Online bin packing: Old algorithms and new results

Jiří Sgall[1]

Computer Science Institute of Charles University, Faculty of Mathematics and Physics, Malostranské nám. 25, CZ-11800 Praha 1, Czech Republic, Czech Republic.
sgall@iuuk.mff.cuni.cz

Abstract. In the bin packing problem we are given an instance consisting of a sequence of items with sizes between 0 and 1. The objective is to pack these items into the smallest possible number of bins of unit size. FirstFit and BestFit algorithms are simple online algorithms introduced in early seventies, when it was also shown that their asymptotic approximation ratio is equal to 1.7. We present a simple proof of this bound and survey recent developments that lead to the proof that also the absolute approximation ratio of these algorithms is exactly 1.7. More precisely, if the optimum needs Opt bins, the algorithms use at most $\lfloor 1.7 \cdot \mathrm{OPT} \rfloor$ bins and for each value of Opt, there are instances that actually need so many bins. We also discuss bounded-space bin packing, where the online algorithm is allowed to keep only a fixed number of bins open for future items. In this model, a variant of BestFit also has asymptotic approximation ratio 1.7, although it is possible that the bound is significantly smaller if also the offline solution is required to satisfy the bounded-space restriction.

1 Introduction

Johnson's thesis [13] on bin packing together with Graham's work on scheduling [11, 12] belong to the early influential works that started and formed the whole area of approximation algorithms. The proof that the asymptotic approximation ratio of FirstFit and BestFit bin packing is 1.7 given by Ullman [20] and subsequent works by Garey et al. and Johnson et al. [10, 15] were among these first results on approximation algorithms.

We survey this area with emphasis on the new results, approaches and open problems.

1.1 The algorithms

Bin packing is a classical combinatorial optimization problem in which we are given an instance consisting of a sequence of items with rational sizes between 0 and 1, and the goal is to pack these items into the smallest possible number of bins of unit size. Since bin packing is NP-hard, one particularly active branch of research has concentrated on approximation algorithms that find near-optimal packings.

The main topic of this survey are BestFit, FirstFit and related algorithms. BestFit algorithm packs each item into the most full bin where it fits, possibly opening a new bin if the item does not fit into any currently open bin. FirstFit algorithm packs each item into the first bin where it fits, again opening a new bin only if the item does not fit into any currently open bin.

Both BestFit and FirstFit as well as any algorithm that we will consider belong to a wide class of algorithms called AnyFit algorithms. An algorithm is called an AnyFit algorithm if it opens a new bin only if the item being packed does not fit into any previously open bin. Otherwise the item is allowed to be packed into any bin where it fits.

A relevant subclass are AlmostAnyFit algorithms (AAF algorithms for short). An algorithm is an AAF algorithm if it is an AnyFit algorithm and additionally, if an item fits into more than one of the open bins, it is not packed into the smallest open bin (breaking ties arbitrarily).

The worst-case performance of an approximation algorithm A is measured by comparing it against the optimal packing. Let Opt denote the number of bins used in an optimal packing for instance I, and let A denote the number of bins used by algorithm A for the same instance. The algorithm A is an **absolute R-approximation** algorithm if for every instance I

$$A \leq R \cdot \textsc{Opt},$$

and it is an **asymptotic R-approximation** algorithm if for every instance I

$$A \leq R \cdot \textsc{Opt} + f(\textsc{Opt}),$$

where $f(n) \in o(n)$; typically, and in all cases relevant for us, $f(n)$ is a global constant.

As far as the worst-case ratios are concerned, FirstFit can be considered as a special case of BestFit or any other AnyFit algorithm: The items in any instance can be reordered so that they arrive in the order of bins in the FirstFit packing. This changes neither FirstFit, nor the optimal packing. Thus it is sufficient to analyze FirstFit on such instances. On the other hand, on them any AnyFit algorithm behaves exactly as FirstFit, as there is always a single bin where the new item fits. Thus any lower bound for FirstFit applies immediately to an arbitrary AnyFit algorithms and any upper bound for FirstFit is equivalent to a bound for this very restricted subset of instances and BestFit or another AnyFit algorithm.

Both FirstFit and BestFit are **online** algorithms. An online bin packing algorithm places each item without knowledge of the future items, that is based only on the size of this item and the already packed items, and cannot change the packing of the items in the future.

