Lower bounds for batched bin packing

János Balogh [*] József Békési [†] György Dósa[‡] Leah Epstein[§] Asaf Levin[¶]

Abstract

We consider batched bin packing. Items are presented in a constant number of batches, and each batch should be packed before the next batch is presented. The cases of two, three, and four batches are studied. We prove improved lower bounds for the standard and parametric variants in some of the cases, and shorten the proofs for all other cases. To achieve this, we apply a new technique in our analysis, which differs from the ones previously used for proving such results.

1 Introduction

Batched bin packing is an intermediate model between online bin packing and offline bin packing. In bin packing problems, items of sizes in $(0,1]$ are given, and the goal is to partition them into the minimum number of sets called bins, under the condition that no bin will have a total size of items above 1. The process of partitioning is also called packing or assigning. If item sizes are limited to an interval $(0,\frac{1}{r}]$ for a positive integer $r \geq 1$, we refer to the problem as parametric (with parameter r). The parametric case where we have $r=1$ is just the standard problem.

In online bin packing, items are presented one by one, and the assignment to bins is performed such that every item is packed before the next item is presented. In offline bin packing, the entire set of items is presented at once. Batched bin packing is the variant of the problem where a positive integer parameter k is given, and items are presented in k batches. When a new batch is presented, its items have to be packed (into existing bins and possibly new bins) before the next batch is seen by the algorithm.

For a bin packing algorithm A and an input I, let $A(I)$ be the cost (number of bins) used by A for I. The bin packing algorithm A can be an online or offline algorithm or an algorithm for batched bin packing, and it can also be an optimal offline algorithm OPT. The absolute competitive ratio of algorithm A for input I is the ratio between $A(I)$ and $OPT(I)$. The absolute competitive ratio of A is the worst-case (or supremum) absolute competitive ratio over all inputs. Given an integer M, we can consider the worst-case absolute competitive ratio over inputs where $OPT(I)$ is not smaller than M. Considering this sequence of ratios and letting M grow to infinity, the limit is the

[*]Institute of Informatics, University of Szeged, Szeged, Hungary. `baloghj@inf.u-szeged.hu`.

[†]Department of Applied Informatics, Gyula Juhász Faculty of Education, University of Szeged, Hungary. `bekesi@jgypk.u-szeged.hu`. Supported by the EU-funded Hungarian grant EFOP-3.6.2-16-2017-00015 and by National Research, Development and Innovation Office NKFIH under the grant SNN 129178.

[‡]Department of Mathematics, University of Pannonia, Veszprém, Hungary. `dosagy@almos.vein.hu`.

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

[¶]Faculty of Industrial Engineering and Management, The Technion, Haifa, Israel. `levinas@ie.technion.ac.il`.Partially supported by grant number 308/18 of ISF - Israeli Science Foundation.

asymptotic competitive ratio of A. This measure is the standard one for analysis of the bin packing problem, and it is considered to be more meaningful than the absolute ratio (which is affected by very small inputs).

Here, we revisit the batched bin packing problem, and apply new methods of proving lower bounds for asymptotic competitive ratios to find new results, and to show shorter proofs for known results. We consider both the classic problem, and the parametric case with $r \geq 2$. Since batched bin packing is not an online problem, lower bound results for online algorithms are not valid for it. On the other hand, if one considers upper bounds on the asymptotic competitive ratio, obviously all results known for online bin packing are valid for any number of batches k, since the items of each batch can simply be packed online. The current best result by Balogh et al. [1] has an asymptotic competitive ratio of at most 1.57829. We discuss other known results for batched bin packing below.

Even though generally one cannot borrow lower bounds for the online problem and use them as lower bounds for batched bin packing, still many of the earlier lower bounds on asymptotic competitive ratios for batched bin packing follow from lower bounds for online bin packing, since those lower bounds were proved using batches of items. For example, the first lower bound for two batches and items of arbitrary sizes is by Liang [15], and its value is $\frac{4}{3}$. The current best result for this case is by Gutin, Jensen, Yeo [13] and its value is 1.3871356. Four batches were considered by Brown [7]. The lower bound constructions of van Vliet [16], Galambos [12], and [5] imply lower bounds for batched bin packing for the standard case and the parametric case.

The current best lower bound for the asymptotic competitive ratio [3] (whose value is 1.54278) for standard online bin packing does not imply any results for batched bin packing as it has a part that is constructed in a fully adaptive way, such that every item depends on the action of the algorithm. In fact, the reason for the implied results for batched bin packing is that most constructions were based on batches with identical items. Moreover, the number of batches required to obtain a lower bound are small, and the effect of adding a very large number of batches is minor. Note that batches may be empty, and therefore any upper bound on the asymptotic competitive ratio for $k+1$ batches is valid for k batches, and any lower bound on the asymptotic competitive ratio for k batches is valid also for $k+1$ batches.

While batched bin packing is a natural problem, there are just a few articles studying it. Gutin, Jensen, Yeo [13] introduced the problem, and proved lower bounds. Dósa [9], analyzed the asymptotic competitive ratio for a greedy algorithm based on First Fit Decreasing and showed that it is $\frac{19}{12} \approx 1.5833$ for the case of two batches. A different algorithm was analyzed later [10], with an asymptotic competitive ratio of 1.5 for two batches and gradually increasing to approximately 1.69103 for larger numbers of batches (such that already for three batches, online algorithms perform better). A lower bound of 1.51211 for three batches was shown by Balogh et al. [4]. This result was surprising since the previous result by Yao [17] was just 1.5, and it was unclear whether the actual bound could be above 1.5. A different model where bins of one batch cannot be used for another batch was studied as well [9, 10].

