ_i are

defined such that these requirements are satisfied. In particular, the x_i values of any item satisfying C_1 are larger than those of items satisfying C_2 . Next, we establish the invariants of this procedure.

► **Lemma 2.** *Let N' be the number of items. For any i such that $1 \leq i \leq N'$, $\rho_i \leq \rho_{i-1}$ and $\tau_i \geq \tau_{i-1}$. Additionally, we have $\rho_i - \tau_i = 2^{N+2-i}$, all x_i values are integral, if item i satisfies C_1 , $x_i \geq \rho_{N'}$ and otherwise $x_i \leq \tau_{N'}$.*

Proof. We start with showing that the x_i values as well as ρ_i and τ_i are integral and $\rho_i - \tau_i = 2^{N+2-i}$. We prove this by induction. Indeed $\rho_0 = 2^{N+3}$ that is integral, $\tau_0 = 2^{N+2}$ that is an integer as well. Furthermore, $\rho_0 - \tau_0 = 2^{N+2}$, and $x_1 = 3 \cdot 2^{N+1}$ that is an integer, and no matter if the first item satisfies C_1 or C_2 , we have that both ρ_1 and τ_1 are integers, and $\rho_1 - \tau_1 = 2^{N+1}$. Thus, the cases $i = 0$ and $i = 1$ for the induction claim hold. Assume that $\rho_{i-1} - \tau_{i-1} = 2^{N+3-i}$ holds for some i where $1 \leq i \leq N' - 1$. Then,

$$x_i = \frac{\tau_{i-1} + \rho_{i-1}}{2} = \tau_{i-1} + \frac{\rho_{i-1} - \tau_{i-1}}{2} = \tau_{i-1} + 2^{N+3-i},$$

which is an integer for $1 \leq i \leq N$, since τ_{i-1} is an integer. Moreover, if $\tau_i = \tau_{i-1}$ and $\rho_i = x_i$, then $\rho_i - \tau_i = x_i - \tau_{i-1} = \frac{\rho_{i-1} - \tau_{i-1}}{2}$, and otherwise $\tau_i = x_i$ and $\rho_i = \rho_{i-1}$, then $\rho_i - \tau_i = \rho_{i-1} - x_i = \frac{\rho_{i-1} - \tau_{i-1}}{2}$. In both cases, $\rho_i - \tau_i = 2^{N+2-i}$ and both τ_i and ρ_i are integers. Since, in particular, for any i , $\rho_i > \tau_i$ holds and x_{i+1} is their average, we find $\tau_i < x_{i+1} < \rho_i$. Thus, $\rho_i \leq \rho_{i-1}$ and $\tau_i \geq \tau_{i-1}$ holds for any i .

Finally, since in the case that item i satisfies C_1 , we let $\rho_i = x_i$, and in the case that item i satisfies C_2 , we let $\tau_i = x_i$, we get $x_i = \rho_i \geq \rho_{i+1} \geq \dots \geq \rho_{N'}$ in the first case, and $x_i = \tau_i \leq \tau_{i+1} \leq \dots \leq \tau_{N'}$ in the second case. ◀

► **Corollary 3.** *For any item i , $a_i \in \left(\varepsilon \cdot k^{-2^{N+3}}, \varepsilon \cdot k^{-2^{N+2}} \right)$, and in particular $a_i \leq \frac{1}{k^4}$. For any item i_1 satisfying C_1 and any item i_2 satisfying C_2 , it holds that $\frac{a_{i_2}}{a_{i_1}} > k$.*

Note that it is possible that the constructed input is such that there are only items satisfying C_1 or only items satisfying C_2 .

Proof. The first claim holds by definition. Since we have $x_{i_1} \geq \rho_{N'}$ and $x_{i_2} \leq \tau_{N'}$, we get $\frac{a_{i_2}}{a_{i_1}} > k^{\rho_{N'} - \tau_{N'}}$, Using $\rho_{N'} - \tau_{N'} = 2^{N+2-N'} \geq 4$ as $N' \leq N$, we find $\frac{a_{i_2}}{a_{i_1}} \geq k^4 > k$. ◀

3.2 Procedure LARGE

The second type of input is such that all items have sizes in $(1 - \varepsilon, 1)$ for a given value $\varepsilon > 0$. The construction is the same as before, but the size of the i th item is $b_i = 1 - a_i$. The terms “small” and “large” refer to the difference between the size of the item and 1.

