A new analysis of Best Fit bin packing

Jiří Sgall[1]

Computer Science Institute of Charles University, Prague, Czech Republic.
sgall@iuuk.mff.cuni.cz

Abstract. We give a simple proof and a generalization of the classical result which says that the (asymptotic) approximation ratio of BestFit algorithm is 1.7. We generalize this result to a wide class of algorithms that are allowed to pack the incoming item to any bin with load larger than 1/2 (if it fits), instead to the most full bin, and at the same time this class includes the bounded-space variants of these algorithms.

1 Introduction

An instance of the classical bin packing problem consists of a sequence a_1, a_2, $\ldots$, a_n 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. Bin packing is a fundamental problem in combinatorial optimization, and it has been studied extensively since the early 1970s. Since bin packing is NP-hard, one active branch of research has concentrated on approximation algorithms that find near-optimal packings; see for instance [1, 2] for surveys.

In this note we give a simple proof and a generalization of the classical result which says that the (asymptotic) approximation ratio of BestFit algorithm is 1.7. We generalize this result to a wide class of algorithms that are allowed to pack the incoming item to any bin with load larger than 1/2 (if it fits), instead to the most full bin as in BestFit. The analysis covers also the bounded space variants of all these algorithms, including the bounded-space variant of BestFit called BBF_k analyzed by Csirik and Johnson [3].

The analysis uses a combination of a weight technique, used in the classical proofs in the area of bin packing, and amortized analysis. The amortization reduces the proof to simple local considerations, in place of a relatively complicated analysis of possible global configurations in the previous proofs.

Intuitively, our amortization can be described as a group management technique that may resemble some procedures in companies, political parties, or even our academic departments. Over their lifetime, individuals (the bins) receive certain credit (the items). The goal of the game is to retire all individuals (close the bins) with a sufficient credit. In the group, there is always one senior and possibly many juniors. Juniors retire with the credit they have collected. However, when a senior retires, he chooses one junior as his successor and acquires a certain part of his credit. In return, this junior becomes the new senior (and later may acquire some credit from another junior). Unlike in recently popular Ponzi or pyramid schemes, each senior collects credit of only one junior. Thus we end up with only one or two individuals with a low credit (instead of a crowd), and this is sufficient for the claimed result.

Related results

The asymptotic approximation ratio of 1.7 was given by Johnson et al. [5], together with examples showing that this bound is asymptotically tight. The additive constant was later tightened by Garey et al. [4] to $\lceil 1.7 \cdot OPT(I) \rceil$. We further tighten the constant to $1.7 \cdot OPT(I) + 0.7$.

Csirik and Johnson [3] show that the 2-space-bounded variant Best Fit algorithm BBF_2 has the same asymptotic worst case ratio of 1.7. We follow the general outline of their proof and a general treatment of algorithms that include both BestFit and BBF_2.

For more results on bounded-space algorithms, we mention Lee and Lee [6] who design a k-bounded-space online bin packing algorithms whose asymptotic ratio comes 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 bin packing algorithm A has asymptotic approximation ratio at least h_∞. A more space-efficient algorithm with the same performance was given by Woeginger [7].

2 The class of GoodFit algorithms

We analyze a class of algorithms which we call GoodFit algorithms. This extends the approach of Csirik and Johnson [3], who formulated a class of algorithms containing BestFit together with its k-bounded-space versions BBF_k.

Our main generalization concerns the packing rule. While the BestFit packing rule requires to pack the item into the fullest bin among those where it fits, we allow packing into any open bin if the item fits; the only restriction is that we prefer bins that are more than half-full. This forbids at most one bin, so the restriction is very mild. The exact formulation of the packing rule is this:

GoodFit **packing rule:** The new item is packed as follows:
(i) If possible, it is packed into an arbitrary bin of size more than $\frac{1}{2}$ where it fits;
(ii) if there is no such bin, it is packed into a bin of size smaller than $\frac{1}{2}$ if such a bin exists and the item fits;
(iii) else it is packed into a newly opened bin.

Note that the packing rule indeed implies that at each time, there is at most one bin of size at most $\frac{1}{2}$. Since at most one bin is forbidden for packing, our class of algorithms includes previously studied heuristics such as AlmostWorstFit (if there are more possible bins to pack an item, choose the second smallest one) and the class AlmostAnyFit (if there are more possible bins to pack an item, choose any bin except the smallest one).