In this work, we use a method for analyzing lower bounds on the asymptotic competitive ratio, resulting from certain types of constructions [6, 3, 2, 11]. Here, we do not use the most basic approach, and we explain our approach below. The reason for this is as follows. The studied problem is batched bin packing, and in the constructions, a large number of items are presented at once such that items presented in every batch will be identical. However, this common size is not known in advance, unlike many constructions for online bin packing [17, 15, 7, 16, 5] where the sizes are known in advance and the only decision of an adversary is the input stopping point, or in our terms, the point after which all batches are empty. Thus, in our constructions every possible input will have a small number of item sizes, but the analysis requires us to take into account the branching. The method of analysis of [3] allows branching and other features of inputs. Here, we adopt the method of analysis for branching, while the papers [6, 2, 11] contain a similar analysis, but only for inputs without any branching.

It is common to use weights in the analysis of bin packing algorithms [14, 1]. In this type of analysis, weights are assigned to items based on sizes or other properties of the packing. Then, the costs of the analyzed algorithm and an optimal solution are bounded via the total weight of input items. In this work, we use weights in another way, and weighting functions are employed for proving lower bounds on the asymptotic competitive ratio of batched bin packing algorithms. Weights are once again used for comparison. Specifically, they are used for bounding the cost of the algorithm, while calculating the optimal cost for specific inputs (used as inputs of an adversary) is more straightforward than for general inputs. There is still an analysis of total weights for bins, but it is easier in some sense than such computations for upper bounds, since there are only a few kinds of items, which may be a part of the input construction.

The idea (on which we elaborate below in the specific cases) is to assign weights to items, once again allowing an indirect comparison between the algorithm and an optimal solution. Since we use inputs with branching, weights of bins will not be defined in a usual way, but such that all possible contents will be taken into account simultaneously. We focus on the cases of small numbers of batches (since for larger numbers of batches, the known lower bounds are almost as good as those of the purely online problem), and on small parameters r (since larger values of r correspond to very small items, and the improvement compared to previous lower bound is extremely small). Our focus allows us to provide several new results and to exhibit the simplicity of the method. It also allows us to prove previous results in a significantly easier way.

2 The constructions

In this section we present our results. First, we present notation and definitions used in the rest of the section. Then, we present the results in increasing order of the number of batches.

2.1 Common definitions and notation

We define inputs denoted by $I(r, b)$, where $r \geq 1$ is an integer parameter such that all items sizes are in $(0, \frac{1}{r}]$, and the number of batches b satisfies $b \in \{2, 3, 4\}$. Input $I(r, b, i)$, where $1 \leq i \leq b$, will be defined as the prefix of input $I(r, b)$ consisting of its first i batches (so $I(r, b, b) = I(r, b)$).

Assume that r is fixed. Let $j > 100$ be a fixed integer and let N denote an integer divisible by $(j! \cdot ((r+2)^2)!)$ (where different values of N may be used, and in particular we will let N grow to infinity). For every integer a with $a \geq r+1$ (for the parametric problem with items of sizes at most $\frac{1}{r}$), we define a value $\varepsilon(a) = \frac{1}{j \cdot a(a-1)}$, where $\varepsilon(a) > 0$ and $\varepsilon(a) < \frac{1}{200}$.

For every input, a will be fixed (in addition to j and r, which are fixed for every input), and a tiny item will be an item of size $\varepsilon(a)$. We use ε to denote $\varepsilon(a)$, when a is fixed (i.e., $\varepsilon = \frac{1}{j \cdot a(a-1)}$). Each one of the inputs $I(r, b)$ starts with N tiny items, which will be the first batch of every input. To define the first batch completely for each input, we will specify the value of a and fix it. Let $\delta(a, r) = \frac{(\varepsilon(a))^2}{8r^3}$.

Given a value of r, for an integer a, an item of size $\frac{1}{a}+k\cdot\varepsilon(a)-4r\delta(a,r)$ with $1\leq k\leq j$ is called a large (k,a)-item. For every construction, we will have a single value of a for which there may be large (k,a)-items (while k can take multiple values even in the same construction). Given a fixed value of a, let $\delta=\delta(a,r)$. An item of size $\frac{1}{\alpha}+\delta$ (for an integer $2\leq\alpha\leq\min\{a,r+2\}$, where typically $\alpha\in\{r+1,r+2\}$) is called a regular (r,α)-item. We have $k\cdot\varepsilon(a)-4r\delta=k\varepsilon-\frac{\varepsilon^2}{2r^2}\geq\varepsilon-\varepsilon^2>0$, as $\varepsilon<1$. On the other hand $k\cdot\varepsilon(a)-4r\delta<k\varepsilon\leq\frac{1}{a(a-1)}\leq\frac{1}{\alpha(\alpha-1)}$, as $k\leq j$ and $\alpha\leq a$, so the size of a large (k,a)-item is in

$$(\frac{1}{a},\frac{1}{a-1})\ ,$$

and we refer to these bounds as the property of large items. A regular (r,α)-item has size above $\frac{1}{\alpha}$ and at most $\frac{1}{\alpha}+\delta<\frac{1}{\alpha-1}$, since $\delta<\frac{1}{8r^3}<\frac{1}{(r+2)(r+1)}$ holds by $r\geq 1$ and by $\alpha\leq r+2$.

2.2 Two batches

The classic approach. Before we analyze our input consisting of two batches, we analyze a simpler input, and this simpler input motivates our improved construction. Let $a=r+1$. The input will have a batch with items of sizes in $(\frac{1}{r+2},\frac{1}{r+1})$ and another batch with items of sizes in $(\frac{1}{r+1},\frac{1}{r})$. The first batch has N items, each of size $\frac{1}{r+1}-r\cdot\delta(r+1,r)$, and it is possibly followed by second a batch of $r\cdot N$ items that are regular $(r,r+1)$-items (of sizes $\frac{1}{r+1}+\delta(r+1,r)$, and in the interval $(\frac{1}{r+1},\frac{1}{r})$). By $\delta<\varepsilon<1$, we get $r\cdot\delta\leq\frac{1}{8r^2}<\frac{1}{(r+1)(r+2)}$, and therefore $\frac{1}{r+1}-r\cdot\delta>\frac{1}{r+2}$, and the items of the first batch have the claimed sizes.