► **Corollary 4.** *All b_i for $1 \leq i \leq N$ are in $(1 - \varepsilon \cdot k^{-2^{N+2}}, 1 - \varepsilon \cdot k^{-2^{N+3}})$. The sizes of any small item i_s and any large item i_l satisfy $1 - b_{i_l} > k \cdot (1 - b_{i_s})$.*

3.3 Procedure SMALLandLARGE

We will also use a procedure where the conditions C_1 and C_2 are not fixed, and they are based on additional properties of the packing and the input that has been presented so far. Moreover, in this case the size of each item is based on a_i , but it is fixed for each item separately (it will be either a_i or $1 - a_i$). In this construction the sub-input will be decomposed into *parts* where for an item of an odd indexed part the size of the item will be $1 - a_i$, whereas for an item of an even indexed part the size of the item will be a_i . The definitions of C_1 and C_2 will also depend on the parity of the index of the part containing the item. This procedure is called SMALLandLARGE.

4 A lower bound of 2 for CCBP

The general structure of inputs constructed in this section is as follows. There are a large number of very small items, such that the first item packed into a bin by the algorithm is significantly larger than small items packed as a second item or later. Afterwards, there are two cases. In the first case there are very large items (of sizes almost 1) that can be combined with $k - 1$ items that arrived earlier, but only with those that are smaller. Thus, an optimal solution can pack all items densely except for those items that are first in their bins (for the algorithm). The algorithm cannot use its previously packed bins again to pack new items, and therefore the best approach is to pack a large number of items into each bin (otherwise the percentage of larger small items is larger, which makes the optimal packing more sparse, but the algorithm has an even larger number of bins, and the effect of the last property is more significant). Another option is that instead of the very large items, items slightly larger than $\frac{1}{2}$ will arrive, in which case it turns out that the algorithm should have packed $k - 1$ items into each bin (so that a new item could be still packed there). For very large values of k , the two values $k - 1$ and k are not very different, and since the algorithm does not know which items will arrive, packing $k - 1$ items into each bin (if k is very large) is a good strategy. The result of packing $k - 1$ items into each bin is that in the first case the very large items increase the number of bins roughly by a factor of 2, while an optimal solution has relatively few bins with k small items. Note that the order of options in the construction below is reversed for the sake of convenience.

Let N be a large integer. Apply procedure SMALL with $\varepsilon = 1$ for the construction of N items (i.e., condition C_3 never happens). The condition C_2 is that the item is packed as the first item of some bin (into an empty bin), and the condition C_1 is that the item is packed into a non-empty bin. The item sizes are no larger than $\frac{1}{k^4}$. The multiplicative gap between the smallest large item and the largest small item is larger than k . The N items presented so far will be called the first phase items. Let $\delta > 0$ denote the largest size of any first phase item packed not as a first item of a bin (the largest small item). Let $\alpha = k \cdot \delta$. Any first phase item that is packed as the first item of a bin (a large item) has size strictly above α . Let $\Delta < \frac{1}{k^3}$ be the largest size of any first phase item. Obviously, $1 - k\Delta > 1 - \frac{1}{k^2} > \frac{1}{2}$.

For the first phase items, let X_k denote the number of bins packed by the algorithm that contain k items, and let Y denote the number of other bins (such that there are $X_k + Y$ bins in total after N items have been presented).

