More on batched bin packing

Leah Epstein*

Abstract

Bin packing is the problem of partitioning a set of items into subsets of total sizes at most 1. In batched bin packing, items are presented in k batches, such that the items of a batch are presented as a set, to be packed before the next batch. In the disjunctive model, a algorithm must use separate bins for the different batches. We analyze the asymptotic and absolute approximation ratios for this last model completely, and show tight bounds as a function of k.

keywords: Bin packing; approximation ratio; batched problems.

1 Introduction

We study the bin packing problem [23, 15], and analyze algorithms for it that receive the input in a small number of batches. In the bin packing problem, the goal is to allocate input items of sizes in $(0, 1]$ to blocks called bins, such that the total size of items assigned to each block does not exceed 1, and the number of non-empty bins is minimized. The process of allocation of items to bins is also referred to as the process of packing items into bins. The items are are denoted by $1, 2, \ldots, n$, and the size of item i is denoted by s_i.

In the offline variant, all input items are presented together as a set. The problem is NP-hard in the strong sense, and thus, approximation algorithms were studied. An approximation algorithm has an asymptotic approximation ratio of at most R, if there exists a constant $C_1 \geq 0$ (which is independent of the input), such that for any input I, the cost of the algorithm for I does not exceed the following value: R times the optimal cost for this input plus C_1. If $C_1 = 0$, then the approximation ratio is called absolute. A specific optimal algorithm as well as its cost are denoted by $OPT(I)$ or OPT, when the input is fixed. An alternative definition of the asymptotic approximation ratio is the supreme limit of the ratio between the cost of the algorithm and OPT, as a function of this last cost, taking the maximum or supremum over the inputs with the same optimal cost.

For this classic variant, an asymptotic fully polynomial time approximation scheme (AFPTAS) is known [11, 18]. This is a class of algorithms containing an approximation algorithm with an asymptotic approximation ratio of $1 + \varepsilon$ for any $\varepsilon > 0$, with running time polynomial in the size of the input and in $\frac{1}{\varepsilon}$. Many fast heuristics are known [23, 17, 16], including First-Fit (FF) [23, 17], which processes a list of items, and assigns each item, in turn, to the minimum index bin where it can be added. First-Fit-Decreasing (FFD) [15] acts identically to FF, but it requires the list of input items to be sorted according to non-increasing sizes. FFD is known to have the best possible

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

absolute approximation ratio 1.5 (this is the best possible unless P=NP) [22]. Next-Fit (NF) [16] assigns each item into the maximum index non-empty bin if it can be packed there, and otherwise to the minimum index empty bin. Next-Fit-Decreasing (NFD) [1] and Next-Fit-Increasing (NFI) [12] are the algorithms that apply NF to lists that are sorted by non-increasing and non-decreasing orders, respectively. First-Fit-Increasing (FFI) is identical to NFI. The asymptotic and absolute approximation ratio of FF and NF are 1.7 and 2, respectively [17, 8, 16]. The sorted versions have a better performance, FFD has an asymptotic approximation ratio of 11/9 [15], while NFD and NFI have an asymptotic approximation ratio of approximately 1.69103 [1, 12].

In the online scenario, an algorithm is presented with the items one by one, and each item must be packed before the next item can be seen. In this variant, the best possible asymptotic approximation ratio (also called competitive ratio, for online algorithms, as an online algorithm is compared to an optimal offline algorithm) is at least 1.5403 [4] and at most 1.58889 [21], and the best possible absolute approximation ratio is $\frac{5}{3}$ [27, 3]. The algorithms FF and NF are online algorithms, but FFD, NFD, and NFI, are not online algorithms.

In batched bin packing, items are presented in k batches, for an integer $k \geq 1$. For each batch, the algorithm receives all its items at once, and these items are to be packed irrevocably before the next batch is presented (if the current batch is not the last one). This last model is an intermediate model, which bridges between the two extreme known models. The case $k = 1$ corresponds to the offline problem. If the number of batches that may be presented is unbounded, this scenario corresponds to the online problem. There are two models for any fixed $k \geq 2$. In the disjunctive model [7], the algorithm must use separate bins for the different batches. In the augmenting model [14], the algorithm may use existing bins, where items were already packed in previous batches, as well as new bins. Obviously, any algorithm for the disjunctive model can be seen as an algorithm for the augmenting model, with the same performance.