We use an auxiliary tool called weights for the analysis. The weights are used for comparing the online packings to optimal ones. We start with analyzing this simple input using weights in order to introduce a straightforward version of our method of analysis. Each input item has a weight associated with it. The weight of a bin is defined as the total weight of its items. The weight will include all items of all batches that will possibly be packed there for all possible inputs and all cases, no matter how the input continues. Obviously, every item is packed into one bin and therefore the total weight of all items, denoted by W, is equal to the total weight of all bins of some packing. Note that the total size of items of a bin may exceed 1 if there is branching in the instance, but the total size of items that will be packed simultaneously will not exceed 1. One can think of such a bin as a virtual bin. In the first example, there is just one way to continue the inputs.

Specifically, we will use unit weights for all items of both batches of the current input. Let X denote the number of bins opened by the algorithm for the first batch (the first item ever packed into it is of the first batch), and let Y be the number of bins opened by the algorithm for items of the second batch.

As all item sizes are above $\frac{1}{r+2}$, the weight of a bin opened for the first batch is at most $r+1$. In the second batch, item sizes are above $\frac{1}{r+1}$, and therefore the weight of a bin opened for the second batch is at most r. The total weight of all items (of both batches) is $W=N+r\cdot N=N(r+1)$, and based on the upper bounds on weights of bins, we get $W\leq(r+1)X+rY$. Based on the two properties of W we get $N(r+1)\leq(r+1)X+rY$.

An offline solution for the first batch uses $\frac{N}{r+1}$ bins, each with $r+1$ items. An offline solution for the two batches has N bins, each with one item of the first batch and r items of the second batch (such that the total size for each bin is 1). Using the definition of (the asymptotic) competitive ratio and the property that the cost of an optimal solution is no smaller than $\frac{N}{r+1}$, and it is not smaller than N if there are two batches (so we can treat the asymptotic competitive ratio as the absolute competitive ratio), we use $X \leq R \cdot \frac{N}{r+1}$ and $X + Y \leq R \cdot N$, for the competitive ratio R. Taking the sum of the first inequality plus the second inequality times r, we have

$$(r+1)X + rY \leq R \cdot N \cdot (\frac{1}{r+1} + r) = RN \cdot \frac{r^2+r+1}{r+1} \ .$$

Next, we consider the bound which was proved above based on total weights: $N(r+1) \leq (r+1)X + rY$, and get $N(r+1) \leq (r+1)X + rY \leq RN \cdot \frac{r^2+r+1}{r+1}$, to find $R \geq \frac{(r+1)^2}{r^2+r+1}$. For $r = 1$, this gives a lower bound of $\frac{4}{3}$ on the asymptotic competitive ratio (see [8, 15]). For $r = 2, 3, 4, 5$, it gives 1.285714, 1.230769, 1.190476, and 1.16129, respectively.

The improved construction. The drawback of this construction is that the structure of the second batch is known in advance, and therefore there are just two cases for the input and its analysis, and an algorithm will be sufficiently successful in preparing for the two options. In our construction of an improved input, the first batch has N items of sizes $\varepsilon = \varepsilon(r+1)$, and the second batch of $I(r,2)$ consists of $\frac{N}{j-k}$ large (k,a)-items, where $a = r+1$ again, so these are large $(k, r+1)$-items. The possible values of k are $k = 1, 2, \ldots, d$, where $d < j$ is chosen later to optimize the resulting lower bound.

Using the property that $\varepsilon \cdot j \cdot a(a-1) = 1$, one can pack $j \cdot a(a-1)$ first batch items into every bin, and since N is divisible by $j \cdot a(a-1) = j \cdot r(r+1)$, we have $OPT(I(r,2,1)) = \varepsilon \cdot N = \frac{N}{j \cdot r(r+1)}$. Next, we show that $OPT(I(r,2,2)) \leq \frac{N}{r(j-k)}$. A feasible packing of the entire input is constructed by packing r large $(k, r+1)$-items into each bin, and adding $r(j-k)$ tiny items to each bin. Using the value of $\varepsilon = \varepsilon(r+1)$, the total size of items packed into a bin is

$$r(j-k)\varepsilon + r \cdot (\frac{1}{r+1} + k \cdot \varepsilon - 4r\delta) < r(j-k)\varepsilon + r \cdot (\frac{1}{r+1} + k \cdot \varepsilon) = \frac{r}{r+1} + r \cdot j \cdot \varepsilon = \frac{r}{r+1} + \frac{1}{r+1} = 1 \ .$$

Thus, for the given value of k, we have $OPT(I(r,2,2)) \leq \frac{N}{r(j-k)}$ (since N is divisible by $r(j-k)$).

For the second construction we use weights of 1 for tiny items. The weight of a large $(1, r+1)$-item is $j \cdot r + d - r(d-1)$, and for $k \geq 2$, the weight of a large $(k, r+1)$-item is r. Recall that the weight of a bin opened for the first batch is defined based on all possible contents of the bin in all cases, that is, on d cases.

Lemma 1 *The total weight of a bin created for the first batch is at most $j \cdot r(r+1)$. The total weight of a bin created for the second batch is at most r^2 if $k \geq 2$, and it is at most $r(j \cdot r + d - r(d-1))$ if $k = 1$.*

Proof. The upper bound on the weight of a bin opened for the second batch follows from the property that every bin can have at most r items of this batch.

For the first batch, we take into account all possible items of the second batch. Consider a bin, and let β be its number of tiny items. Note that the maximum number of tiny items which a bin can contain is $\frac{1}{\varepsilon} = j \cdot r(r+1)$. The remaining space in the bin is $1 - \beta\varepsilon = 1 - \frac{\beta}{jr(r+1)} = (jr(r+1) - \beta) \cdot \varepsilon$, while the size of a large $(k, r+1)$-item is $\frac{1}{r+1} + k \cdot \varepsilon - 4r\delta = (jr+k)\varepsilon - 4r\delta$, satisfying the property of large items for $a = r+1$, that is, the size is in $(\frac{1}{r+1}, \frac{1}{r})$.