In the context of online algorithms, a very interesting concept is that of bounded-space algorithms. We distinguish the bins to be either open or closed and items are allowed to be packed only into open bins. When a new bin is started, we set the status of the bin to be open. At some point, the algorithm may decide to close some bin(s). This means that no future items can be packed there; once a bin is closed, it cannot be reopened. A bin packing algorithm is a **k-bounded-space** algorithm, if the number of open bins is at most k at all times. That is, if there are k open bins and an item is packed into a new bin, one of the open bins needs to be closed.

The natural bounded-space variant of BestFit is BBF_k, which packs an item into the most full open bin where it fits. If the item does not fit into any of them, a new bin is opened; if already k bins are open, the most full bin is closed. Note that BBF_k is not an AnyFit algorithm. It may happen that an item fits in one of the already closed bins but not into any open one.

1.2 The old results

The upper bound on BestFit (and FirstFit) was first shown by Ullman in 1971 [20]; he proved that for any instance, $\mathrm{BF}, \mathrm{FF} \leq 1.7 \cdot \mathrm{Opt} + 3$, where BF, FF and Opt denote the number of bins used by BestFit, FirstFit and the optimum, respectively. Still in seventies, the additive term was improved first in [10] to 2 and then in [9] to $\mathrm{BF} \leq \lceil 1.7 \cdot \mathrm{Opt} \rceil$; due to integrality of BF and Opt this is equivalent to $\mathrm{BF} \leq 1.7 \cdot \mathrm{Opt} + 0.9$. These bounds actually hold for any AAF algorithms, see [13, 14].

For the lower bound, the early works give examples both for the asymptotic and absolute ratios. The example for the asymptotic bound gives $\mathrm{FF} = 17k$ whenever $\mathrm{Opt} = 10k + 1$, thus it shows that the asymptotic upper bound of 1.7 is tight, see [20, 10, 15]. For the absolute ratio, an example is given with $\mathrm{FF} = 17$ and $\mathrm{Opt} = 10$, i.e., an instance with approximation ratio exactly 1.7 [10, 15], but no such example was known for large Opt. (Also an example with $\mathrm{FF} = 34$ and $\mathrm{Opt} = 20$ is claimed and it can be constructed in a similar manner, but it seems that this example has never been published.)

Johnson [13, 15] has also analyzed the Best Fit Decreasing and First Fit Decreasing algorithms, which behave like BestFit and FirstFit but receive the items on the input sorted from the largest one to the smallest, and proved that the asymptotic approximation ratio is equal to 11/9. Johnson's bound had an additive constant of 4; this was improved several times.

Turning to the bounded-space algorithms, a central result of Lee and Lee [16] designs k-bounded-space online bin packing algorithms whose asymptotic ratios come arbitrarily close to the magic harmonic number $h_\infty \approx 1.69103$, as the space bound k tends to infinity. They also show that every bounded-space online bin packing algorithm A satisfies $R^\infty(A) \geq h_\infty$

Csirik and Johnson [5] show that the k-space-bounded Best Fit algorithm BBF_k has the asymptotic worst case ratio 1.7 for any $k \geq 2$. (The case of $k = 1$ is trivial, 2-approximation is possible and no better algorithm exists.) Among all 2-space-bounded online algorithms in the bin packing literature, this online algorithm is the champion with respect to worst-case ratios.

We have mentioned only directly relevant work. Of course, there is much more work on bin packing, in particular there exist asymptotic approximation schemes for this problem, as well as many other algorithms. We refer to the surveys [3, 4] or to the recent excellent book [21].

1.3 The new results

The absolute approximation ratio of FirstFit and BestFit was proven to be at most 1.75 by Simchi-Levy [19]. This started a renewed interest in these algorithm and their absolute ratio and the closely related question of decreasing the additive constant in the approximation ratio. A natural approach to improve absolute upper bounds is to study fixed small values of Opt and to exclude the possibility of a higher absolute ratio for them. Indeed, solving a few such cases necessarily improves upper bounds on the absolute ratio—but cannot give a tight result. Of course, this is still far from trivial: Even for a fixed Opt, each such problem seems to lead to a new and more extensive case analysis.