In this work, we analyze the asymptotic approximation ratio and the absolute asymptotic approximation ratio for the disjunctive model completely, and show tight bounds for them as a function of k. The asymptotic approximation ratio tends to approximately 1.69103, while the tight absolute approximation ratio is exactly k. Moreover, our results provide an improved upper bound on the asymptotic approximation ratio for the augmenting model with two batches. This last algorithm has an asymptotic approximation ratio of 1.5, improving over the algorithm of Dósa [7] (see below). For the analysis, we will define subset of items called *combined items*, and use them for the analysis of an optimal solution rather than dealing with the actual items. Moreover, we analyze a particular offline algorithm for the combined items, rather than analyzing an optimal solution. The other features of our analysis are related to those used in [1, 19, 25, 13, 10, 7]. Finally, we show that the absolute approximation ratio for the augmenting model with two batches is $\frac{3}{2}$, while for at least three batches, it is exactly $\frac{5}{3}$.

There is an additional relation between batched bin packing and the online problem. All lower bounds on the asymptotic competitive ratio [26, 24, 4] are of the form where a pre-determined number batches of identical items are presented to the algorithm, such that the algorithm does not know how many non-empty batches will arrive. These lower bounds are valid for both models of batched bin packing, with the corresponding number of batches (the number of batches in the lower bound construction). Moreover, there is a certain relation of the disjunctive model to bounded space online bin packing. In the latter problem, an online algorithm may keep a constant number of bins open, and it must close all other bins that were used, in the sense that they can no longer be used for packing new items. The lower bounds for online bounded space bin packing are of the form where batches of identical items arrive, and the algorithm must pack almost all items (except for a constant number of items) of a batch into new bins, as there is only a constant number of open bins that were used before. The best possible asymptotic competitive ratio for online bounded space bin packing is the sum of a series and tends to approximately 1.69103 [19, 25]. This is the same value as the asymptotic approximation of NFD and NFI, and the series will be discussed in what follows.

In [14], Gutin, Jensen, and Yeo, proved a lower bound of approximately 1.3871 on the asymptotic approximation ratio for the augmenting model with two batches. In [7], Dósa analyzed FFD for two batches and both models of batched bin packing. The algorithm applies FFD on each batch independently, using separate bins. The asymptotic approximation ratio was shown to be $\frac{19}{12} \approx 1.5833$ even for the disjunctive model. Moreover, in the same paper it is shown that the asymptotic approximation ratio of any algorithm for the disjunctive model (and $k = 2$) is at least $\frac{3}{2}$. In the case $k \geq 3$, there are online algorithms that perform better than the best algorithms for the disjunctive model (which are analyzed here), and thus better algorithms for the augmenting model cannot be those of the disjunctive model. Balogh et al. [2] proved lower bounds on the asymptotic competitive ratios of algorithms for the augmenting model and different numbers of batches, and in particular, they showed a lower bound of 1.51211 for three batches (while the previously known lower bound was 1.5 [26]). For four batches, the lower bound of van Vliet [24] is approximately 1.539, thus, the effect of a small number of batches is mostly noticeable for $k = 2$ and $k = 3$.

2 Main result

We will prove our results in this section. We start with the required definitions, then we prove the upper bounds, and finally we will that these bounds cannot be improved for the disjunctive model.