Let

$$t = \left\lfloor \left(\frac{jr(r+1) - \beta}{jr(r+1)} \right) \div \left(\frac{1}{r+1} + \frac{1}{jr(r+1)} - 4r\delta \right) \right\rfloor = \left\lfloor \frac{jr(r+1) - \beta}{jr+1} \right\rfloor$$

be the number of large $(1, r+1)$-items that can be added to the bin (we consider the smallest large $(k, r+1)$ item here, for which the multiplier of ε in the size is 1). The equality in the definition of t holds as $t \cdot 4r\delta$ is clearly smaller than ε (using $t \le r$) and therefore the number of tiny items that fit together with t large $(k, r+1)$-items is the same as the number of tiny items that fit together with t items of sizes of $\frac{1}{r+1} + k\varepsilon$. Recall that due to the property of large items, we have $0 \le t \le r$, which we will use several times.

If $t = 0$, then no other items can be added to the bin except for tiny items, and the total weight is at most $jr(r+1)$, as this is the maximum number of tiny items. Thus, in further calculations we assume $1 \le t \le r$. If it is also feasible to add t items of any size (of the second batch), i.e., $\beta + t(jr+d) \le jr(r+1)$ (by considering integer multiples of ε), the total weight is at most

$$\beta + t((jr + d - r(d-1)) + (d-1) \cdot r)) = \beta + t(jr + d) \le jr(r+1) \ .$$

We used here the property that at most $jr(r+1)$ items of size ε can be packed into a bin, and every large $(d, r+1)$-item occupies the space of at least $j \cdot r + d$ items of size ε, and if we replace the size of such items with $\frac{1}{r+1} + d\varepsilon$, then it occupies the space of exactly these new tiny items.

Thus, using integrality of all values, assume that $\beta + t(jr+d) \ge jr(r+1)+1$ (but $\beta + t(jr+1) \le jr(r+1)$, by the definition of t).

There are some values of k such that t large $(k, r+1)$-items can be added to the bin, and the largest such value of $k \ge 1$ is $k' = \lfloor \frac{jr(r+1)-\beta}{t} - jr \rfloor \le d-1$ (by $\beta + t(jr+d) \ge jr(r+1)+1$), while for larger values of k at most $t-1$ items can be added.

For the first $t-1$ items, weights are as in the previous calculation, while for the tth item there are only $k'-1$ values of k contributing r, and the total weight is at most

$$\beta + (t-1) \cdot ((jr + d - r(d-1) + (d-1)r)) + (jr + d - r(d-1) + (k'-1)r)$$

$$= \beta + (t-1) \cdot (jr + d) + (jr + d - rd + k'r) = \beta + tjr + td - rd + k'r \ .$$

We would like to show that $\beta + tjr + td - rd + k'r \le jr(r+1)$ holds. To this end, we define $x = jr(r+1) - \beta - tjr$, where $t \le x \le td - 1$ by earlier bounds (and $k' = \lfloor \frac{x}{t} \rfloor$). The value x is the number of items of size ε that can still be added to a bin containing the already existing β such items, and t items, each of size $\frac{1}{r+1}$, which is smaller than any large item that we consider.

We have

$$\beta + tjr + td - rd + k'r \le jr(r+1) - x + td - rd + \frac{xr}{t} \ .$$

It is sufficient to show that $-x + td - rd + \frac{xr}{t} \le 0$ holds. Indeed $-x + td - rd + \frac{xr}{t} = (x - td)(r/t - 1)$, which is non-positive as $t \le r$ and $x < td$. ■

Let X be the number of bins opened for the first batch, and Y_i the number of bins opened for the second batch in the case where its items are large $(i, r+1)$-items. If R is the asymptotic competitive ratio, we have $X \le R \cdot OPT(I(r,2,1)) \le \frac{RN}{jr(r+1)}$ and $X + Y_i \le R \cdot OPT(I(r,2,2)) \le \frac{RN}{r(j-i)}$. Multiplying the first inequality by $r(j-d)$, the second inequality for $i = 1$ by $jr^2 + dr - dr^2 + r^2$, and multiplying all inequalities of the second family for $i = 2, 3, \ldots, d$ by r^2 (where all these multipliers are non-negative since $d < j$), and taking the sum we get

$$X((rj-rd)+(jr^2+dr-dr^2+r^2)+r^2(d-1))+(jr^2+dr-dr^2+r^2)Y_1+\sum_{i=2}^{d} r^2 Y_i$$
$$\le RN\cdot\left(\frac{(r(j-d))}{jr(r+1)}+\frac{jr^2+dr-dr^2+r^2}{r(j-1)}+\sum_{i=2}^{d}\frac{r^2}{r(j-i)}\right)$$
$$=RN\cdot\left(\frac{j-d}{j(r+1)}+\frac{jr+d-dr}{j-1}+\sum_{i=1}^{d}\frac{r}{j-i}\right) .$$

Note that the coefficient of X on the left hand side is $jr(r+1)$ (which is easy to see by algebraic transformations).