The first phase items are followed by another set of items called the second phase items. This set of items is selected out of two possible options. The first option is that $\lceil \frac{N}{k-1} \rceil$ items of size $1 - k\Delta$ arrive, and the second option is that $\lceil \frac{N - X_k - Y}{k-1} \rceil$ items of size $1 - \alpha = 1 - k\delta$ arrive. In both cases it is possible to create an offline solution such that each bin (except for possibly two bins) has k items. In the first case, an offline solution has $\lceil \frac{N}{k-1} \rceil$ bins, each with one item of size $1 - k\Delta$ and an arbitrary subset of $k - 1$ first phase items (the last bin may have a smaller number of such items). Such a solution is optimal. In the second case, an offline solution has $\lceil \frac{N - X_k - Y}{k-1} \rceil$ bins, each with one item of size $1 - k\delta$ and $k - 1$ small first phase items, and $\lceil \frac{X_k + Y}{k} \rceil$ bins with k large first phase items (for each one of these two bin types, the last bin may have a smaller number of such items). Indeed the last solution is an optimal solution though we will only use that it is a feasible solution.

In the first case, the algorithm cannot use the bins that already have k items for packing second phase items, and its cost is at least $X_k + \lceil \frac{N}{k-1} \rceil \geq X_k + \frac{N}{k-1}$. In the second case, the algorithm cannot use any of its bins to pack any second phase item, as each bin has a large

first phase item of size above α , so its cost is

$$X_k + Y + \left\lceil \frac{N - X_k - Y}{k - 1} \right\rceil \geq X_k + Y + \frac{N - X_k - Y}{k - 1}.$$

We call the two inputs (of the two cases) I_1 and I_2 . Obviously, since each input consists of more than N items, $OPT(I_1) = \Omega(\frac{N}{k})$ and $OPT(I_2) = \Omega(\frac{N}{k})$ hold. Letting $N = kn$ provides an input I^n as required. By Lemma 1, we will analyze modified competitive ratios of the form $\frac{ALG(I)+C_a}{OPT(I)-C_b}$ for fixed constants C_a and C_b .

For the input I_1 , $OPT(I_1) - 1 \leq \frac{N}{k-1}$ and $ALG(I_1) \geq X_k + \frac{N}{k-1}$. For the input I_2 , $OPT(I_2) - 2 \leq \frac{N-X_k-Y}{k-1} + \frac{X_k+Y}{k}$ and $ALG(I_2) \geq X_k + Y + \frac{N-X_k-Y}{k-1}$.

First, we analyze the competitive ratio r for input I_2 and show that it tends to 2 as k grows to infinity. Let $Z = X_k + Y$. We have $OPT(I_2) - 2 \leq \frac{N-Z}{k-1} + \frac{Z}{k}$ and $ALG(I_2) \geq Z + \frac{N-Z}{k-1}$. Thus,

$$r \geq \frac{kZ(k-1) + k(N-Z)}{k(N-Z) + (k-1)Z} = \frac{Z(k^2 - 2k) + kN}{kN - Z}.$$

Since $Z \geq \frac{N}{k}$ and the last lower bound on r is a ratio between an increasing function of Z and a decreasing function of Z , we conclude that by substituting $\frac{N}{k}$ instead of Z in the last bound, we achieve a valid lower bound on r . Thus, we have $r \geq \frac{N(k-2)+kN}{kN-\frac{N}{k}} = \frac{2-2/k}{1-1/(k^2)} = \frac{2k}{k+1}$ and the last bound tends to 2 when k grows to infinity. By Lemma 1, the overall (asymptotic) competitive ratio is at least 2. Since there is a 2-competitive algorithm for any value of k [4] (even for the absolute competitive ratio [6]), we establish the following.

► **Theorem 5.** *The overall best possible asymptotic and absolute competitive ratios for bin packing with cardinality constraints are equal to 2.*

To obtain a better lower bound on the asymptotic competitive ratio r for a fixed value of $k \geq 3$, we use I_1 as well. By $r \geq \frac{ALG(I_1)}{OPT(I_1)-1} \geq \frac{X_k+N/(k-1)}{N/(k-1)}$ we have $(k-1)X_k \leq (r-1) \cdot N$. By counting arguments, $N \leq kX_k + (k-1)Y$ holds, and we get $X_k \geq N - (k-1)Z$, and $(r-1)N \geq (k-1)X_k \geq (k-1)(N - (k-1)Z) = (k-1)N - (k-1)^2 \cdot Z$. Rearranging gives

$$Z \geq \frac{(k-r)N}{(k-1)^2}.$$

As we saw earlier, by using I_2 we have $r \geq \frac{Z(k^2-2k)+kN}{kN-Z}$, which is equivalent to