Preliminaries. We define a sequence π_j (for any integer $j \geq 1$) as follows. Let $\pi_1 = 1$, and for $i \geq 1$, $\pi_{i+1} = \pi_i(\pi_i + 1)$. Let $\Gamma_k = \sum_{i=1}^{k} \pi_i$. We have $\Gamma_1 = 1$, $\Gamma_2 = \frac{3}{2}$, $\Gamma_3 = \frac{5}{3}$, $\Gamma_4 = \frac{71}{42} \approx 1.690476$, and $\Gamma_\infty = \lim_{k\to\infty} \Gamma_k \approx 1.69103$. Recall that the last value is the asymptotic approximation ratio of several well-known bin packing algorithms (for example, it is the asymptotic approximation ratio of Harmonic, NFI, and NFD). This sequence is frequently used in analysis of bin packing [1, 19, 25, 13, 10]. In particular, the following claim is often used, and we will use it in our analysis as well.

Claim 1 *For any integer $i \geq 2$, $\pi_i > \pi_{i-1}$, and $\pi_i \geq i$. For any two integers $i \geq 1$ and $i' \geq i$, $\pi_{i'}$ is divisible by π_i. Moreover, we have $\sum_{i=1}^{j} \frac{1}{\pi_i+1} = 1 - \frac{1}{\pi_{j+1}}$ for $j \geq 1$.*

Proof. The claims are proved by induction. In the base case $i = 2$, $\pi_2 = 2$. Assume that $\pi_i > \pi_{i-1}$ and $\pi_i \geq i$ hold. Then, $\pi_{i+1} = \pi_i(\pi_i + 1)$. Using $\pi_i \geq i$, we get $\pi_i + 1 > i \geq 2$, so $\pi_{i+1} > \pi_i$. Using $\pi_i \geq i \geq 2$, we get $\pi_i(\pi_i + 1) \geq 2(i + 1) > i + 1$.

The base case of the second claim ($i' = i$) is trivial, and the induction is simple as well: Assume that $\pi_{i'-1}$ is divisible by π_i. This implies that $\pi_{i'}$ is divisible by π_i as $\pi_{i'}$ is divisible by $\pi_{i'-1}$.

Consider the third claim. In the base case $j = 1$, and $\frac{1}{\pi_1+1} = \frac{1}{2} = 1 - \frac{1}{2} = 1 - \frac{1}{\pi_2}$. Next, assume that $\sum_{i=1}^{j} \frac{1}{\pi_i+1} = 1 - \frac{1}{\pi_{j+1}}$ holds. We show that $\sum_{i=1}^{j+1} \frac{1}{\pi_i+1} = 1 - \frac{1}{\pi_{j+2}}$ holds as well. Indeed, $\sum_{i=1}^{j+1} \frac{1}{\pi_i+1} = \sum_{i=1}^{j} \frac{1}{\pi_i+1} + \frac{1}{\pi_{j+1}+1} = 1 - \frac{1}{\pi_{j+1}} + \frac{1}{\pi_{j+1}+1} = 1 - \frac{1}{\pi_{j+1}(\pi_{j+1}+1)} = 1 - \frac{1}{\pi_{j+2}}$. ■

In what follows, we will prove the following theorem.

Theorem 2 *The absolute approximation ratio for batched bin packing in the disjunctive model is k, and the asymptotic approximation ratio is Γ_k. The additive constant for the last approximation ratio is $\Theta(k)$, and for $k = 2$ it is exactly $\frac{1}{2}$.*

Upper bounds. We will analyze a specific input I. Let OPT_i denote the optimal cost for packing the items of batch i, let ALG_i denote the cost of a given algorithm ALG for batch i, and $ALG = \sum_{i=1}^{k} ALG_i$. Obviously, $OPT_i \leq OPT$.

Consider the following algorithm. The algorithm FF-Batch (FFB) applies FF on each batch separately.

Proposition 3 *For any $k \geq 2$, the absolute approximation ratio of FFB is at most k.*

Proof. If the input is empty, we are done. If FFB uses a single bin for every batch, $FFB_i = 1$, and $FFB = k$. In the latter case, as $OPT \geq 1$, the absolute approximation ratio does not exceed k.