By counting the number of items and their weights, we have

$$W=N+\frac{N}{j-1}(j\cdot r+d-r(d-1))+r\cdot\sum_{i=2}^{d}\frac{N}{j-i}=N+\frac{N}{j-1}(j\cdot r+d-rd)+r\cdot\sum_{i=1}^{d}\frac{N}{j-i} .$$

By using the upper bounds on weights of bins, we also have

$$W\le X\cdot j\cdot r(r+1)+Y_1\cdot r(j\cdot r+d-r(d-1))+r^2\cdot\sum_{i=2}^{d}Y_i\le RN\cdot\left(\frac{j-d}{j(r+1)}+\frac{jr+d-dr}{j-1}+\sum_{i=1}^{d}\frac{r}{j-i}\right) .$$

Thus,

$$R\ge\frac{1+\frac{jr+d-dr}{j-1}+\sum_{i=1}^{d}\frac{r}{j-i}}{\frac{j-d}{j(r+1)}+\frac{jr+d-dr}{j-1}+\sum_{i=1}^{d}\frac{r}{j-i}} .$$

Letting $d=\gamma\cdot j$ where $\gamma<1$, and letting j grow to infinity, we get

$$R\ge\frac{1+r+\gamma-\gamma\cdot r-r\cdot\ln(1-\gamma)}{(1-\gamma)/(r+1)+r+\gamma-\gamma\cdot r-r\cdot\ln(1-\gamma)} .$$

By selecting appropriate values of γ (that is, by selecting d, j with these approximate ratios), we get the following lower bounds for $r=1,2,3,4,5$, respectively: 1.3871356 (see [13]), 1.291832, 1.231961, 1.190812, and 1.161411. The approximate values of γ are 0.442, 0.192, 0.098, 0.058, and 0.0383 respectively.

We summarize the new results of this section in the following theorem.

Theorem 2 *The asymptotic competitive ratio for batched bin packing with two batches and items of sizes in* $(0,\frac{1}{r}]$ *is at least* 1.291832*,* 1.231961*,* 1.190812*, and* 1.161411 *for* $r=2,3,4$ *and* 5*, respectively.*

2.3 Three batches

We will use the same approach with respect to weights as in the previous section.

Here, the input starts with N tiny items again. We use tiny items of sizes $\varepsilon(r+2)=\frac{1}{j(r+1)(r+2)}$. The second batch consists of large $(k,r+2)$-items (i.e., $a=r+2$) for $1\le k<d$, and the third batch consists of regular $(r,r+1)$-items no matter what the value of k is. The number of large $(k,r+2)$-items is $\frac{N}{j-k}$, and the number of regular $(r,r+1)$-items is $\frac{r\cdot N}{j-k}$.

We have $OPT(I(r,3,1)) = \varepsilon(r+2)\cdot N = \frac{N}{j(r+1)(r+2)}$, since N is divisible by $j(r+1)(r+2)$. Additionally, $OPT(I(r,3,2)) \le \frac{N}{(r+1)\cdot(j-k)}$, since every $r+1$ large $(k,r+2)$-items can be combined with $(r+1)(j-k)$ tiny items in a bin, which holds as

$$(r+1)(j-k)\varepsilon + (r+1)(\frac{1}{r+2} + k\varepsilon - 4r\delta) < \frac{r+1}{r+2} + (r+1)j\varepsilon = \frac{r+1}{r+2} + \frac{1}{r+2} = 1 \ .$$

Finally, $OPT(I(r,3,3)) \le \frac{N}{j-k}$, as it is possible to pack r regular $(r,r+1)$-items, one large (k,ℓ)-item, and $j-k$ tiny items into a bin since

$$(j-k)\varepsilon + (\frac{1}{r+2} + k\varepsilon - 4r\delta) + r(\frac{1}{r+1} + \delta)$$

$$< j\varepsilon + \frac{1}{r+2} + \frac{r}{r+1} = \frac{1}{(r+1)(r+2)} + \frac{1}{r+2} + \frac{r}{r+1} = 1 \ ,$$

where the calculation is based on algebraic transformations.

We define weights as follows. Every tiny item has weight 1. For $k \ge 2$, every large $(k,r+2)$-item has weight $r+1$, and so does every regular $(r,r+1)$-item that is presented after large $(k,r+2)$-items (for $k \ge 2$). For $k=1$, every large $(1,r+2)$-item has weight $j\cdot(r+1)+d-(r+1)(d-1) = j(r+1)+r+1-rd$, and so does every regular $(r,r+1)$-item that is presented after large $(1,r+2)$-items.

Lemma 3 *The total weight of a bin created for the first batch is at most $j\cdot(r+1)(r+2)$. The total weight of a bin created for the second batch is at most $(r+1)^2$ if $k \ge 2$, and it is at most $(r+1)(j\cdot(r+1)+d-(r+1)(d-1))$ if $k=1$. The total weight of a bin created for the third batch is at most $r(r+1)$ if $k \ge 2$, and it is at most $r(j\cdot(r+1)+d-(r+1)(d-1))$ if $k=1$.*

Proof. The upper bound on the weight of a bin opened for the second batch follows from the property that every such bin can have at most $r+1$ items, and for the third batch, since it can have at most r items.

Consider a bin created in the first batch. For any case where the bin has a regular $(r,r+1)$-item, replace every such item with a large $(k,r+2)$-item arriving just before it (a second batch item of the same case). The replacing item is not larger (as the large item has size below $\frac{1}{r+1}$ and the regular item has size above $\frac{1}{r+1}$), and by definition these two items have equal weights. We are left with items of two batches and the claim follows from the previous lemma with $r+1$ instead of r, which is possible since r is a variable. ■

Let X be the number of bins opened for the first batch, Y_i the number of bins opened for the second batch if its items are large $(i,r+2)$-items (for $1 \le i \le d$), and let Z_i be the number of bins opened in that case for the third batch.

If R is the asymptotic competitive ratio, we have $X \le R\cdot OPT(I(r,3,1)) \le \frac{RN}{j(r+1)(r+2)}$, $X+Y_i \le R\cdot OPT(I(r,3,2)) \le \frac{RN}{(r+1)(j-i)}$, and $X+Y_i+Z_i \le R\cdot OPT(I(r,3,3)) \le \frac{RN}{j-i}$.

