New Worst-Case Results for the Bin-Packing Problem

David Simchi-Levi
Department of Industrial Engineering and Operations Research, Columbia University, New York, New York 10027
and
Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois 60208-3119

In this note we consider the familiar bin-packing problem and provide new worst-case results for a number of classical heuristics. We show that the first-fit and best-fit heuristics have an absolute performance ratio of no more than 1.75, and first-fit decreasing and best-fit decreasing heuristics have an absolute performance ratio of 1.5. The latter is the best possible absolute performance ratio for the bin-packing problem, unless $P = NP$.

1. INTRODUCTION

The bin-packing problem can be stated as follows: given a list of n real numbers $L = (w_1, w_2, \ldots, w_n)$, where we call $w_i \in (0, 1]$ the size of item i, the problem is to assign each item to a bin such that the sum of the item sizes in a bin does not exceed 1, while minimizing the number of bins used. For simplicity, we also use L as a set, but this should cause no confusion. In this case, we write $i \in L$ to mean $w_i \in L$.

The bin-packing problem belongs to the class of NP-hard problems, and therefore the existence of a polynomial time algorithm to solve the problem optimally is unlikely. As a result, since the early 1970s much research has been conducted on heuristics that solve the problem to near optimality. An excellent survey of the research on this problem is available in Coffman, Garey, and Johnson [2].

This note is concerned with the performance of the first-fit (FF), best-fit (BF), first-fit decreasing (FFD), and best-fit decreasing (BFD) heuristics developed and analyzed by Johnson et al. [5]. The heuristic FF assigns items to bins according to the order they appear in the list without using any knowledge of subsequent items in the list. It can be described as follows: Place item 1 in bin 1. Suppose we are packing item j, place item j in the lowest indexed bin whose current content does not exceed $1 - w_j$. The BF heuristic is similar to FF except that it places item j in the bin whose current content is the largest but does not exceed $1 - w_j$. In contrast to these heuristics, FFD first sorts the items in nonincreasing order of their size and then performs FF. Similarly, BFD first sorts the items in nonincreasing order of their size and then performs BF.

Let $b^H(L)$ be the number of bins produced by a heuristic H on list L. Similarly, let $b^*(L)$ be the minimum number of bins required to pack the items in list L; that is, $b^*(L)$ is the optimal solution to the bin-packing problem defined on list L.

Naval Research Logistics, Vol. 41, 579–585 (1994)

CCC 0894-069X/94/040579-07

The best bounds on the performance of FF and BF heuristics are given in Garey, Graham, Johnson, and Yao [3], where they show that

$$b^{\mathrm{FF}}(L) \leq \left\lceil \left(\frac{17}{10}\, b^*(L)\right\rceil,$$

and

$$b^{\mathrm{BF}}(L) \leq \left\lceil \frac{17}{10}\, b^*(L)\right\rceil.$$

Currently, the best bounds on the performance of FFD and BFD heuristics have been obtained by Baker [1], who shows that

$$\frac{b^{\mathrm{FFD}}(L)}{b^*(L)} \leq \frac{11}{9} + \frac{3}{b^*(L)}$$

and

$$\frac{b^{\mathrm{BFD}}(L)}{b^*(L)} \leq \frac{11}{9} + \frac{3}{b^*(L)}.$$

Johnson et al. [5] provide examples of instances with arbitrarily large values of $b^*(L)$ such that the ratios $b^{\mathrm{FF}}(L)/b^*(L)$ and $b^{\mathrm{BF}}(L)/b^*(L)$ approach $\frac{17}{10}$, and $b^{\mathrm{FFD}}(L)/b^*(L)$ and $b^{\mathrm{BFD}}(L)/b^*(L)$ approach $\frac{11}{9}$. Thus, the *asymptotic performance ratio* of the FF and BF heuristics is $\frac{17}{10}$, and FFD and BFD heuristics have an asymptotic performance ratio of $\frac{11}{9}$. That is, the maximum deviation from optimality for all lists that are sufficiently large is no more than $\frac{7}{10}$ times the minimal number of bins in the case of FF and BF, and $\frac{2}{9}$ in the case of FFD and BFD.