Finally, assume that at least two bins were used by FFB for at least one batch. Let θ be such that $1 \leq \theta \leq k$ is the number of batches for which there are at least two bins, and let ℓ denote the total number of bins in such batches. For inputs where FF creates at least two bins, the total size of items packed into each bin is above $\frac{1}{2}$ on average (as any item of a bin could not be packed into a bin of a smaller index, see e.g. [5], page 1918). Thus, the total size of items is above $\frac{\ell}{2}$, and $OPT > \frac{\ell}{2}$, and therefore, $OPT \geq \lceil \frac{\ell+1}{2} \rceil \geq 2$, and $OPT \geq \frac{\ell+1}{2}$.

If $\theta = k$, then $FFB \leq \ell < 2 \cdot OPT$. Otherwise, there are $k - \theta \geq 1$ bins for other batches, and we find $FFB \leq (k - \theta) + \ell \leq k - \theta + 2 \cdot OPT - 1 \leq (k - \theta - 1)/2 \cdot OPT + 2 \cdot OPT = (k - \theta + 3)/2 \cdot OPT \leq k \cdot OPT$, as $k + \theta \geq 3$, which holds as $k \geq 2$ and $\theta \geq 1$. ■

Consider an an algorithm ALG that runs an approximation algorithm with approximation ratio at most $\phi \geq 1$ for each batch (this last approximation algorithm does not have to be an online algorithm, as it is applied on one batch at a time). The absolute approximation ratio of ALG is at most $\phi \cdot k$, since $ALG = \sum_{i=1}^{k} ALG_i \leq \sum_{i=1}^{k} \phi \cdot OPT_i \leq \sum_{i=1}^{k} \phi \cdot OPT = k\phi \cdot OPT$. If we are interested in the effect of splitting the input into batches rather than the effect of limited computational power and we assume that the items of each batch are packed optimally (that is, $\phi = 1$), the absolute approximation ratio here is k as well. We will analyze the last algorithm with respect to the asymptotic approximation ratio.

For the analysis of the asymptotic approximation ratio, we will show $\sum_{i=1}^{k} OPT_i \leq \Gamma_k \cdot OPT + O(k)$, and for $k = 2$, we will show $OPT_1 + OPT_2 \leq \frac{3}{2} \cdot OPT + \frac{1}{2}$. This will prove that the asymptotic approximation ratio for k batches is Γ_k. If instead of ALG, an approximation algorithm is used for each batch, where that algorithm computes for each input I' a solution of cost at most $\psi \cdot OPT(I') + C_1$ (where C_1 is a non-negative constant), then the total cost of the solution will be $\sum_{i=1}^{k}(\psi \cdot OPT_i + C) \leq \psi\Gamma_k \cdot OPT + O(k)$. As an approximation scheme for bin packing can be applied (an APTAS or an AFPTAS), the approximation ratio increases only by a multiplicative factor of $1+\varepsilon$ (where $\varepsilon > 0$ can be chosen to be arbitrarily small). Thus, in what follows, we will analyze the algorithm ALG that computes an optimal solution for each batch.

Given a fixed optimal solution OPT for the entire input, we create combined-items (C-items) as follows. For a bin B of the optimal solution, let B^j denote the items of B that belong to batch j (for $1 \leq j \leq k$). A C-item a of size $\sigma_a = \sum_{i \in B^j} s_i$ is created for any j such that $1 \leq j \leq k$. The C-item that was created for a given value of j is called a C-item of batch j. The size of a C-item is non-negative (it can be zero if B^j is empty), and it cannot exceed 1, as $\sum_{i \in B^j} s_i \leq \sum_{i \in B} s_i \leq 1$. Any solution (i.e., packing) for the C-items induces a solution for the original input. Moreover, a solution where every bin contains only C-items of one batch induces solutions for the k inputs of the k batches (into separate bins for the different batches).

The analysis will consist of bounding the cost of an optimal solutions for the C-items from above. Specifically, we will present a solution SOL_i (whose cost is also denoted by SOL_i) for the C-items of batch i, and we will find an upper bound on $SOL = \sum_{i=1}^{k} SOL_i$. This will provide an upper bound for $\sum_{i=1}^{k} OPT_i$, as $OPT_i \leq SOL_i$.