$$Z(k^2 - 2k + r) \leq kN(r - 1).$$

Combining the lower bound and upper bound on Z results in

$$\frac{(k-r)N(k^2 - 2k + r)}{(k-1)^2} \leq kN(r - 1),$$

or equivalently

$$r^2 + r(k^3 - k^2 - 2k) - (2k^3 - 4k^2 + k) \geq 0.$$

Since $k^3 - k^2 - 2k \geq 0$ holds for $k \geq 2$ and $2k^3 - 4k^2 + k > 0$ holds for $k \geq 2$, it is sufficient to find the (unique) positive root which is equal to $\frac{2k+k^2-k^3+\sqrt{(k^3-k^2-2k)^2+4(2k^3-4k^2+k)}}{2}$. The last expression is a lower bound on r and thus the following holds.

► **Theorem 6.** *For any $k \geq 3$, the asymptotic competitive ratio for bin packing with cardinality constraints is at least*

$$\frac{2k + k^2 - k^3 + \sqrt{k^6 - 2k^5 - 3k^4 + 12k^3 - 12k^2 + 4k}}{2}.$$

The last lower bound is equal to approximately 1.54983 for $k = 3$, 1.63330 for $k = 4$, 1.69047 for $k = 5$, 1.73214 for $k = 6$, 1.76388 for $k = 7$, 1.78888 for $k = 8$, 1.80909 for $k = 9$, and 1.82575 for $k = 10$. For $k = 2$ the resulting lower bound is $\sqrt{2}$ and the construction (for the case $k = 2$) is indeed similar to that of [7, 4].

5 Better lower bounds for CCBP for some small values of k

In this section we prove the next theorem that improves the resulting bounds of Theorem 6 for these values of k .

► **Theorem 7.** *The following approximate values are lower bounds on the asymptotic competitive ratio: The value 1.42857 for $k = 2$ (the exact value of this lower bound is $\frac{10}{7}$), 1.55642 for $k = 3$, 1.69776 for $k = 5$, 1.74093 for $k = 6$, 1.77223 for $k = 7$, 1.79634 for $k = 8$, 1.81563 for $k = 9$, and 1.83148 for $k = 10$.*

6 Vector packing

As explained in the introduction, vector packing is a generalization of CCBP, and thus the results of the previous sections imply, in particular, a lower bound of 2 on the asymptotic competitive ratio for VP in two or more dimensions. In this section we show that VP is more general, by improving the result, and showing a lower bound above 2 for VP with constant dimensions. Our result is the first lower bound strictly above 2 for any fixed dimension VP (recall that currently, the best known upper bound for d -dimensional VP is $d + 0.7$ and for 2-dimensional VP 2.7 [14]). We prove the result for two dimensions (and the result for higher dimensions follows since the asymptotic competitive ratio is monotone in the dimension, as any d -dimensional vector can be augmented by $d' - d$ zeroes to become a d' -dimensional vector). Once again we consider a fixed deterministic online algorithm *ALG*, but this time it is an algorithm for VP. Let R be the asymptotic competitive ratio.

The main idea of the lower bound is as follows. First, there are items whose first component is $\frac{1}{k}$ for an appropriately chosen integer k , while the second components are very small. The items are such that the second components are sufficiently larger for items packed first into their bins by the algorithm compared to those that are not packed first. Afterwards, one option is that the following items have a very large second component and their first component is zero (this is equivalent to the items in the construction for CCBP). Every such item could be packed with k items that arrived earlier, but never with the first item of a bin of the algorithm, and thus the new items require new bins, while an optimal solution can pack almost everything densely. For this option it is most profitable for the algorithm to pack k items in each bin. In the other cases, it turns out that it is better to pack much less than k items per bin, as further items will have first components of $\frac{a}{k}$ for an integer value of a (which is selected based on the action of the algorithm). Those items will have second components above $\frac{1}{3}$, and there may be further items whose second components are above $\frac{1}{2}$.