Such analysis lead to a sequence of improvements for FirstFit: Xia and Tan [22] showed that $\mathrm{FF} \leq 1.7 \cdot \mathrm{Opt} + 0.7$ and that the absolute ratio of FirstFit is at most $12/7 \approx 1.7143$; the second bound was given independently by Boyar et al. [1] and later improved to $101/59 \approx 1.7119$ by Németh [17].

In [7, 8], Dósa and the author have shown that $\mathrm{FF}, \mathrm{BF} \leq 1.7 \cdot \mathrm{Opt}$ and that this bound is the best possible for every single value of Opt. Thus we have a complete understanding of the worst-case behavior of these algorithm. In particular, we know that the asymptotic and absolute bounds coincide and no additive term is needed. These results are based on new insights that also make possible a simple proof and a slight generalization and strengthening of the asymptotic bound for AAF algorithms.

Turning to other online algorithms, these new techniques can be also applied to BBF_k and simplify the proof that BBF_k is asymptotic 1.7-approximation, even though BBF_k is not AnyFit algorithm [18].

It is easy to observe that no online algorithm has an absolute ratio less than $5/3 \approx 1.666$. Recently, again using the new techniques, we (G. Dósa, R. van Stee and the author) have been able to find a matching algorithm (article in preparation).

For First Fit Decreasing, it was shown that the additive constant is exactly 2/3 [6]. That is, $\frac{11}{9}\mathrm{Opt} + \frac{2}{3}$ bins are sufficient for First Fit Decreasing, but this number of bins is actually also necessary for some instances for infinitely many values of Opt. Thus for First Fit Decreasing, the asymptotic and absolute approximation ratios are not equal. In fact, the results of [6] give the exact value of the worst case for every value of Opt. In light of this result, it is rather surprising that for BestFit and FirstFit the asymptotic and absolute approximation ratios are equal and no additive term is needed.

1.4 This survey

We focus on the results connected to the recent optimal analysis of FirstFit and BestFit. We start in Section 2 by sketching the lower bound, as it gives a good intuition for the choices made in the remaining parts of the paper.

In the Section 3 we give a simple proof of the asymptotic upper bound of 1.7 for AAF algorithms, in fact, even for a slightly more generalized class of GAAF algorithms (Generalized AAF algorithms). This proof uses the same insights as the optimal analysis of FirstFit and BestFit, namely a combination of weight functions and amortized analysis. In addition, to demonstrate that the equality of the absolute and asymptotic bounds for FirstFit and BestFit is by far not automatic, we show a GAAF packing with an absolute ratio strictly larger than 1.7.

Finally, in Section 4 we discuss a new measure for bounded space algorithm introduced in [2].

2 The lower bound

Suppose for a moment that the algorithm is not allowed to pack a bin with item sizes summing to exactly 1, while the optimum can. This model gives an unfair advantage to the optimum, but we hope that lower bound examples in this model can be later modified by changing the item sizes by very small amount to provide a correct lower bound. This is a bold assumption that is not quite true, however, this simplified model provides a very good high-level intuition in a number of different bin packing scenarios.

In the simplified model, the lower bound instance for $\text{Opt} = 10m$ is this: Start by Opt items of size $1/6$, continue by Opt items of size $1/3$, and conclude by Opt items of size $1/2$. The optimum uses $\text{Opt} = 10m$ bins by packing three items of different sizes $1/6 + 1/3 + 1/2 = 1$ in each bin. FirstFit uses $2m$ bins with 5 items of size $1/6$, $5m$ bins with 2 items of size $1/3$ and $10m$ bins with a single item of size $1/2$. Total of $17m$ bins, giving a lower bound of 1.7.

To modify the instance to be valid in the real model, we modify the item sizes. First, for all items $1/2$ the size is changed into $1/2+\varepsilon$ for a tiny $\varepsilon > 0$. The sizes of items $1/3$ is changed into carefully chosen $1/3+\delta_i$ and $1/3-\delta_i$, for δ_i tiny and exponentially decreasing to 0. This guarantees that the previous packing works if we arrange the sequence so that FirstFit packs $1/3+\delta_i$ and $1/3-\delta_{i-1}$ together. Now, to allow the optimal packing, the remaining items have size $1/6 + \delta_i - \varepsilon$ and $1/6 - \delta_i - \varepsilon$. We need to order them so that the FirstFit packing works as we wish. Here we come to a point where the simplified model fails. We modify the FirstFit packing so that the first bin contains 6 items instead of 5 and the last bin of the first phase contains 4 items instead of 5. Of course, this needs to be carefully checked, but the main idea why the construction works is that the exponential decrease of δ_i guarantees that only the item with the largest δ_i in a bin is relevant for its final size. Also, we need a proof for other residue classes of Opt, but the case of $\text{Opt} = 10m$ is tightest and easily modified to other $\text{Opt} = 10m + i$. We obtain