Similarly to Csirik and Johnson [3], we accommodate also the analysis of bounded space algorithms. For this, following their approach, we separate closing of the bins from the packing steps. For the purpose of analysis, it is convenient to perform first the packing step possibly followed by a closing step (which is then not allowed to close the just open bin). Again, as in [3], our analysis works for algorithms that close bins at any time, with the only restriction that two bins remain open at all times after opening the first two bins. I.e., the closing step is only allowed when three or more bins are open. This includes k-bounded versions of the algorithms mentioned above for any $k \geq 2$. The introduction of the closing rule is also convenient for the analysis of standard algorithms like BestFit that never close a bin: We simply let them to close (almost all) bins at the end.

The BestFit closing rule requires to close the largest open bin, excluding the bin that was open by the last item (if there is such). Every algorithm can be modified to use this rule without any loss of performance. However, our analysis holds for any algorithm that closes bins according to following relaxed rule:

GoodFit **closing rule:** If there are at least three open bins, the algorithm can close one of the open bins, excluding the bin just opened by a newly arrived item (if there is such), and satisfying:
(i) either its size is at least $\frac{5}{6}$;
(ii) or, if there is no bin satisfying (i), it is the largest bin of size more than $\frac{2}{3}$;
(iii) or else, if there is no bin satisfying (i) or (ii), it has size more than $\frac{1}{2}$.

Since the packing rule guarantees there is at most one bin of size at most $\frac{1}{2}$, at most one bin is the newly opened one, and there are at least three open bins when closing a bin, a bin may be always chosen and the rule is well-defined. In particular, the closed bin has always size at least $\frac{1}{2}$.

An algorithm is a GoodFit algorithm if it follows the GoodFit packing and closing rules. We prove that any GoodFit algorithm has asymptotic approximation ratio at most 1.7. For the rest of the analysis we fix some GoodFit algorithm A.

3 Seniors, juniors and their credits

Items are denoted a_t, in the order of their arrivals; a_t also denotes the size of each item. The bins are denoted B_i, indexed in the order in which they have been opened. For any bin or other set of items B, let $s(B)$ be the sum of the sizes of the items in B.

We define one of the open bins to be the *senior* bin, the other bins are *junior*. Usually the oldest bin (the one with the lowest index) is the senior one, sometimes the second oldest one. Initially, after packing the first item, B_1 is the senior bin. When the senior bin is closed, the new senior bin is chosen according to the following rule: If the oldest open bin has a single item and its size is at most $\frac{1}{2}$, choose the second oldest open bin to be the senior one. Otherwise choose the oldest open bin as the senior one. Note that there are always two open bins remaining, so this is well-defined. Also, if the senior bin is not the oldest one, then it has load more than $\frac{1}{2}$, as there is at most one bin with load at most $\frac{1}{2}$. The current senior bin is denoted B_s.

We first prove an important property of junior bins. Similarly to real life, the smaller the senior bin is, the more stringent are the requirements on the junior bins.

Lemma 3.1. *Any junior bin B_i contains either*
(i) an item $a > \frac{1}{2}$ (and possibly some other items); or
(ii) two items $a, a' > 1 - s(B_s)$ (and possibly some other items); or
(iii) an item $a > 1 - s(B_s)$ and no other items.

Proof. By induction. After packing the first item, the lemma is trivially true, as there is no junior bin. After closing a bin other than the senior bin, the lemma clearly continues to hold. Suppose that the senior bin has just been closed and the new senior bin is chosen to be B_s.

If B_i is older than B_s, it must be the case that B_i is the oldest bin and it contains a single item a, due to the rules for choosing the senior bin. But then the first item assigned to B_s is larger than $1 - a$, thus (iii) applies to B_i.

Otherwise B_s is older than B_i. Thus the first item a in B_i did not fit into B_s at the time when it was packed, consequently $a > 1 - s(B_s)$ also now. If $a > \frac{1}{2}$, (i) applies. If $a \le \frac{1}{2}$ is the single item in B_i, (iii) applies. Otherwise the size of B_s was more than $\frac{1}{2}$ already when a was packed. Consider the second item a' packed into B_i. As at that moment B_s had size more than $\frac{1}{2}$ and B_i at most $\frac{1}{2}$, thus a' can be packed into B_i only if it does not fit into B_s. It follows that (ii) applies.