Let $k \geq 10$ be a large integer. The set of inputs we define will consist of at most three phases (where a phase is a sub-input). The first phase of the input is based on the construction for CCBP as follows. For a large integer $N \geq 1000$, there are N items whose

10:10 Online Bin Packing with Cardinality Constraints Resolved

first component is $\frac{1}{k}$. The second components of items are constructed using procedure SMALL with $\varepsilon = k^{-2^{N+4}}$, such that SMALL is applied for the construction of N elements (i.e., condition C_3 never happens). The condition C_2 is that the item is packed as the first item of some bin (i.e., it is packed into an empty bin), and the condition C_1 is that the item is packed into a non-empty bin. The N (two-dimensional) items presented so far will be called the first phase items. The second components of the first phase items are no larger than $k^{-2^{N+4}} \leq \frac{1}{k^4}$. Due to the value of the first component, in any packing every bin has at most k first phase items. A first phase item packed as the first item of a bin will be called large and any other first phase item will be called small.

The multiplicative gap between the smallest second component of any large item and the largest second component of any small item is greater than k . Let $\delta > 0$ denote the largest second component of any small first phase item. Let $\alpha = k \cdot \delta$. Any large first phase item has a second component strictly above α . Let $\Delta < \frac{1}{k^3}$ be the largest second component of any first phase item. Obviously, $1 - \alpha = 1 - k\delta > 1 - \frac{1}{k^3} > 0.999$.

Let X_i denote the number of bins packed with i first phase items and let $\Theta = \frac{(\sum_{i=1}^k X_i)}{N}$, where $\Theta \leq 1$ as every bin has at least one item out of the N items. Let the input of first phase items be denoted by I . At this time, any k items can be packed into a bin, and thus $OPT(I) \leq \lceil \frac{N}{k} \rceil$. If $ALG(I) = \Theta N \geq \frac{3N}{k}$, we get $R \geq \frac{ALG(I)}{OPT(I)-1} \geq 3$. Thus, we assume in what follows that $\Theta < \frac{3}{k}$. Since every bin of the algorithm contains exactly one large item and the remaining items are small, there are $\Theta N < \frac{3N}{k}$ large items and at least $N - \Theta N > \frac{(k-3)N}{k} \geq \frac{7N}{10}$ small items.

The first option for the second part of the input is similar to the construction for CCBP (the second part of the input will also be the last part of the input in this specific case). The next phase of items will consist of $\lceil \frac{N-\Theta N}{k} \rceil$ items called second phase items, whose first component is zero and the second component is $1 - \alpha = 1 - k\delta$. This input (consisting of the first phase items and the second phase items) is called I' . By the following lemma we have $1 + (k-1)\Theta \leq R$.

► **Lemma 8.** *We have $ALG(I') = \Theta N + \lceil \frac{N-\Theta N}{k} \rceil \geq \Theta N + \frac{N-\Theta N}{k} = \frac{N+(k-1)\Theta N}{k}$ and $OPT(I') - 2 \leq \frac{N-\Theta N}{k} + \frac{\Theta N}{k} = \frac{N}{k}$.*

Proof. It is possible to create a feasible solution for I' where each bin (except for possibly two bins) has k first phase items. This solution has $\lceil \frac{N-\Theta N}{k} \rceil$ bins, each with one second phase item and k small first phase items, and $\lceil \frac{\Theta N}{k} \rceil$ bins with k large first phase items (for each one of these two bin types, the last bin may have a smaller number of first phase items). Indeed the last solution is an optimal solution (since second phase items cannot be packed with large first phase items), though we will only use that it is a feasible solution. We conclude that $OPT(I') - 2 \leq \frac{N-\Theta N}{k} + \frac{\Theta N}{k} = \frac{N}{k}$. The algorithm uses a different new bin for each second phase item, since every such item has a second component larger than $\frac{1}{2}$, and every bin with first phase items has a total size above α in its second component. Thus, we get $ALG(I') = \Theta N + \lceil \frac{N-\Theta N}{k} \rceil \geq \Theta N + \frac{N-\Theta N}{k} = \frac{N+(k-1)\Theta N}{k}$. ◀