In this article, however, we are interested in the so-called *absolute performance ratio* of these heuristics. This performance measure is defined as follows: For any heuristic H, the absolute performance ratio R^H is given by

$$R^H \equiv \inf\left\{ r \geq 1 \,\middle|\, \frac{b^H(L)}{b^*(L)} \leq r, \text{ for all lists } L \right\}.$$

Thus, the absolute performance ratio for a heuristic H gives, for all possible lists, the heuristic solution's maximum deviation from optimality.

Let XF be either FF or BF and let XFD be either FFD or BFD. We prove the following.

THEOREM 1.1: For all lists L,

$$\frac{b^{\mathrm{XF}}(L)}{b^*(L)} \leq 1.75$$

and

$$\frac{b^{\mathrm{XFD}}(L)}{b^*(L)} \leq 1.5.$$

Garey and Johnson [4, p. 128] point out that it is easy to construct examples in which an optimal solution uses 2 bins, and FFD uses 3 bins. The same holds for BFD. Similarly, Johnson et al. give examples in which an optimal solution uses 10 bins while FF and BF use 17 bins. Thus, the absolute performance ratio for FFD and BFD is exactly 1.5, and it is at least 1.7 but no more than 1.75 for FF and BF.

Observe that these new worst-case results are better than those already known for instances of the bin-packing problem in which the optimal number of bins is not too large. That is, the new worst-case bounds for FF and BF are tighter when the number of bins in an optimal solution is no more than 13, and they are sharper in the case of FFD and BFD when the optimal number of bins is no more than 9. In all of the above cases, i.e., even when $b^*(L)$ is small, the bin-packing problem is still difficult. This is true because the 2-partition problem, which is known to be NP-complete (see [4]) can be polynomially reduced to the problem of deciding whether or not it is possible to pack all the items in two bins. This also implies that no polynomial-time heuristic has an absolute performance ratio smaller than 1.5 for the bin-packing problem, unless $P = NP$. This is obvious, because such a heuristic could be used to solve the 2-partition in polynomial time. Thus, we conclude that both the FFD and BFD heuristics have the best possible absolute performance ratios for the bin-packing problem, among all polynomial-time heuristics.

We now define the following terms which will be used throughout the article: An item is called *large* (*small*) if its size is greater than $\frac{1}{2}$ (no more than $\frac{1}{2}$). Similarly, define a bin to be of Type I if it has only small items, and a bin is of Type II if it is not a Type I bin; that is, it has at least one large item in it.

Finally, we recall some definitions used throughout the bin-packing literature. Call a bin *feasible* if the sum of the item sizes in the bin does not exceed 1. An item is said to *fit* in a bin if the bin resulting from the insertion of this item is a feasible bin. In addition, a bin is said to be *opened* when an item is placed in a bin that was previously empty.

The article is organized as follows. In Section 2, we prove the worst-case results for FF and BF, and in Section 3 we establish the absolute performance ratios for FFD and BFD.

2. FIRST-FIT AND BEST-FIT

The proof of the worst-case bounds for FF and BF heuristics is based on the following observation.

PROPERTY 2.1: Consider the jth bin, $j \geq 2$, opened by XF. Any item that was assigned to it before it was more than half full does not fit in any bin opened by XF prior to bin j.