Multiplying the first inequality by $(r+1)(j-d)$, the second inequality for $i=1$ by $j(r+1)+r+1-rd$, all these inequalities for $i=2,3,\ldots,d$ by $r+1$, and the third inequality for $i=1$ by $r(j(r+1)+r+1-rd)$ and $r(r+1)$ for $i=2,3,\ldots,d$ (where all these multipliers are non-negative since $d < j$), and taking the sum we get

$$(r+1)(j-d)\cdot X + (j(r+1)+r+1-rd)(X+Y_1) + (r+1)\sum_{i=2}^{d}(X+Y_i)$$

$$+(r(j(r+1)+r+1-rd))(X+Y_1+Z_1) + r(r+1)\sum_{i=2}^{d}(X+Y_i+Z_i)$$

$$\leq R\cdot N\cdot\left(\frac{(r+1)(j-d)}{j(r+1)(r+2)} + \frac{j(r+1)+r+1-rd}{(r+1)(j-1)}\right.$$

$$\left. +(r+1)\sum_{i=2}^{d}\frac{1}{(r+1)(j-i)} + \frac{r(j(r+1)+r+1-rd)}{j-1} + r(r+1)\sum_{i=2}^{d}\frac{1}{j-i}\right),$$

which is equivalent to

$$\Big((r+1)(j-d)+(r+1)(j(r+1)+r+1-rd)+(r+1)^2(d-1)\Big)X + \Big((r+1)(j(r+1)+r+1-rd)\Big)Y_1$$

$$+\sum_{i=2}^{d}(r+1)^2Y_i + \Big(r(j(r+1)+r+1-rd)\Big)Z_1 + \sum_{i=2}^{d} r(r+1)Z_i$$

$$\leq RN\cdot\left(\frac{(r+1)(j-d)}{j(r+1)(r+2)} + \frac{(r^2+r+1)(j(r+1)+r+1-rd)}{(r+1)(j-1)} + \sum_{i=2}^{d}\frac{r^2+r+1}{j-i}\right)$$

$$= RN\cdot\left(\frac{j-d}{j(r+2)} + \frac{(r^2+r+1)(j(r+1)-rd)}{(r+1)(j-1)} + \sum_{i=1}^{d}\frac{r^2+r+1}{j-i}\right).$$

Note that the coefficient of X on the left hand side is $j(r+1)(r+2)$.

Counting the items and their weights, we have

$$W = N + \frac{N}{j-1}(r+1)(j\cdot(r+1)+r+1-rd) + (r+1)^2\cdot\sum_{i=2}^{d}\frac{N}{j-i}$$

$$= N + \frac{N}{j-1}(r+1)(j\cdot(r+1)-rd) + (r+1)^2\cdot\sum_{i=1}^{d}\frac{N}{j-i}.$$

Based on the upper bound on bin weights, we also have

$$W \leq X\cdot j\cdot(r+1)(r+2) + ((r+1)Y_1+rZ_1)\cdot(j\cdot(r+1)+d-(r+1)(d-1)) + \sum_{i=2}^{d}((r+1)^2Y_i + r(r+1)Z_i)$$

$$\leq RN\cdot\left(\frac{j-d}{j(r+2)} + \frac{(r^2+r+1)(j(r+1)-rd)}{(r+1)(j-1)} + \sum_{i=1}^{d}\frac{r^2+r+1}{j-i}\right).$$

Thus,

$$R \geq \frac{1+(r+1)\cdot\frac{j(r+1)-dr}{j-1} + (r+1)^2\cdot\sum_{i=1}^{d}\frac{1}{j-i}}{\frac{j-d}{j(r+2)} + \frac{r^2+r+1}{r+1}\cdot\frac{j(r+1)-dr}{j-1} + (r^2+r+1)\cdot\sum_{i=1}^{d}\frac{1}{j-i}}.$$

Letting $d = \gamma \cdot j$ and letting j grow to infinity, we get

$$R \geq \frac{1 + (r+1)(r+1-\gamma \cdot r) - (r+1)^2 \ln(1-\gamma)}{(1-\gamma)/(r+2) + \frac{r^2+r+1}{r+1}(r+1-\gamma \cdot r) - (r^2+r+1)\ln(1-\gamma)} .$$

The lower bounds of [5, 16] for three batches and $r = 1, 2, 3, 4, 5$ were 1.5, 1.3793103, 1.2878787, 1.2283464, and 1.1880733, respectively. The case $r = 1$ follows from an earlier result of Yao [17].

By selecting appropriate values of γ, we get the following lower bounds for $r = 1, 2, 3, 4, 5$, respectively: 1.51211 (see [4]), 1.3807199, 1.28812, 1.2284, 1.18809. The approximate values of γ are 0.30544921, 0.15664, 0.0871875, 0.053984375, and 0.03628125 respectively.

We summarize the new results of this section in the following theorem.

Theorem 4 *The asymptotic competitive ratio for batched bin packing with three batches and items of sizes in* $(0, \frac{1}{r}]$ *is at least* 1.3807199*,* 1.28812*,* 1.2284 *and* 1.18809*, for* $r = 2, 3, 4$ *and* 5*, respectively.*

2.4 Four batches

Here, the input starts with N tiny items again. Let $\ell = (r+1)(r+2)+1$. We use tiny items of sizes $\varepsilon(\ell) = \frac{1}{j\ell(\ell-1)}$. The second batch consists of large (k, ℓ)-items for $1 \leq k \leq d$, where $d < j$. The third batch and fourth batches consist of regular $(r, r+2)$-items and regular $(r, r+1)$-items, respectively, no matter what the value of k is. The number of large (k, ℓ)-items is $\frac{N}{j-k}$, the number of regular $(r, r+2)$-items is $\frac{N}{j-k}$, and the number of regular $(r, r+1)$-items is $\frac{r \cdot N}{j-k}$.