It remains to verify that the lemma holds after packing a new item. If it is packed into B_s, all the conditions continue to hold. If an item a is packed into B_i, then similarly to the previous paragraph the only non-trivial case is when it is the second item packed into a bin smaller than $\frac{1}{2}$ and then (ii) applies afterwards. □

We note that Lemma 3.1 is the only part of the proof that uses the definition of the GoodFit packing rule.

Finally, we define the weight (credit) function. We define a quantity $b(a)$, a bonus of an item as follows:

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

Note that $b(a)$ has a jump at $\frac{1}{2}$ and is continuous elsewhere. We define a weight (or a credit) of an item to be

$$w(a) = \frac{6}{5}a + b(a).$$

For a set of items or a bin B, let $w(B)$ and $b(B)$ denote the sum of the weights and bonuses of all the items in B, respectively. The bonus of a junior bin $b(B)$ represents exactly the part of credit that may be acquired by a senior bin upon its closing.

The amortized analysis

The easy part is to show that each bin in the optimal schedule has weight at most 1.7; this part is known already from the previous papers [3–5] and we provide a simplified proof for completeness in Lemma 3.3.

The main ingredient is to show that on average each bin closed by GoodFit has weight at least 1, more precisely that the number of bins used by the algorithm is at most $\lceil w(I) \rceil$. The crucial part is Lemma 3.2 which shows that the amortized weight of each closed bin is at least 1, using the amortization described intuitively above.

Suppose that we are closing a bin B_i. If $B_i = B_s$ is the senior bin, let B_j be the newly chosen senior bin; otherwise let $B_j = B_i$. I.e., B_j is always the (currently) junior bin whose bonus we are using, and B_i is one of the two distinct bins B_s and B_j. The weight (credit) available upon closing B_i is exactly $\frac{6}{5}s(B_i) + b(B_j)$; this is formalized by a potential later in proof of Theorem 3.4. The key lemma is thus this:

Lemma 3.2. *Using the notation defined above, upon closing bin B_i we have $\frac{6}{5}s(B_i) + b(B_j) \geq 1$.*

Proof. We distinguish three cases.

Case 1: $s(B_i) \geq \frac{5}{6}$. Then $\frac{6}{5}s(B_i) \geq 1$ and we are done.

Case 2: $b(B_j) \geq 0.4$. Since the closed bin always has size more than $\frac{1}{2}$, we have $\frac{6}{5}s(B_i) + b(B_j) > 0.6 + 0.4 = 1$.

Case 3: Otherwise we prove that $s(B_i) > \frac{2}{3}$ and at the same time B_j contains two items with a sufficient bonus. First we apply Lemma 3.1 to the junior bin B_j and claim that the case (ii) of Lemma 3.1 must hold. The bin B_j does not contain an item of size larger than $\frac{1}{2}$, as then $b(B_j) \geq 0.4$ and this is covered by Case 2 above. Thus the case (i) of Lemma 3.1 cannot hold for B_j. The case (iii) of Lemma 3.1 cannot hold for B_j, since either $B_j = B_i$ and then B_j as the closed bin has size more than $\frac{1}{2}$, or else B_j is the new senior bin and as such it does not have single item of size at most $\frac{1}{2}$. Thus the case (ii) of the lemma holds and B_j contains two items $a, a' > 1 - s(B_s)$.

Next we claim that $s(B_i) > \frac{2}{3}$ and $s(B_i) \geq s(B_s)$. If $s(B_s) \leq \frac{2}{3}$ then $a, a' > \frac{1}{3}$ and $s(B_j) > \frac{2}{3}$; by the GoodFit closing rule it has to be the case that B_j was closed rather than B_s, thus $B_i = B_j$, and $s(B_j) > \frac{2}{3} \geq s(B_s)$; the claim follows. If $s(B_s) > \frac{2}{3}$ then by the GoodFit closing rule the largest bin is closed (using also the fact that there is no bin of size at least $\frac{5}{6}$, as Case 1 does not occur), thus $B_i \geq B_s > \frac{2}{3}$ and the claim also follows.