Theorem 2.1. *For all values of* Opt*, there exists an instance I with such that* $\text{FF} = \lfloor 1.7 \cdot \text{Opt} \rfloor$.

Since FirstFit is equivalent to an arbitrary AnyFit algorithm on a subset of instances, the lower bound holds for a general AnyFit algorithm.

3 The simplified and generalized asymptotic upper bound

We now present the simple proof of the asymptotic ratio 1.7. This proof holds for a wide class of any-fit-type algorithms: Call an AnyFit algorithm a GAAF algorithm, if it uses the bin with level at most 1/2 only when the item does not fit into any previous bin. (It is easy to verify that there is always at most one such bin, see Lemma 3.1(i).) The GAAF condition say that the AAF condition holds whenever the smallest bin has level at most 1/2; thus any AAF algorithm is a GAAF algorithm.

The asymptotic bound for AAF algorithms was proved in [13, 14]. Theorem 3.3 improves the additive term and generalizes the bound to the slightly less restrictive GAAF condition (although it seems that the original proof also uses only the GAAF condition).

Let us fix an instance I with items $a_1, \ldots, a_n$ and denote the number of bins in the BestFit and optimal solutions by BF and Opt, respectively. We will often identify an item and its size. For a set of items A, let $s(A) = \sum_{a\in A} a$, i.e., the total size of items in A and also for a set of bins $\mathcal{A}$, let $s(\mathcal{A}) = \sum_{A\in\mathcal{A}} s(A)$. Furthermore, let $S = s(I)$ be the total size of all items of I. Obviously $S \leq \text{Opt}$.

We classify the items by their sizes: items $a \leq 1/6$ are **small**, items $a \in (1/6, 1/2]$ are **medium**, and items $a > 1/2$ are **huge**.

The bins in the BF packing are ordered by the time they are opened (i.e., when the first item is packed into them). Expressions like "before", "after", "first bin", "last bin" refer to this ordering. At any time during the packing, the **level of a bin** is the total size of items currently packed in it, while by **size of a bin** we always mean its final level.

Lemma 3.1. *At any moment, in any* GAAF *packing the following holds:*

(i) *The sum of levels of any two bins is greater than 1. In particular, there is at most one bin with level at most* 1/2.

(ii) *Any item a packed into a bin with level at most* 1/2 *(i.e., a new bin or the single bin with level at most* 1/2 *guaranteed by (i)) does not fit into any bin open at the time of its arrival, except for the bin where the item a is packed.*

(iii) *If there are two bins B, B' with level at most* 2/3*, in this order, then either B' contains a single item or the first item in B' is huge.*

Proof. **(i):** The first item in any bin does not fit in any open bin by the definition of GAAF (in fact, the AnyFit condition is sufficient here), thus the sum of the levels of the two bins is greater than 1 already at the time when the second bin is opened.

(ii): If a is packed into a new bin, this follows by the GAAF (or AnyFit) condition again. Otherwise Let $x \leq 1/2$ be the level of the bin where a is packed, just before a is packed there. By (i), there is at most one bin with level at most 1/2, thus at the time of packing of a all the other bins have level strictly greater than x. By the definition of GAAF, a does not fit into any of these bins.

(iii): If B' contains two items and the first one is not huge, then by (ii) the first two items in B' do not fit into B. Thus they are larger than $1/3$ and the level of B' is greater than $2/3$. □

The short proof of the asymptotic ratio 1.7 for GAAF algorithms uses the same weight function as the traditional analysis of BestFit and FirstFit. (In some variants the weight of an item is capped to be at most 1, which makes almost no difference in the analysis.)