We have $OPT(I(r, 4, 1)) = \varepsilon(\ell) \cdot N = \frac{N}{j \cdot \ell(\ell-1)}$. Additionally, $OPT(I(r, 4, 2)) \leq \frac{N}{(\ell-1)\cdot(j-k)}$, since every $\ell - 1$ large (k, ℓ)-items can be combined with $(\ell-1)(j-k)$ tiny items in a bin, which holds as

$$(\ell-1)(j-k)\varepsilon + (\ell-1)(\frac{1}{\ell} + k\varepsilon - 4r\delta(\ell, r)) < \frac{\ell-1}{\ell} + (\ell-1)j\varepsilon = 1 .$$

Next, $OPT(I(r, 4, 3)) \leq \frac{N}{(r+1)(j-k)}$, since every $r+1$ large (k, ℓ)-items can be combined with $(r+1)(j-k)$ tiny items and $r+1$ regular $(r, r+2)$-items in a bin, which holds as $(r+1)(j-k)\varepsilon + (r+1)(\frac{1}{\ell} + k\varepsilon - 4r\delta) + (r+1)(\frac{1}{r+2} + \delta) < (r+1)j\varepsilon + \frac{r+1}{\ell} + \frac{r+1}{r+2} = \frac{r+1}{\ell(\ell-1)} + \frac{r+1}{\ell} + \frac{r+1}{r+2} = \frac{r+1}{\ell-1} + \frac{r+1}{r+2} = 1$, by the definition of ε and $\ell = (r+1)(r+2)+1$.

Finally, $OPT(I(r, 4, 4)) \leq \frac{N}{j-k}$, as it is possible to pack r regular $(r, r+1)$-items, one regular $(r, r+2)$-item, one large (k, ℓ)-item, and $j-k$ tiny items into a bin since

$$(j-k)\varepsilon + (\frac{1}{\ell} + k\varepsilon - 4r\delta) + (\frac{1}{r+2} + \delta) + r(\frac{1}{r+1} + \delta)$$

$$< \frac{1}{\ell(\ell-1)} + \frac{1}{\ell} + \frac{1}{r+2} + \frac{r}{r+1} = \frac{1}{(r+1)(r+2)} + \frac{1}{r+2} + \frac{r}{r+1} = 1 .$$

We define weights as follows. Every tiny item has weight 1. For $k \geq 2$, every large (k, ℓ)-item has weight $\ell - 1$. The weight of every regular $(r, r+2)$-item and every regular $(r, r+1)$-item that is presented in these cases after large (k, ℓ)-items is $(\ell-1)(r+1)$. Every large $(1, \ell)$-item has weight $j \cdot (\ell-1) + d - (\ell-1)(d-1)$, and every regular $(r, r+2)$-item and every regular $(r, r+1)$-item presented after large $(1, \ell)$-items has weight $(r+1)(j \cdot (\ell-1) + d - (\ell-1)(d-1))$.

Lemma 5 *The total weight of a bin created for the first batch is at most* $j \cdot \ell(\ell-1)$*. If* $k \geq 2$*, the total weights of bins created for the second, third, and fourth batches are at most* $(\ell-1)(r+1)(r+2)$*,*

$(\ell-1)(r+1)^2$, and $(\ell-1)r(r+1)$, respectively. If $k=1$, the total weights of bins created for the second, third, and fourth batches are at most

$$(j\cdot(\ell-1)+d-(\ell-1)(d-1))(r+1)(r+2)\ ,$$

$(j\cdot(\ell-1)+d-(\ell-1)(d-1))(r+1)^2$, and $(j\cdot(\ell-1)+d-(\ell-1)(d-1))r(r+1)$, respectively.

Proof. The upper bound on the weight of a bin opened for the second batch, third batch, and fourth batch, respectively, follows from the property that every bin can have at most $\ell-1$ items for the second batch, at most $r+1$ items for the third batch, and at most r items for the fourth batch. This leads using the fact that in each case the weight of all these items is equal to the conclusion that the total weight cannot be larger than the claimed value.

Consider a bin created in the first batch. For any case where the bin has a regular $(r, r+2)$-item or a regular $(r, r+1)$-item, replace every such item with $r+1$ large (k,ℓ)-items arriving just before it (a second batch item of the same case). The total size of replacing items is not larger (as the large item has size below $\frac{1}{(r+1)(r+2)}$ and the regular item has size above $\frac{1}{r+2}$, and by definition the total weight is not decreased. We are left with items of two batches and the claim follows from the Lemma 1 with $\ell-1$ instead of r. ■

Let X be the number of bins opened for the first batch, Y_i the number of bins opened for the second batch if its items are large (i,ℓ)-items (for $1\leq i\leq d$), let Z_i be the number of bins opened in that case for the third batch, and let U_i be the number of bins opened in that case for the fourth batch.

If R is the asymptotic competitive ratio, we have $X\leq R\cdot OPT(I(r,4,1))\leq\frac{RN}{j\cdot\ell(\ell-1)}$, $X+Y_i\leq R\cdot OPT(I(r,4,2))\leq\frac{RN}{(\ell-1)\cdot(j-i)}$, $X+Y_i+Z_i\leq R\cdot OPT(I(r,4,3))\leq\frac{RN}{(r+1)(j-i)}$, and

$$X+Y_i+Z_i+U_i\leq R\cdot OPT(I(r,4,4))\leq\frac{RN}{j-i}\ .$$

Multiplying the first inequality by $(\ell-1)(j-d)$, the second and third inequalities for $i=1$ by $(r+1)(j\cdot(\ell-1)+d-(\ell-1)(d-1))$ and the second and third inequalities for $i=2,3,\ldots,d$ by $(r+1)(\ell-1)$, and the fourth inequality by $r(r+1)(j\cdot(\ell-1)+d-(\ell-1)(d-1))$ for $i=1$ and by $r(r+1)(\ell-1)$ for $i=2,3,\ldots,d$ (where all these multipliers are non-negative since $d<j$), and taking the sum we get

$$X((\ell-1)(j-d)+(r+1)(r+2)(j(\ell-1)+d-(\ell-1)(d-1))$$
$$+(r+1)(r+2)(\ell-1)(d-1)))+(r+1)(r+2)(j(\ell-1)+d-(\ell-1)(d-1))Y_1$$
$$+(r+1)^2(j(\ell-1)+d-(\ell-1)(d-1))Z_1+r(r+1)(j(\ell-1)+d-(\ell-1)(d-1))U_1$$
$$+(\ell-1)(r+1)\sum_{i=2}^{d}((r+2)Y_i+(r+1)Z_i+rU_i)$$
$$\leq RN\cdot(\frac{(\ell-1)(j-d)}{j\ell(\ell-1)}+\frac{(r^3+3r^2+3r+3)(j(\ell-1)+d-(\ell-1)(d-1))}{(r+2)(j-1)}$$
$$+\sum_{i=2}^{d}\frac{(r^3+3r^2+3r+3)(r+1)}{j-i})$$

$$= RN \cdot \left(\frac{j-d}{j\ell} + \frac{(r^3+3r^2+3r+3)(j(\ell-1)-(\ell-2)d))}{(r+2)(j-1)} + \sum_{i=1}^{d} \frac{(r^3+3r^2+3r+3)(r+1)}{j-i} \right) .$$

Using $\ell - 1 = (r+1)(r+2)$ and algebraic transformations, we find that the coefficient of X on the left hand side is $j\ell(\ell-1)$.