Now let $\alpha = \frac{5}{6} - s(B_i)$; using $s(B_i) > \frac{2}{3}$ we have $\alpha < \frac{1}{6}$. Furthermore, $a, a' > 1 - s(B_s) \geq 1 - s(B_i) = \frac{1}{6} + \alpha$. Thus $b(a), b(a') > \frac{3}{5}\alpha$ and $\frac{6}{5}s(B_i) + b(B_j) > (1 - \frac{6}{5}\alpha) + 2 \cdot \frac{3}{5}\alpha = 1$. □

Lemma 3.3. *For any bin B, i.e., any set B with $s(B) \leq 1$, we have $w(B) \leq 1.7$.*

Proof. It is sufficient to prove that $b(B) \leq 0.5$, as $\frac{6}{5}s(B) \leq 1.2$. Any item with non-zero bonus has size larger than $\frac{1}{6}$, thus each bin contains at most 5 of them. If all items have bonus at most 0.1, we are done. Otherwise there is an item of size larger than $\frac{1}{2}$ and there can be at most two other items with non-zero bonus. If their sizes are $\frac{1}{6} + \alpha$ and $\frac{1}{6} + \beta$ then $\alpha + \beta < \frac{1}{6}$, their bonus is at most $\frac{3}{5}(\alpha + \beta) < 0.1$ and $b(B) < 0.5$, including the bonus 0.4 of the large item. □

Lemma 3.3 is the only place that uses the definition of the GoodFit closing rule.

Theorem 3.4. *Let A be any* GoodFit *algorithm. For any instance I, we have $A(I) \leq \lfloor 1.7 \cdot OPT(I) + 0.7 \rfloor \leq \lceil 1.7 \cdot OPT(I) \rceil$, where $A(I)$ denotes the number of bins used by A and $OPT(I)$ denotes the optimal (minimal) number of bins.*

Proof. We define a potential $\Phi = \sum_B w(B) - b(B_s)$, where the sum is over all currently open bins. Initially $\Phi = 0$. When a new item a is packed, the potential increases by at most $w(a)$, thus during the whole instance the total increase is at most $w(I)$. Lemma 3.2 shows that upon closing a bin, Φ decreases by at least 1. If the algorithm never opens the second bin, the result is trivial. Otherwise, at the end, we close all but two bins using the GoodFit closing rule, if the algorithm has not done so. At this point, the potential of the algorithm is more than 1.2, as the sum of the sizes of the two open bins is larger than 1. So the number of bins used by the algorithm is bounded by $A(I) < 2 + (w(I) - 1.2) = w(I) + 0.8$. Lemma 3.3 implies that $w(I) \leq 1.7 \cdot OPT(I)$; since $A(I)$ and $OPT(I)$ are integers, this implies $A(I) \leq 1.7 \cdot OPT(I) + 0.7$ and the theorem follows. □

Acknowledgements

Partially supported by Inst. for Theor. Comp. Sci., Prague (project P202/12/G061 of GA ČR) and project IAA100190902 of GA AV ČR. I'm grateful to anonymous referees for useful comments.

References

1. E.G. Coffman, Jr., J. Csirik, G.J. Woeginger (2002). 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.
2. E.G. Coffman, Jr., M.R. Garey and D.S. Johnson (1996). Approximation algorithms for bin packing: A survey. In Approximation Algorithms for NP-Hard Problems, D. Hochbaum (ed.), PWS Publishing, Boston. Chapter 2, 46-93

3. J. Csirik and D.S. Johnson (2001). Bounded space on-line bin packing: Best is better than first. *Algorithmica 31*, 115–138.
4. M.R. Garey, R. L. Graham, D.S. Johnson, and A. C. Yao (1976). Resource constrained scheduling as generalized bin packing. *Journal of Combinatorial Theory Series A 21*, 257–298.
5. D. S. Johnson, A. Demers, J. D. Ullman, M. R. Garey and R. L. Graham (1974). Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM Journal on Computing 3*, 299–325.
6. C.C. Lee and D.T. Lee (1985). A simple on-line bin-packing algorithm. *Journal of the ACM 32*, 562–572.
7. G.J. Woeginger (1993). Improved space for bounded-space, on-line bin-packing. *SIAM Journal on Discrete Mathematics 6*, 575–581.