Let b be an integer such that $b < \frac{k-4}{2}$. For any integer a such that $1 \leq a \leq b$, there will be two possible inputs I_a^1 and I_a^2 . All inputs start with the first phase items defined above. The second phase of items is identical for the two inputs I_a^1 and I_a^2 (but it is different for different values of a). Let $\Gamma_a = \lceil \frac{N-\Theta N}{k-2a} \rceil$. Intuitively, when considering an optimal packing of the small first phase items in I_a^1 and I_a^2 , most of the bins will contain $k-2a$ small first phase items, and thus Γ_a is approximately their number. The second phase items are constructed

using SMALL with the same value of k as for the first phase items as follows. The number of items is $N_a = \Gamma_a$ (and once again C_3 never happens and all items are presented). The sizes are built using $\varepsilon = 1$, and the conditions C_1 and C_2 are as follows. We let C_2 be the condition that the item is packed into a bin that does not have a second phase item, and C_1 is the condition that the item is packed into a bin that already has a second phase item. The first component of each item is $\frac{a}{k}$. Given the i th output of SMALL denoted by z , for the i th item, the second component is defined as $\frac{1}{3} + z$. If z is defined when C_2 holds, we say that the item whose vector is $(\frac{a}{k}, \frac{1}{3} + z)$ is large, and otherwise it is small. Since $0 < z \leq \frac{1}{k^4}$ for any item, the items satisfy that their second components are strictly larger than $\frac{1}{3}$, and they are not larger than $\frac{1}{3} + \frac{1}{k^4} < 0.3335$. Furthermore, we conclude that the difference between the smallest second component of a large second phase item and the largest second component of a small second phase item is at least k^{-2N+3} .

Obviously, since second phase items have second components above $\frac{1}{3}$, no bin can have more than two such items. Let Y_1^a and Y_2^a denote the numbers of bins with one second phase item and two second phase items, respectively (note that there may be such bin that contain first phase items and bins that do not, and both kinds are included in these two values according to their numbers of second phase items, while bins with only first phase items are not included). There are Y_2^a small second phase items and $Y_1^a + Y_2^a$ large second phase items (and $Y_1^a + 2Y_2^a = \Gamma_a$). Note that since the first component of second phase items is $\frac{a}{k}$, they could not have been packed into bins with at least $k - a + 1$ first phase items.

Input I_a^1 continues with Γ_a items, each of the form $(\frac{a}{k}, 0.6)$. Let $\frac{1}{3} + \delta'$ be the largest second component of a small second phase item (such that for any large second phase item, its second component is larger than $\frac{1}{3} + 2\delta'$), and observe that since $\delta' \geq k^{-2N+3}$, the total sum of second component of a set of at most k first phase items is at most δ' . Input I_a^2 continues with the third phase items as follows. $\lceil \frac{\Gamma_a + 2Y_2^a}{4} \rceil$ items, each of the form $(\frac{a}{k}, \frac{2}{3} - 2\delta')$, and $\lceil \frac{N}{4k} \rceil$ items, each of the form $(0, 1 - \alpha)$. Let $\Delta_c = \sum_{i=c}^k X_i$.

► **Lemma 9.** *The costs of the algorithm satisfy*

$$ALG(I_a^1) \geq \Delta_{k-a+1} + Y_2^a + \Gamma_a$$

and

$$ALG(I_a^2) \geq \Delta_{k-a+1} + Y_1^a + Y_2^a + \frac{\Gamma_a + 2Y_2^a}{4} + \frac{N}{4k}.$$

Proof. For I_a^1 , the algorithm cannot use any bin with at least $k - a + 1$ first phase items to pack any other items (as second phase and third phase items afterwards have a first component of value $\frac{a}{k}$), and the algorithm cannot pack an item of the form $(\frac{a}{k}, 0.6)$ into a bin with two second phase items. Thus, using $\Delta_c = \sum_{i=c}^k X_i$, the total number of bins of the algorithm is at least $\Delta_{k-a+1} + Y_2^a + \Gamma_a$.