The weight function assigns a weight to each item, depending on its size. Intuitively, the weight measures how much space is needed to pack each item. For small items, it should be proportional to its size, but large items may generate empty space in some of the bins and their weight is accordingly larger. The overall idea is to show that the weight of each optimal bin is at most 1.7 while the average weight of each algorithm's bin is at least 1. Looking at the lower bound example with this outline in mind, it is easy to see that we should set $w(0) = 0$, $w(1/6) = 0.2$, $w(1/3) = 0.5$, and $w(1/2 + \varepsilon) \geq 1$. The weight function we define below fits these values by a piecewise linear function, with a discontinuity at $1/2$.

To use amortization, we split the weight of each item a into two parts, namely its bonus $\overline{w}(a)$ and its scaled size $\overline{\overline{w}}(a)$, defined as

$$\overline{w}(a) = \begin{cases} 0 & \text{if } a \leq \frac{1}{6}\,, \\ \frac{3}{5}(a - \frac{1}{6}) & \text{if } a \in \left(\frac{1}{6}, \frac{1}{3}\right)\,, \\ 0.1 & \text{if } a \in \left[\frac{1}{3}, \frac{1}{2}\right]\,, \\ 0.4 & \text{if } a > \frac{1}{2}\,. \end{cases}$$

For every item a we define $\overline{\overline{w}}(a) = \frac{6}{5}a$ and its weight is $w(a) = \overline{\overline{w}}(a) + \overline{w}(a)$. For a set of items B, $w(B) = \sum_{a \in B} w(a)$ denotes the total weight, similarly for $\overline{w}$ and $\overline{\overline{w}}$.

It is easy to observe that the weight of any bin B, i.e., of any set with $s(B) \leq 1$, is at most 1.7: The scaled size of B is at most 1.2, so we only need to check that $\overline{w}(B) \leq 0.5$. If B contains no huge item, there are at most 5 items with non-zero $\overline{w}(a)$ and $\overline{w}(a) \leq 0.1$ for each of them. Otherwise the huge item has bonus 0.4; there are at most two other medium items with non-zero bonus and it is easy to check that their total bonus is at most 0.1. This implies that the weight of the whole instance is at most $1.7 \cdot \text{Opt}$.

The key part is to show that, on average, the weight of each BF-bin is at least 1. Lemma 3.2 together with Lemma 3.1 implies that for almost all bins with two or more items, its scaled size plus the bonus of the **following** such bin is at least 1.

Lemma 3.2. *Let B be a bin such that $s(B) \geq 2/3$ and let c, c' be two items that do not fit into B, i.e., $c, c' > 1 - s(B)$. Then $\overline{\overline{w}}(B) + \overline{w}(c) + \overline{w}(c') \geq 1$.*

Proof. If $s(B) \geq 5/6$, then $\overline{\overline{w}}(B) \geq 1$ and we are done. Otherwise let $x = 5/6 - s(B)$. We have $0 < x \leq 1/6$ and thus $c, c' > 1/6 + x$ implies $\overline{w}(c), \overline{w}(c') > \frac{3}{5}x$. We get $\overline{\overline{w}}(B) + \overline{w}(c) + \overline{w}(c') > \frac{6}{5}(\frac{5}{6} - x) + \frac{3}{5}x + \frac{3}{5}x = 1$. □

Any GAAF-bin (i.e., a bin of the GAAF algorithm) D with a huge item has $\overline{w}(D) \geq 0.4$ and $\frac{6}{5}s(D) > 0.6$, thus $w(D) > 1$.

For the amortization, consider all GAAF-bins B with two or more items, size $s(B) \geq 2/3$, and no huge item. For any such bin except for the last one choose C as the next bin with the same properties. Since C has no huge item, its first two items c, c' have level at most $1/2$ and by Lemma 3.1(ii) they do not fit into B. Lemma 3.2 implies $\overline{\overline{w}}(B) + \overline{w}(C) \geq \overline{\overline{w}}(B) + \overline{w}(c) + \overline{w}(c') \geq 1$.