PROOF: The property is clearly true for FF (and in fact holds for any item assigned to the jth bin, $j \geq 2$, not necessarily to items assigned to it before it was more than half full). To prove the property for BF, suppose, by contradiction, item i was assigned to the jth bin before it was more than half full, and this item fits in one of the previously opened bins, say the kth bin. Clearly, in that case i cannot be the first item assigned to the jth bin, because BF would not have opened a new bin if i fits in one of the previously opened bins. Let the levels of bins k and j, just before the time item i was packed, be L_k and L_j and let item h be the first item in bin j. Hence, $w_h < L_j < 0.5$ by hypothesis. Because BF assigns an item to the bin where it fits with the largest content, and item i

would have fit in bin k, we have $L_j > L_k$. Thus, $L_k < 0.5$ meaning that item h would have fit in bin k, a contradiction. ■

A similar property is used in Johnson et al. [5] for their proof of the asymptotic performance ratio of BF and FF. We use Property 2.1 to construct a lower bound on the optimal number of bins. For this purpose, we introduce the following procedure. For a given integer v, $2 \le v \le b^{\mathrm{XF}}(L)$, select v bins from those produced by XF. Index the v bins in the order they are opened starting with index 1 and ending with index v. Let X_j be the set of items assigned by XF to the jth bin before it was more than half full, $j = 1, 2, \ldots, v$. Let S_j be the set of items assigned by XF to the jth bin, $j = 1, 2, \ldots, v$. Observe that $X_j \subseteq S_j$ for all $j = 1, 2, \ldots, v$.

PROCEDURE A:

Step 1: Let $X_i' = X_i$, $i = 1, 2, \ldots, v$.
Step 2: For $i = 1$ to $v - 1$ do begin
  Let X_j' be the nonempty set X_k' with the highest index.
  If $j = i$ Stop.
  Else, let u be the smallest item in X_j'.
    Set $S_i \leftarrow S_i \cup \{u\}$,
    and $X_j' \leftarrow X_j' \backslash \{u\}$.
  end.

In view of Property 2.1 it is clear that Procedure A generates nonempty subsets $S_1, S_2, \ldots, S_m$, for some $m \le v$, such that $\sum_{i \in S_j} w_i > 1$ for $j = 1, 2, \ldots, m - 1$ and possibly for $j = m$. This is true because (using the definition of u, i, and j in Procedure A) by Property 2.1 item u, originally assigned to bin j before it was more than half full, does not fit in any bin i with $i < j$.

PROPERTY 2.2:

$$\max\left\{ \left| \bigcup_{j=m+1}^{v} X_j \right|, m - 1 \right\} < \sum_{j=1}^{v} \sum_{i \in S_j} w_i.$$

PROOF: Because bins $1, 2, \ldots, m - 1$ generated by Procedure A are not feasible, we have $\sum_{j=1}^{v} \sum_{i \in S_j} w_i > m - 1$. Note that every item in $\bigcup_{j=m+1}^{v} X_j$ is moved by Procedure A to exactly one S_j, $j = 1, 2, \ldots, m - 1$ and possibly to S_m. Thus, if S_m is feasible, that is, no (additional) item is assigned by Procedure A to S_m, then $|\bigcup_{j=m+1}^{v} X_j| \le m - 1 < \sum_{j=1}^{v} \sum_{i \in S_j} w_i$. If, on the other hand, an item is assigned by Procedure A to S_m, then all the subsets S_j, $j = 1, 2, \ldots, m$, are not feasible, and therefore $m = |\bigcup_{j=m+1}^{v} X_j| < \sum_{j=1}^{v} \sum_{i \in S_j} w_i$. ■

We are now ready to prove our result on the absolute performance ratio of the XF heuristic (with XF being either FF or BF). Let c be the number of large items in the list L. Without loss of generality, assume $b^{\mathrm{XF}}(L) > c$, because otherwise the solution produced by XF is optimal. So, $b^{\mathrm{XF}}(L) - c > 0$ is the number of Type I bins produced by XF. We consider the two cases of c being even or odd.

CASE 1 (c is even): In this case we partition the bins produced by XF into two sets. The first set includes only Type I bins, and the second set includes the remaining bins produced by XF, that is, all the Type II bins. Index the bins in the first set in the order they are opened, from 1 to $b^{XF}(L) - c$. Let $v = b^{XF}(L) - c$, and apply Procedure A to the set of Type I bins, producing m bins out of which at least $m - 1$ are not feasible.