We start with a simple analysis of the case $k = 2$. In this case, every bin of OPT has two C-items. In order to construct these solutions (SOL_1 and SOL_2), we consider the packing of OPT, and split bins of OPT into pairs (possibly leaving one unpaired bin). We repack the C-items such that every bin will contain only C-items of one batch as follows. Given two bins of OPT that are a pair, let the sizes of their C-items be α_1, α_2, β_1, β_2 (where the index of the size of an item corresponds to its batch). Since these four items are packed into two bins in OPT, $\alpha_1 + \alpha_2 + \beta_1 + \beta_2 \leq 2$. Thus, at least one of $\alpha_1 + \beta_1 \leq 1$ and $\alpha_2 + \beta_2 \leq 1$ must hold. In the former case, create one bin in SOL_1 with the two C-items of the first batch, and two dedicated bins for the other items in SOL_2. In the latter case, create one bin for α_2 and β_2 in SOL_2, and one bin for each of α_1 and β_1 in SOL_1. In both cases, at most three bins were created. If an unpaired bin remains, create two bins (one for each C-item, such that one bin is in SOL_1 and the other is in SOL_2).

Proposition 4 *We have* $SOL \leq \frac{3}{2} \cdot OPT + \frac{1}{2}$.

Proof. As at most three bins of SOL_1 and SOL_2 (together) are created from every pair of bins of OPT, and two bins are created from an unpaired bin, if it exists, we find the following. We have $SOL \leq \frac{3}{2} OPT$, if OPT is even, and otherwise $SOL \leq \frac{3}{2} \cdot \frac{OPT-1}{2} + 1 = \frac{3}{2} OPT + \frac{1}{2}$. ■

Next, we consider a set of solutions SOL_i ($i = 1, 2, \ldots, k$), where SOL_i is constructed by running Next Fit Decreasing (NFD) on the C-items of batch i. Recall that this algorithm sorts the C-items in a non-increasing order (by size), and assigns each item, in turn, into a bin of the maximum index that is non-empty, and into an empty bin of a minimum index if this is impossible (in the case where the total packed size would exceed 1). We assume that all zero size items are packed into the last non-empty bin. Note that this algorithm cannot be simply applied on the input, as the identity of the C-items is not known to the algorithm.

While a weight-based analysis became fairly common for bin packing problems, here we will apply it on C-items rather than on the actual items. We will find a lower bound on the total weight of C-items in each solution SOL_i compared to the number of bins, and we will find an upper bound on the total weight of C-items packed into a bin of OPT.

Several weight functions were defined for bounded space online algorithms, and for offline algorithms that have the same asymptotic approximation ratio Γ_∞ [1, 19, 25, 13, 10]. Here, we adopt the weights given in [1]. We define a function $f:[0,1]\to[0,1]$ as follows. We let $f(0)=0$. For $x\in(0,1]$, $f(x)$ is defined as follows, let $j_x\geq 1$ be an integer such that $x\in(\frac{1}{j_x+1},\frac{1}{j_x}]$. If $j_x=\pi_i$ for some $i\geq 1$, then we let $f(x)=\frac{1}{\pi_i}$. Otherwise, we let $f(x)=\frac{j_x+1}{j_x}\cdot x$. Any $x>0$ satisfies $f(x)\leq\frac{j_x+1}{j_x}\cdot x$, since in the cases where this does not hold with equality, $x\in(\frac{1}{\pi_i+1},\frac{1}{\pi_i}]$ for some $i\geq 1$, and $\frac{f(x)}{x}=\frac{1}{\pi_i}\cdot\frac{1}{x}<\frac{1}{\pi_i}\cdot(\pi_i+1)=\frac{j_x+1}{j_x}$. Moreover, this implies that if $y\leq\frac{1}{j}$ for an integer $j\geq 1$, then $f(y)\leq\frac{j+1}{j}\cdot y$.

For a C-item a of size σ_a, we let $w_a=f(\sigma_a)$. Let W_i denote the total weight of all C-items of batch i, and let $W=\sum_{i=1}^{k}W_i$.