Summing all these inequalities (note that each bin is used at most once as B and at most once as C) and $w(D) > 1$ for the bins with huge items we get $w(I) \geq \mathrm{BF} - 3$. The additive constant 3 comes from the fact that we are missing an inequality for at most three BF-bins: the last one from the amortization sequence, possibly one bin B with two or more items and $s(B) < 2/3$ (cf. Lemma 3.1(iii)) and possibly one bin B with a single item and $s(B) < 1/2$ (cf. Lemma 3.1(i)). Combining this with the previous bound on the total weight, we obtain $\mathrm{GAAF} - 3 \leq w(I) \leq 1.7 \cdot \textsc{Opt}$, where GAAFdenotes the number of bins of the GAAF algorithm, and the asymptotic bound follows.

This simple proof of the asymptotic ratio can be tightened so that the additive constant is smaller. We save one of the three bins by noticing that we do not need to do amortization for bins that are after the bin of size smaller than 2/3. The remaining two bins have total size larger than 1, which brings the additive constant further down to 0.7.

Theorem 3.3. *For any* GAAF *algorithm and any instance of bin packing we have GAAF* $\leq \lfloor 1.7 \cdot \textsc{Opt} + 0.7 \rfloor$.

Proof. Any GAAF-bin D with a huge item has $\overline{w}(D) \geq 0.4$ and $\frac{6}{5}s(D) > 0.6$, thus $w(D) > 1$. Similarly, any GAAF-bin with two items larger than $1/3$ has $\overline{w}(D) \geq 0.2$ and $\frac{6}{5}s(D) > 0.8$, thus $w(D) > 1$.

For the amortization, consider all the GAAF-bins B, with (i) two or more items, (ii) no huge item, and (iii) no pair of items both larger than $1/3$. For any such bin except for the last one choose C as the next bin with the same properties. Since C has no huge item, its first two items c, c' have level at most $1/2$ and by the GAAF condition they do not fit into B. Since C has no pair of items larger than $1/3$, we have $c \leq 1/3$ or $c' \leq 1/3$ and thus $s(B) \geq 2/3$. Lemma 3.2 now implies $\overline{\overline{w}}(B) + \overline{w}(C) \geq \overline{\overline{w}}(B) + \overline{w}(c) + \overline{w}(c') \geq 1$.

Let $\overline{C}$ be the last bin used in the amortization, if it exists, and $\overline{D}$ be the single bin with $s(\overline{D}) \leq 1/2$, if it exists.

If $\overline{C}$ and $\overline{D}$ both exist, we have $s(\overline{C}) + s(\overline{D}) > 1$ by Lemma 3.1(i) and thus $\overline{\overline{w}}(\overline{C}) + w(\overline{D}) > 1.2$. Summing this, all the amortization inequalities (note that each bin is used at most once as B or $\overline{C}$ and at most once as C) and $w(D) > 1$ for the bins with huge items or two items larger than $1/3$ we get $w(I) > \mathrm{GAAF} - 0.8$. Combining this with the previous bound $w(I) \leq 1.7 \cdot \textsc{Opt}$ on the total weight, we obtain $\mathrm{GAAF} < w(I) + 0.8 \leq 1.7 \cdot \textsc{Opt} + 0.8$ and the theorem follows from the integrality of GAAF and Opt.

If $\overline{C}$ exists but $\overline{D}$ does not, we have $s(\overline{C}) > 1/2$ and thus $\overline{\overline{w}}(\overline{C}) > 0.6$. Summing this and again both all the amortization inequalities and $w(D) > 1$

for the bins with huge items or two items larger than 1/3 we get $w(I) \geq A - 0.4$ and the theorem follows again.

If $\overline{C}$ does not exist but $\overline{D}$ does, let C be an arbitrary bin other than $\overline{D}$ (if none exists, GAAF $= 1 =$ Opt and the theorem is trivial). We have $s(C) + s(\overline{D}) > 1$ and thus $w(\overline{C}) + w(\overline{D}) > 1.2$. Summing this and $w(D) > 1$ for all the remaining bins, we get $w(I) >$ GAAF $- 0.8$ and the theorem follows as above.

Finally, if neither $\overline{C}$ nor $\overline{D}$ exists, we have $w(D) > 1$ for all the GAAF-bins, thus $w(I) > A$ and the theorem follows as well. □

Next we give an example showing that GAAF algorithms do not have an absolute approximation ratio 1.7. In particular, we give an instance with Opt = 7 and GAAF packing with 12 bins.