PROPERTY 2.3: If c is even, then

$$\max\left\{\frac{c}{2} + m, 2(b^{XF}(L) - m) - \frac{3c}{2}\right\} \leq b^*(L).$$

PROOF: Combining Property 2.2 with the fact that no two large items fit in the same bin, we have $\sum_{i\in L} w_i > m - 1 + c/2$. On the other hand, every bin in an optimal solution is feasible, and therefore $\sum_{i\in L} w_i \leq b^*(L)$. Hence, $m + c/2 \leq b^*(L)$. Because we applied Procedure A only to the Type I bins produced by XF, each one of these bins has at least two items, except possibly one, which may have only one item. Hence, $2(b^{XF}(L) - m - c - 1) + 1 \leq |\cup_{j=m+1}^{v} X_j|$ and therefore, using Property 2.2,

$$2(b^{XF}(L) - m - c - 1) + c/2 + 1 < \sum_{i\in L} w_i \leq b^*(L),$$

or

$$2(b^{XF}(L) - m - c - 1) + c/2 + 2 \leq b^*(L).$$

Rearranging the left-hand side gives the second lower bound. ■

THEOREM 2.4: If c is even, then

$$b^{XF}(L) \leq 1.75b^*(L).$$

PROOF: From Property 2.3 we have $2(b^{XF}(L) - m) - 3c/2 \leq b^*(L)$. Hence,

$$\begin{aligned} b^{XF}(L) &\leq \frac{b^*(L)}{2} + \frac{3c}{4} + m \\ &= \frac{b^*(L)}{2} + \left(m + \frac{c}{2}\right) + \frac{c}{4} \\ &\leq 1.75b^*(L), \end{aligned}$$

because $m + c/2$ is a lower bound (Property 2.3) on $b^*(L)$ and c is also a lower bound. ■

CASE 2: (c is odd): In this case we partition the set of all bins generated by the XF heuristic in a slightly different way. The first set of bins, called B_1, comprises all the Type I bins except the last Type I bin opened by XF. The second set is made up of the remaining bins, that is, these are all the Type II bins together with the Type I bin not included in B_1. We now apply procedure A to the bins in B_1 (with $v = b^{XF}(L) - c - 1$), producing m bins, out of which at least $m - 1$ bins are not feasible.

PROPERTY 2.5: If c is odd, then

$$\max\left\{\frac{c}{2} + m + \frac{1}{2}, 2(b^{\mathrm{XF}}(L) - m) - \frac{3c}{2} - \frac{1}{2}\right\} \le b^*(L).$$

PROOF: Take one of the Type II bins and match it with the only Type I bin not in B_1; the total weight of these two bins is more than 1. Thus, using Property 2.2, we have $(c - 1)/2 + 1 + (m - 1) < \Sigma_{i \in L} w_i \le b^*(L)$ which proves the first lower bound. To prove the second lower bound, we use the fact that every bin in B_1 has at least two items and therefore $2(b^{\mathrm{XF}}(L) - m - c - 1) \le |\cup_{j=m+1}^{v} X_j|$. Using Property 2.2, we get

$$2(b^{\mathrm{XF}}(L) - m - c - 1) + \frac{c - 1}{2} + 1 < \sum_{i \in L} w_i \le b^*(L),$$

or

$$2(b^{\mathrm{XF}}(L) - m - c - 1) + \frac{c - 1}{2} + 2 \le b^*(L).$$

Rearranging the left-hand side gives the second lower bound. ■

THEOREM 2.6: If c is odd, then

$$b^{\mathrm{XF}}(L) \le 1.75 b^*(L) - \tfrac{1}{4}.$$

PROOF: From Property 2.5 we have $2(b^{\mathrm{XF}}(L) - m) - 3c/2 - \frac{1}{2} \le b^*(L)$. Hence,

$$\begin{aligned} b^{\mathrm{XF}}(L) &\le \frac{b^*(L)}{2} + m + \frac{3c}{4} + \frac{1}{4} \\ &= \frac{b^*(L)}{2} + \left(m + \frac{c}{2} + \frac{1}{2}\right) + \frac{c}{4} - \frac{1}{4} \\ &\le 1.75 b^*(L) - \frac{1}{4}. \end{aligned}$$ ■