For I_a^2 , the algorithm cannot use any bin with at least $k - a + 1$ first phase items to pack items whose first component is $\frac{a}{k}$, and it cannot use any bins with first phase items to pack items whose second component is $1 - \alpha$. Moreover, since every bin with second phase items has a large second phase item, the algorithm cannot pack any third phase item into a bin containing at least one second phase item (and each bin with a third phase item will contain exactly one third phase item). The only bins that can possibly be used for third phase items are those with at most $k - a$ first phase items and no other items. Thus, the number of bins is at least $\Delta_{k-a+1} + Y_1^a + Y_2^a + \frac{\Gamma_a + 2Y_2^a}{4} + \frac{N}{4k}$. ◀

We next analyze optimal solutions for I_a^1 and I_a^2 .

10:12 Online Bin Packing with Cardinality Constraints Resolved

► **Lemma 10.** *The cost of the optimal solutions for I_a^1 and I_a^2 satisfy*

$$OPT(I_1^a) \leq \Gamma_a + \lceil \frac{N\Theta}{k} \rceil$$

and

$$OPT(I_2^a) \leq \frac{N}{4k} + \frac{Y_1^a + Y_2^a}{2} + \frac{\Gamma_a + 2Y_2^a}{4} + \frac{9N}{4k^2} + 4 = \frac{N}{4k} + \frac{3\Gamma_a}{4} + \frac{9N}{4k^2} + 4 .$$

Proof. For I_a^1 consider the following feasible solution. There are Γ_a bins, each with a second phase item (whose first component is $\frac{a}{k}$ and its second component is in $(\frac{1}{3}, 0.3335)$), one item of the form $(\frac{a}{k}, 0.6)$, and $k - 2a$ first phase items where each such item has a first component of $\frac{1}{k}$ and its second component is no larger than $\frac{1}{k^4}$ (the last bin may contain a smaller number of first phase items). The first component of the sum of the vectors of these items is 1, and the second component is at most $0.3335 + 0.6 + \frac{1}{k^3} < 1$. The remaining first phase items (there are at most $N\Theta$ such items) are packed k in a bin. We find that $OPT(I_1^a) \leq \Gamma_a + \lceil \frac{N\Theta}{k} \rceil$.

For I_a^2 , there are $\lceil \frac{N}{4k} \rceil$ bins, each with one item of the form $(0, 1 - \alpha)$ and k small first phase items (recall that the number of small first phase items is larger than $\frac{N}{4} + k$), $\lceil \frac{Y_1^a + Y_2^a}{2} \rceil$ bins with at most two large second phase items and at most $k - 2a$ first phase items, $\lceil \frac{\Gamma_a + 2Y_2^a}{4} \rceil$ bins with one item of the form $(\frac{a}{k}, \frac{2}{3} - 2\delta')$, and at most one small second phase item, and at most $k - 2a$ first phase items. The remaining first phase items are packed into additional bins, such that every bin has k such items. All items are packed since the number of small second phase items, Y_2^a , is no larger than $\frac{\Gamma_a}{2}$, so $\frac{\Gamma_a + 2Y_2^a}{4} \geq Y_2^a$. The total space for first phase items in the first three kinds of bins is at least

$$\begin{aligned} & \frac{N}{4} + (k - 2a) \left(\frac{Y_1^a + Y_2^a}{2} + \frac{\Gamma_a + 2Y_2^a}{4} \right) = \frac{N}{4} + \frac{k - 2a}{4} \cdot 3\Gamma_a \\ & \geq \frac{N}{4} + \frac{3}{4}N(1 - \Theta) = N - \frac{3}{4}N\Theta , \end{aligned}$$

so the number of bins of the last kind is at most $\lceil \frac{3N\Theta}{4k} \rceil \leq \frac{9N}{4k^2} + 1$ since $\Theta \leq \frac{3}{k}$. We find that

$$OPT(I_2^a) \leq \frac{N}{4k} + \frac{Y_1^a + Y_2^a}{2} + \frac{\Gamma_a + 2Y_2^a}{4} + \frac{9N}{4k^2} + 4 = \frac{N}{4k} + \frac{3\Gamma_a}{4} + \frac{9N}{4k^2} + 4 . \quad \blacktriangleleft$$