We first describe the GAAF packing; the input sequence contains items in the order of the bins, i.e., it starts by all the items from the first bin, then continues by items from the second bin, etc. The first bin contains 6 items of size 0.12, total of 0.72. The next three bins contain each 2 items of size 0.34; note that these do not fit into any previous bin. The fifth bin has items 0.52 and 0.01; the item 0.01 fits into the previous bins, but it is packed at a level larger than 0.5, so this satisfies the GAAF condition. The sixth bin contains a single item of size 0.48 and the remaining six bins contain each an item of size 0.53; again, these items do not fit into any previous bin.

Opt contains a bin with two items of sizes 0.52 and 0.48. The remaining 6 bins contain each three items of sizes 0.53, 0.34, and 0.12, total of 0.99; in addition one of them contains also the item 0.01. This packs all the items in the 7 bins and completes the example.

Thus removing the additive constant completely is impossible for GAAF algorithms, and thus it needs to use additional properties of the algorithm. We are able to do this for FirstFit and BestFit, using additional ideas [7, 8]. It remains an interesting open problem whether this is possible for all AAF algorithms.

4 Bounded space algorithms

The lower bound constructions in [16] is a variant of the FirstFit lower bound which we have presented above. In its first approximation, giving a lower bound of $5/3 \approx 1.666$, we give the algorithm the same number of items of sizes $1/7 + \varepsilon$, $1/3 + \varepsilon$, and $1/2 + \varepsilon$. The optimal packing uses almost full bins with one item of each size. The full construction repeatedly prepends this sequence by another group of same-size items, the size is set to $1/t + \varepsilon$ for the smallest integer so that this item fits into the optimal bin (for a tiny $\varepsilon > 0$). On any such sequence, **any** k-bounded-space algorithm uses bins grouping items of the same size, with a constant number of exceptions. This shows that every bounded-space algorithm has asymptotic worst case ratio at least $h_\infty \approx 1.69103$; the number h_∞ is defined from the sequence of reciprocals $1/2, 1/3, 1/7, 1/43, \ldots$, which we described above.

The optimal packing above can be achieved only by an algorithm which keeps many bins open even if the whole sequence is known in advance. Or, from another viewpoint, we are allowed the optimal algorithm to reorder sequence. It is natural to ask how much inefficiency is caused by not knowing the future, i.e., by the fact that the algorithm is online. For this purpose, we restrict the offline algorithms so that they can also have only k bins open. For a given instance I, let $\textsc{Opt}_k = \textsc{Opt}_k(I)$ be the smallest possible number of bins used in a packing produced by a k-bounded-space bin packing algorithm that know the whole instance I before it starts serving it. We say that the algorithm A it is an **asymptotic R-approximation** algorithm **w.r.t. the k-bounded-space optimum** if for every instance I

$$A \leq R \cdot \textsc{Opt}_k + f(\textsc{Opt}_k),$$

where $f(n) \in o(n)$; again, typically $f(n)$ is a global constant. The smallest such R is called the **k-bounded-space ratio** of algorithm A in [2], where it was introduced.

On the lower-bound instances described above, BBF_k is an asymptotic 1-approximation w.r.t. the k-bounded-space optimum, so there is no inefficiency caused by the online environment. It is not hard to construct instances showing that the 2-bounded-space ratio of BBF_2 is at least 1.5, see [2]. However, the best upper bound on the 2-bounded-space ratio of BBF_2 or any other bounded-space algorithm is 1.7 and closing this gap is an interesting question. Levin and Epstein (private communication) have shown that the 2-bounded-space ratio of any online algorithm is at least 1.5, which leaves open the possibility that BBF_2 is the best possible algorithm. The case of $k > 2$ is widely open.

One of the difficulties of studying the k-bounded-space ratio is that it seems non-trivial to understand $\textsc{Opt}_k$. Unlike $\textsc{Opt}$, for which we have an asymptotic approximation scheme, $\textsc{Opt}_k$ is hard to approximate. In [2], it is shown that the 2-bounded-space ratio of any polynomial algorithm is at least $5/4$. There exists a $(3/2 + \varepsilon)$-approximation algorithm for $\textsc{Opt}_k$, based on a partial enumeration of the solutions [2]. We are missing a deeper insight in the structure of the solutions and also nothing is known for $k > 3$.