The next lemma is proved in [1] (see Claim 1 in [1]).

Lemma 5 *Given a set of items of positive sizes, whose weights are defined according to f, if applying NFD on these items results in X bins, then the total weight of the items is at least $X-3$.*

Adding zero size items changes neither the total weight nor the number of bins, if the inputs contains at least one item having a positive size. If all items have zero sizes, then the total weight and the number of bins are both equal to zero. We find the next corollary.

Corollary 6 *Given a set of items of non-negative sizes, whose weights are defined according to f, if applying NFD on these items results in X bins, then the total weight of the items is at least $X-3$.*

The next lemma provides an upper bound for the weights of a bin containing k C-items. It resembles proofs given in [19, 25, 13, 6, 20], but the number of C-items is at most k, similarly to [9].

Lemma 7 *Consider k values $x_1,x_2,\ldots,x_k$ such that for $i=1,\ldots,k$, we have $0\leq x_i\leq 1$, and $\sum_{i=1}^{k}x_i\leq 1$. For these values, $\sum_{i=1}^{k}f(x_i)\leq\Gamma_k$.*

Proof. Without loss of generality, assume $x_1\leq x_2\leq\cdots\leq x_k$. Let $1\leq i\leq k$ be the minimum integer such that $x_i\notin(\frac{1}{\pi_i+1},\frac{1}{\pi_i}]$. If i is undefined, then for $1\leq i\leq k$, $f(x_i)=\frac{1}{\pi_i}$, and $\sum_{i=1}^{k}f(x_i)=\Gamma_k$. Otherwise, as $x_\ell\in(\frac{1}{\pi_\ell+1},\frac{1}{\pi_\ell}]$ for $1\leq\ell<i$, we have $\sum_{j=i}^{k}x_j\leq 1-\sum_{\ell=1}^{i-1}x_\ell<1-\sum_{\ell=1}^{i-1}\frac{1}{\pi_\ell+1}=\frac{1}{\pi_i}$, by Claim 1. Since $x_i\notin(\frac{1}{\pi_i+1},\frac{1}{\pi_i}]$, we have $x_i\leq\frac{1}{\pi_i+1}$, and moreover, as $x_j\leq x_i$ for $j>i$, we have $x_j\leq\frac{1}{\pi_i+1}$ for $j\geq i$. Recall that if $y\leq\frac{1}{t}$, $f(y)\leq\frac{t+1}{t}\cdot y$, and we find $\sum_{j=i}^{k}f(x_j)\leq\frac{\pi_i+2}{\pi_i+1}\cdot\sum_{j=i}^{k}x_j=\frac{\pi_i+2}{\pi_i+1}\cdot\frac{1}{\pi_i}=\frac{1}{\pi_i}+\frac{1}{\pi_i(\pi_i+1)}=\frac{1}{\pi_i}+\frac{1}{\pi_{i+1}}$. This shows $\sum_{j=i}^{k}f(x_j)\leq\sum_{j=1}^{i+1}\frac{1}{\pi_j}=\Gamma_{i+1}$, and proves the claim for the cases where $i<k$. If $i=k$, then $f(x_i)\leq\frac{\pi_i+2}{\pi_i+1}\cdot\frac{1}{\pi_i+1}\leq\frac{1}{\pi_i}$, which follows from simple algebra. In this case $\sum_{j=i}^{k}f(x_j)\leq\sum_{j=1}^{i}\frac{1}{\pi_j}=\Gamma_i=\Gamma_k$. ■

Since any bin of OPT has k items, we find the following.

Corollary 8 *The total weight of any bin of OPT is at most Γ_k.*

Combining the bounds on total weights, we find the following.

Theorem 9 *We have $SOL\leq\Gamma_k\cdot OPT+O(k)$.*

Proof. The total weight of all items does not exceed $\Gamma_k\cdot OPT$, as any bin of OPT has total weight at most Γ_k. As $W_i\geq SOL_i-3$, we get $SOL\leq\sum_{i=1}^{k}(W_i+3)=W+3k\leq\Gamma_k\cdot OPT+O(k)$. ■

Lower bounds. The construction of the lower bound (which will be proved for all cases simultaneously) is similar to that of Lee and Lee [19], and to later constructions [25, 13, 6, 20], but we use a constant number of items, as in [9].