Counting the items and their weights, we have

$$W = N + (r^2+2r+2)\frac{N}{j-1}(j\cdot(\ell-1)+d-(\ell-1)(d-1)) + (\ell-1)(r^2+2r+2)\cdot \sum_{i=2}^{d} \frac{N}{j-i}$$

$$= N(1+(r^2+2r+2)\frac{1}{j-1}(j\cdot(\ell-1)+2d-d\ell) + (\ell-1)(r^2+2r+2)\cdot \sum_{i=1}^{d} \frac{1}{j-i}) .$$

Based on the upper bound on bin weights, we also have

$$W \le X \cdot j \cdot \ell(\ell-1) + (r+1)((r+2)Y_1 + (r+1)Z_1 + rU_1)\cdot(j\cdot(\ell-1)+d-(\ell-1)(d-1))$$

$$+(\ell-1)(r+1)\sum_{i=2}^{d}((r+2)Y_i+(r+1)Z_i+rU_i)$$

$$\le RN \cdot \left(\frac{j-d}{j\ell} + \frac{(r^3+3r^2+3r+3)(j(\ell-1)-(\ell-2)d))}{(r+2)(j-1)} + \sum_{i=1}^{d} \frac{(r^3+3r^2+3r+3)(r+1)}{j-i} \right) .$$

Thus,

$$R \ge \frac{1+\frac{r^2+2r+2}{j-1}(j\cdot(\ell-1)+2d-d\ell)+(\ell-1)(r^2+2r+2)\cdot\sum_{i=1}^{d}\frac{1}{j-i})}{\frac{j-d}{j\ell}+\frac{(r^3+3r^2+3r+3)(j(\ell-1)-(\ell-2)d))}{(r+2)(j-1)}+\sum_{i=1}^{d}\frac{(r^3+3r^2+3r+3)(r+1)}{j-i}} .$$

Letting $d = \gamma \cdot j$ and letting j grow to infinity, we get

$$R \ge \frac{1+(r^2+2r+2)(\ell-1-(\ell-2)\gamma)-(\ell-1)(r^2+2r+2)\cdot\ln(1-\gamma)}{(1-\gamma)/\ell+\frac{(r^3+3r^2+3r+3)}{r+2}(\ell-1-\gamma(\ell-2))-(r^3+3r^2+3r+3)(r+1)\ln(1-\gamma)} .$$

The lower bound of [16, 5] for four batches and $r = 1, 2, 3$ is 1.5390070, 1.3895759, and 1.2914337, respectively.

By selecting appropriate values of γ, we get the following lower bounds for $r = 1, 2, 3$, respectively: 1.5392406134, 1.389582558, and 1.29143414. The approximate values of γ are 0.102, 0.03608, and 0.0155, respectively.

We summarize the new results of this section in the following theorem.

Theorem 6 *The asymptotic competitive ratio for batched bin packing with four batches and items of sizes in* $(0, \frac{1}{r}]$ *is at least* 1.5392406134, 1.389582558, *and* 1.29143414 *for* $r = 1$, $r = 2$ *and* $r = 3$, *respectively.*

References

[1] J. Balogh, J. Békési, Gy. Dósa, L. Epstein, and A. Levin. A new and improved algorithm for online bin packing. In *Proc. of the 26th European Symposium on Algorithms (ESA2018)*, pages 5:1–5:14, 2018. Also in Journal of Computer and System Sciences, to appear.

[2] J. Balogh, J. Békési, Gy. Dósa, L. Epstein, and A. Levin. Lower bounds for several online variants of bin packing. *Theory of Computing Systems*, 63(8):1757–1780, 2019.

[3] J. Balogh, J. Békési, Gy. Dósa, L. Epstein, and A. Levin. A new lower bound for classic online bin packing. In *Proceedings of the 17th Workshop on Approximation and Online Algorithms (WAOA2019)*, pages 18–28, 2019.

[4] J. Balogh, J. Békési, Gy. Dósa, G. Galambos, and Z. Tan. Lower bound for 3-batched bin packing. *Discrete Optimization*, 21:14–24, 2016.

[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] D. J. Brown. A lower bound for on-line one-dimensional bin packing algorithms. *Coordinated Science Laboratory Report no. R-864 (UILU-ENG 78-2257)*, 1979.

[8] E. G. Coffman Jr. and J. Csirik. Performance guarantees for one-dimensional bin packing. In T. F. Gonzalez, editor, *Handbook of Approximation Algorithms and Metaheuristics*, chapter 32. Chapman & Hall/Crc, 2007. 18 pages.

[9] Gy. Dósa. Batched bin packing revisited. *Journal of Scheduling*, 20(2):199–209, 2017.

[10] L. Epstein. More on batched bin packing. *Operations Research Letters*, 44(2):273–277, 2016.

[11] L. Epstein. A lower bound for online rectangle packing. *Journal of Combinatorial Optimization*, 38(3):846–866, 2019.

[12] G. Galambos. Parametric lower bound for on-line bin-packing. *SIAM Journal on Algebraic Discrete Methods*, 7(3):362–367, 1986.

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

[14] 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.

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

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

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