We get

$$\begin{aligned} R & \geq \frac{ALG(I_1^a)}{OPT(I_1^a) - 2} \geq \frac{\Delta_{k-a+1} + Y_2^a + \frac{N-N\Theta}{k-2a}}{\frac{N-N\Theta}{k-2a} + \frac{N\Theta}{k}} , \\ R & \geq \frac{ALG(I_2^a)}{OPT(I_2^a) - 5} \geq \frac{\Delta_{k-a+1} + Y_1^a + Y_2^a + \frac{N-N\Theta}{k-2a} + 2Y_2^a}{\frac{N}{4k} + \frac{3\frac{N-N\Theta}{k-2a}}{4} + \frac{9N}{4k^2}} + \frac{N}{4k} . \end{aligned}$$

We let $\beta = b/k$ and let k grows to infinity. Choosing $\beta \approx 0.192806$ and using the inequalities we showed, we find $R \geq 2.03731129$, and thus we conclude the following theorem.

► **Theorem 11.** *The asymptotic competitive ratio of any online algorithm for vector packing with $d \geq 2$ is at least 2.03731129.*

References

- 1 Y. Azar, I.R. Cohen, A. Fiat, and A. Roytman. Packing small vectors. In *Proc. of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms, (SODA'16)*, pages 1511–1525, 2016.
- 2 Y. Azar, I.R. Cohen, S. Kamara, and F.B. Shepherd. Tight bounds for online vector bin packing. In *Proc. of the 45th ACM Symposium on Theory of Computing (STOC'13)*, pages 961–970, 2013.
- 3 Y. Azar, I.R. Cohen, and A. Roytman. Online lower bounds via duality. In *Proc. of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms, (SODA'17)*, pages 1038–1050, 2017.
- 4 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.
- 5 J. Balogh, J. Békési, and G. Galambos. New lower bounds for certain bin packing algorithms. *Theoretical Computer Science*, 1:1–13, 2012.
- 6 J. Békési, Gy. Dósa, and L. Epstein. Bounds for online bin packing with cardinality constraints. *Information and Computation*, 249:190–204, 2016.
- 7 David Blitz. Lower bounds on the asymptotic worst-case ratios of on-line bin packing algorithms. Technical Report 114682, University of Rotterdam, 1996. M.Sc. thesis.
- 8 David Blitz, Andre van Vliet, and Gerhard J. Woeginger. Lower bounds on the asymptotic worst-case ratio of online bin packing algorithms. Unpublished manuscript, 1996.
- 9 A. Caprara, H. Kellerer, and U. Pferschy. Approximation schemes for ordered vector packing problems. *Naval Research Logistics*, 92:58–69, 2003.
- 10 L. Epstein. Online bin packing with cardinality constraints. *SIAM Journal on Discrete Mathematics*, 20(4):1015–1030, 2006.
- 11 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.
- 12 H. Fujiwara and K.M. Kobayashi. Improved lower bounds for the online bin packing problem with cardinality constraints. *Journal of Combinatorial Optimization*, 29(1):67–87, 2015.
- 13 G. Galambos, H. Kellerer, and G.J. Woeginger. A lower bound for online vector packing algorithms. *Acta Cybernetica*, 10:23–34, 1994.
- 14 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.
- 15 K. Jansen, M. Maack, and M. Rau. Approximation schemes for machine scheduling with resource (in-)dependent processing times. In *Proc. of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms, (SODA'16)*, pages 1526–1542, 2016.
- 16 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.
- 17 H. Kellerer and U. Pferschy. Cardinality constrained bin-packing problems. *Annals of Operations Research*, 92:335–348, 1999.
- 18 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.
- 19 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.

10:14 Online Bin Packing with Cardinality Constraints Resolved

- 20 C. C. Lee and D. T. Lee. A simple online bin packing algorithm. *Journal of the ACM*, 32(3):562–572, 1985.
- 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 A. van Vliet. An improved lower bound for online bin packing algorithms. *Information Processing Letters*, 43(5):277–284, 1992.
- 24 A. C. C. Yao. New algorithms for bin packing. *Journal of the ACM*, 27:207–227, 1980.