Proposition 10 *The absolute approximation ratio of any algorithm for batched bin packing in the disjunctive model is at least k. The asymptotic approximation ratio of any algorithm for batched bin packing in the disjunctive model is at least Γ_k, with an additive constant of $\Omega(k)$, and at least $\frac{1}{2}$ for $k = 2$.*

Proof. Let N be a positive integer such that $N-1$ is divisible by π_k (and therefore by Claim 1, it is divisible by π_i for $1 \leq i \leq k-1$ as well). Batch i consists of N items of size $\frac{1}{\pi_i+1}+\delta$, where $\delta < \frac{1}{\pi_{k+1}^2}$. An optimal solution consists of N bins, such that each bin has one item of each type. By Claim 1, the total size of items packed into each bin is $\sum_{i=1}^{k} \frac{1}{\pi_i+1}+k\delta = 1-\frac{1}{\pi_{k+1}}+k\delta \leq 1-\frac{1}{\pi_{k+1}}+\pi_{k+1}\delta < 1$, and therefore the packing of each bin is valid, and the number of bins in any solution cannot be smaller, as the N items of the first batch have sizes above $\frac{1}{2}$. As the items of batch i are strictly larger than $\frac{1}{\pi_i}$, each bin for this batch can contain at most π_i items, and an optimal solution for batch i consists of $\frac{N-1}{\pi_i}+1$ bins. Thus, the algorithm uses $k+(N-1)\Gamma_k$ bins.

We get $ALG = \Gamma_k \cdot OPT + k - \Gamma_k = \Gamma_k OPT + \Omega(k)$, since $\Gamma_k \leq 2$. For $k = 2$, We get $ALG = \frac{3}{2} \cdot OPT + \frac{1}{2}$. This shows that the asymptotic approximation ratio is at least Γ_k (using large values of N), and the absolute approximation ratio is at least k, using $N = 1$, in which case $OPT = 1$. It also shows that the additive constant for the asymptotic approximation ratio is $\Omega(k)$ and for $k = 2$ it is at least $\frac{1}{2}$. ■

The augmenting model. Finally, we briefly discuss the augmenting model. Here, the absolute approximation ratio is much smaller than that of the disjunctive model.

Proposition 11 *The best possible absolute approximation ratio for the augmenting model is equal to $\frac{3}{2}$ for $k = 2$, and to $\frac{5}{3}$ for $k \geq 3$.*

Proof. First, consider the case $k \geq 3$. The online algorithm of Balogh et al. [3] has an absolute competitive ratio of $\frac{5}{3}$. Moreover, the lower bound construction stated in the last paper consists of at most three batches.

Next, consider the case $k = 2$. The following input provides a simple lower bound. A first batch consists of two items, each of size 0.4, and it is possibly followed by a second batch consisting of two items, each of size 0.6. If the first two items are packed into two bins, the absolute approximation ratio is at least 2, and otherwise, the algorithm uses three bins, while an optimal solution requires two bins (each containing one item of each batch).

We show that the following (exponential time) algorithm has an absolute approximation ratio of at most $\frac{3}{2}$. For a given packing, let its *lower value* be the minimum total size of items in any non-empty bin. The algorithm constructs a packing of the first batch with a minimum number of bins, such that (out of solutions with the minimum number of bins) its lower value is minimal. That is, it does not only construct an optimal packing, but out of optimal solutions, the maximum empty space in any bin is maximal. In the second batch, it creates a solution with the smallest total number of bins, such that it can pack items into the already existing non-empty bins, and into new bins.

We adapt the previous approach, where SOL_1 and SOL_2 are analyzed for bounding the approximation ratio. If OPT is even, we analyze these solutions exactly as before. If OPT is odd, let γ_1 and γ_2 denote the sizes of the C-items of the first batch and the second batch, respectively, for the unpaired bin of OPT.