3. FIRST-FIT DECREASING AND BEST-FIT DECREASING

The proof of the worst-case bounds for FFD and BFD is based on Property 2.1 (which implies that if a bin produced by these heuristics contains no large item, then the first two items assigned to this bin cannot fit in any bin opened prior to it) and the fact that the list is ordered with all large items first.

Recall that XFD denotes either FFD or BFD. Index the bins produced by XFD in the order they are opened. Let $b^{\mathrm{XFD}}(L) = 3x + p$ for some integer x, $x \ge 1$, and $p = 0, 1, 2$. There are three cases depending on the value of p. First, suppose p is either 0 or 1. Consider bin $2x + 1$ (i.e., the bin whose index j satisfies $j = 2x + 1$). If this bin contains a large item we are done, because in that case $b^*(L) \ge 2x + 1 > \frac{2}{3} b^{\mathrm{XFD}}(L)$.

Otherwise, bins $2x + 1$ through $3x + p$ must collectively contain at least $2x + 2p - 1$ small items, none of which can fit in the first $2x$ bins. Hence, the sum of the item sizes exceeds $\min\{2x, 2x + 2p - 1\}$, which is not smaller than $2x + p - 1$, because p is either 0 or 1. This implies that $b^*(L) \geq 2x + p \geq \frac{2}{3}b^{XFD}(L)$.

Similarly, suppose $p = 2$; that is, $b^{XFD}(L) = 3x + 2$. If bin $2x + 2$ contains a large item we are done, because in that case $b^*(L) \geq 2x + 2 > \frac{2}{3}b^{XFD}(L)$. Otherwise, bins $2x + 2$ through $3x + 2$ collectively contain at least $2x + 1$ small items, none of which can fit in the first $2x + 1$ bins, implying the sum of the item sizes exceeds $2x + 1$ and hence $b^*(L) \geq 2x + 2 > \frac{2}{3}b^{XFD}(L)$.

ACKNOWLEDGMENT

The author would like to thank Dr. David S. Johnson for his comments on an earlier draft of the article, and to the referees for suggestions that improved and shortened the presentation. This research was supported in part by ONR Contract No. N00014-90-J-1649 and NSF Contract No. DDM-8922712.

REFERENCES

[1] Baker, B.S., "A New Proof for the First-Fit Decreasing Bin Packing Algorithm," *Journal of Algorithms*, **6**, 49–70 (1985).

[2] Coffman, E.G., Jr., Garey, M.R., and Johnson, D.S., "Approximation Algorithms for Bin-Packing—An Updated Survey," in G. Ausiello, M. Lucertini, P. Serafini (Eds.), *Algorithm Design for Computer System Design*, Springer-Verlag, Berlin, 1984, pp. 49–106.

[3] Garey, M.R., Graham, R.L., Johnson, D.S., and Yao, A.C., "Resource Constrained Scheduling as Generalized Bin Packing," *Journal of Combinatorial Theory*, *Series A*, **21**, 257–298 (1985).

[4] Garey, M.R., and Johnson, D.S., *Computers and Intractability*, W.H. Freeman and Company, New York, 1979.

[5] Johnson, D.S., Demers, A., Ullman, J.D., Garey, M.R., and Graham, R.L., "Worst-Case Performance Bounds for Simple One-Dimensional Packing Algorithms," *SIAM Journal on Computing*, **3**(4), 299–325 (1974).

Manuscript received April 20, 1992
Revised manuscript received October 22, 1993
Accepted November 23, 1993