Acknowledgements

Partially supported by the Center of Excellence – Inst. for Theor. Comp. Sci., Prague (project P202/12/G061 of GA ČR).

References

1. J. Boyar, G. Dósa, and L. Epstein. On the absolute approximation ratio for First Fit and related results. *Discrete Appl. Math.*, 160:1914–1923, 2012.
2. M. Chrobak, J. Sgall, and G. J. Woeginger. Two-bounded-space bin packing revisited. In *Proc. of the 19th Ann. European Symp. on Algorithms (ESA), Lecture Notes in Comput. Sci. 6942*, pages 263–274. Springer, 2011.

3. E. G. Coffman, M. R. Garey, and D. S. Johnson. Approximation algorithms for bin packing: A survey. In D. Hochbaum, editor, *Approximation algorithms.* PWS Publishing Company, 1997.
4. E. G. Coffman, J. Csirik, G. J. Woeginger, Approximate solutions to bin packing problems. In *Handbook of Applied Optimization*, P.M. Pardalos and M.G.C. Resende (eds.), Oxford University Press, New York, 607–615, 2002.
5. J. Csirik and D.S. Johnson. Bounded space on-line bin packing: Best is better than first. *Algorithmica*, 31:115–138, 2001.
6. G. Dósa. The tight bound of First Fit Decreasing bin-packing algorithm is $FFD(I) \leq 11/9\, OPT(I) + 6/9$. In *Proc. 1st International Symp. on Combinatorics, Algorithms, Probabilistic and Experimental Methodologies (ESCAPE)*, volume 4614 of *Lecture Notes in Comput. Sci.*, pages 1–11. Springer, 2007.
7. G. Dósa and J. Sgall. First Fit bin packing: A tight analysis. In *Proc. of the 30th Ann. Symp. on Theor. Aspects of Comput. Sci. (STACS), LIPIcs vol. 3*, pages 538–549. Schloss Dagstuhl, 2013.
8. G. Dósa and J. Sgall. Optimal analysis of Best Fit bin packing. Submitted, 2013.
9. M. R. Garey, R. L. Graham, D. S. Johnson, and A. C.-C. Yao. Resource constrained scheduling as generalized bin packing. *J. Combin. Theory Ser. A*, 21:257–298, 1976.
10. M. R. Garey, R. L. Graham, and J. D. Ullman. Worst-case analysis of memory allocation algorithms. In *Proc. 4th Symp. Theory of Computing (STOC)*, pages 143–150. ACM, 1973.
11. R. L. Graham. Bounds for certain multiprocessing anomalies. *Bell System Technical J.*, 45:1563–1581, 1966.
12. R. L. Graham. Bounds on multiprocessing timing anomalies. *SIAM J. Appl. Math.*, 17:263–269, 1969.
13. D. S. Johnson. *Near-optimal bin packing algorithms.* PhD thesis, MIT, Cambridge, MA, 1973.
14. D. S. Johnson. Fast algorithms for bin packing. *J. Comput. Syst. Sci.*, 8:272–314, 1974.
15. D. S. Johnson, A. Demers, J. D. Ullman, M. R. Garey, and R. L. Graham. Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM J. Comput.*, 3:256–278, 1974.
16. C. C. Lee and D. T. Lee. A simple on-line bin-packing algorithm. *Journal of the ACM*, 32:562–572, 1985.
17. Z. Németh. A first fit algoritmus abszolút hibájáról (in Hungarian). Eötvös Loránd Univ., Budapest, Hungary, 2011.
18. J. Sgall. A new analysis of Best Fit bin packing. In *Proc. of 6th Int. Conference FUN with Algorithms*, volume 7288 of *Lecture Notes in Comput. Sci.*, pages 315–321. Springer, 2012.
19. D. Simchi-Levi. New worst case results for the bin-packing problem. *Naval Research Logistics*, 41:579–585, 1994.
20. J. D. Ullman. The performance of a memory allocation algorithm. Technical Report 100, Princeton Univ., Princeton, NJ, 1971.
21. D. P. Williamson and D. B. Shmoys. *The Design of Approximation Algorithms.* Cambridge University Press, 2011.
22. B. Xia and Z. Tan. Tighter bounds of the First Fit algorithm for the bin-packing problem. *Discrete Appl. Math.*, 158:1668–1675, 2010.