The solution that the algorithm constructs for the first batch has at most SOL_1 bins. One valid option for the second batch would be to pack all items of the second batch into SOL_2 new bins. Thus, if OPT is even, then the cost of the constructed solution is at most $SOL_1 + SOL_2 \leq \frac{3}{2}OPT$. We are left with the case where OPT is odd. If the cost of the solution that is constructed for the first batch is strictly smaller than SOL_1, then the cost is at most $(SOL_1 - 1) + SOL_2 \leq \frac{3}{2}OPT - \frac{1}{2}$. Therefore, we are left with the case that the minimum number of bins required for the first batch is exactly SOL_1 (it cannot be larger, as SOL_1 is a possible packing of this input). Since SOL_1 is a possible solution with the smallest number of bins for the first batch, by the choice of a solution with the smallest lower value, we find that the lower value of the output for the first batch is at most γ_1. A possible output for the second batch would be to create all bins of SOL_2, except for the bin of the C-item of size γ_2, and pack the last item (that is, the items it consists of) into a bin of a smallest lower value constructed by the algorithm for the items of the first batch. This is possible as $\gamma_1 + \gamma_2 \leq 1$. We find that a possible output of the algorithm is a solution of cost $SOL_1 + SOL_2 - 1 < \frac{3}{2}OPT$. ■

References

[1] B. S. Baker and E. G. Coffman, Jr. A tight asymptotic bound for next-fit-decreasing bin-packing. *SIAM J. on Algebraic and Discrete Methods*, 2(2):147–152, 1981.

[2] J. Balogh, J. Békési, Gy. Dósa, G. Galambos, and Z. Tan. Lower bound for 3-batched bin packing. Manuscript, 2015.

[3] 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.

[4] J. Balogh, J. Békési, and G. Galambos. New lower bounds for certain bin packing algorithms. *Theoretical Computer Science*, 1:1–13, 2012.

[5] 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.

[6] J. Csirik and G. J. Woeginger. Resource augmentation for online bounded space bin packing. *Journal of Algorithms*, 44(2):308–320, 2002.

[7] Gy. Dósa. Batched bin packing revisited. *Journal of Scheduling*, 2015. To appear.

[8] 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.

[9] L. Epstein. Online bin packing with cardinality constraints. *SIAM J. Discrete Math.*, 20(4):1015–1030, 2006.

[10] L. Epstein and A. Levin. On bin packing with conflicts. *SIAM Journal on Optimization*, 19(3):1270–1298, 2008.

[11] W. Fernandez de la Vega and G. S. Lueker. Bin packing can be solved within $1 + \varepsilon$ in linear time. *Combinatorica*, 1(4):349–355, 1981.

[12] D. C. Fisher. Next-fit packs a list and its reverse into the same number of bins. *Operations Research Letters*, 7(6):291–293, 1988.

[13] G. Galambos and G. J. Woeginger. Repacking helps in bounded space online bin packing. *Computing*, 49:329–338, 1993.

[14] G. Gutin, T. Jensen, and A. Yeo. Batched bin packing. *Discrete Optimization*, 2(1):71–82, 2005.

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

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

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

[18] N. Karmarkar and R. M. Karp. An efficient approximation scheme for the one-dimensional bin-packing problem. In *Proc. of the 23rd Annual Symposium on Foundations of Computer Science (FOCS'82)*, pages 312–320, 1982.

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

[20] S. S. Seiden. An optimal online algorithm for bounded space variable-sized bin packing. *SIAM Journal on Discrete Mathematics*, 14(4):458–470, 2001.

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

[22] D. Simchi-Levi. New worst-case results for the bin-packing problem. *Naval Research Logistics*, 41(4):579–585, 1994.

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

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

[25] G. J. Woeginger. Improved space for bounded-space online bin packing. *SIAM Journal on Discrete Mathematics*, 6:575–581, 1993.

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

[27] G. Zhang. Private communication.