SIAM J. COMPUT.
Vol. 3, No. 4, December 1974

WORST-CASE PERFORMANCE BOUNDS FOR SIMPLE ONE-DIMENSIONAL PACKING ALGORITHMS*

D. S. JOHNSON†, A. DEMERS‡, J. D. ULLMAN§,
M. R. GAREY‖ AND R. L. GRAHAM‖

Abstract. The following abstract problem models several practical problems in computer science and operations research: given a list L of real numbers between 0 and 1, place the elements of L into a minimum number L^* of "bins" so that no bin contains numbers whose sum exceeds 1. Motivated by the likelihood that an excessive amount of computation will be required by any algorithm which actually determines an optimal placement, we examine the performance of a number of simple algorithms which obtain "good" placements. The first-fit algorithm places each number, in succession, into the first bin in which it fits. The best-fit algorithm places each number, in succession, into the most nearly full bin in which it fits. We show that neither the first-fit nor the best-fit algorithm will ever use more than $\frac{17}{10}L^* + 2$ bins. Furthermore, we outline a proof that, if L is in decreasing order, then neither algorithm will use more than $\frac{11}{9}L^* + 4$ bins. Examples are given to show that both upper bounds are essentially the best possible. Similar results are obtained when the list L contains no numbers larger than $\alpha < 1$.

1. Introduction. Recent results in complexity theory [3], [10] indicate that many combinatorial optimization problems may be effectively impossible to solve, in the sense that a prohibitive amount of computation is required to construct optimal solutions for all but very small cases. In order to solve such problems in practice, one is forced to use approximate, heuristic algorithms which hopefully compute "good" solutions in an acceptable amount of computing time. Thus, instead of seeking the fastest algorithm from the set of exact optimization algorithms, one seeks the best approximation algorithm from the set of "sufficiently fast" algorithms. Unfortunately it is usually difficult to evaluate and compare the performance of heuristic algorithms, other than by running them on large problem sets with known optimal solutions. A more rigorous approach is to mathematically analyze the performance of such algorithms to determine how closely the constructed solutions approximate optimal solutions. In this paper, we consider a number of heuristic algorithms for an important one-dimensional packing problem and determine worst-case performance bounds, relative to the optimal solution for each.

We base our theoretical performance analyses on worst-case, rather than average, behavior. The analysis of expected performance for a realistic probability distribution on the problem domain (which is, in itself, usually difficult to

* Received by the editors December 3, 1973, and in revised form April 8, 1974.

† Department of Mathematics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139. The research reported here was supported in part by Project MAC, an MIT research program sponsored by the Advanced Research Projects Agency, Department of Defense, under Office of Naval Research Contract N00014-70-A-3062-0006, and by the National Science Foundation under Contract GJ00-4327. Now at Bell Laboratories, Murray Hill, New Jersey 07974.

‡ Department of Electrical Engineering, Princeton University, Princeton, New Jersey 08540. This work was supported by the National Science Foundation under Grant GJ-35570.

§ Department of Electrical Engineering, Princeton University, Princeton, New Jersey 08540. This work was supported by the National Science Foundation under Grant GJ-1052.

‖ Bell Laboratories, Murray Hill, New Jersey 07974.

determine) appears at present to be considerably more difficult. Worst-case results are easier—though decidedly nontrivial—to obtain and are quite useful, especially because they enable one to *guarantee* that a particular algorithm will never exceed the optimal solution by more than a known, hopefully small, percentage. Intuitively, one might expect that a "mechanism" which causes a particular algorithm to have a certain worst-case behavior might also be expected to manifest itself to a certain extent in the "average" case. Some experiments [4], [8] with randomly generated data have tended to confirm the hypothesis that, for the algorithms considered here, worst-case analysis does provide valid comparisons.

The basic problem to be considered can be stated quite simply: given a list $L = (a_1, a_2, \cdots, a_n)$ of real numbers in $(0, 1]$, place the elements of L into a minimum number L^* of "bins" so that no bin contains numbers whose sum exceeds 1.

This problem,[1] which is a special case of the one-dimensional "cutting-stock" problem [7] and the "assembly-line balancing" problem [2], models several practical problems in computer science. Some examples are:

1. *Table formatting.* Let the "bins" be computer words of fixed size: k bits. Suppose there are items of data (e.g., bit string of length 6, character string of 3 bytes, half-word integer) requiring $ka_1, ka_2, \cdots, ka_n$ bits, respectively. It is desirable to place the data in as few words as possible. The minimum is given by L^*, where L is the list $(a_1, a_2, \cdots, a_n)$.

2. *Prepaging.* Here, the bins are pages and the numbers in the list L represent fractions of a page required by program segments which should appear on a single page, e.g., inner loops, arrays, etc.

3. *File allocation.* It is desired to place files of varying sizes on as few tracks of a disc as possible, where files may not be broken between tracks.

Brown [1] gives a number of additional applications in industry and business. Since the abstract "bin packing" problem is "NP-complete" in the sense of Cook [3] and Karp [10], we can expect the problem of finding a packing which uses exactly L^* bins to require in general a lengthy combinatorial search for its solution. Thus we feel justified in considering the performance of various heuristic algorithms for constructing packings. In particular we shall consider the following four placement algorithms.

ALGORITHM 1 (First-fit). Let the bins be indexed as $B_1, B_2, \cdots$, with each initially filled to level zero. The numbers $a_1, a_2, \cdots, a_n$ will be placed in that order. To place a_i, find the least j such that B_j is filled to level $\beta \leqq 1 - a_i$, and place a_i in B_j. B_j is now filled to level $\beta + a_i$.

ALGORITHM 2 (Best-fit). Let the bins be indexed as $B_1, B_2, \cdots$, with each initially filled to level zero. The numbers $a_1, a_2, \cdots, a_n$ will be placed in that order. To place a_i, find the least j such that B_j is filled to level $\beta \leqq 1 - a_i$ and β is as large as possible, and place a_i in B_j. B_j is now filled to level $\beta + a_i$.

ALGORITHM 3 (First-fit decreasing). Arrange $L = (a_1, a_2, \cdots, a_n)$ into nonincreasing order and apply Algorithm 1 to the derived list.

ALGORITHM 4 (Best-fit decreasing). Arrange $L = (a_1, a_2, \cdots, a_n)$ into nonincreasing order and apply Algorithm 2 to the derived list.

[1] First brought to the attention of the last-named author by E. Arthurs (via S. A. Burr).

We use FF(L), BF(L), FFD(L) and BFD(L) to denote the number of bins used in applying each of the four algorithms, respectively, to the list L. The performance measure in which we are interested is the ratio of the number of bins used by a particular algorithm executed on L to the optimum number of bins L^*. Accordingly we use $R_{\mathrm{FF}}(k)$ to denote the maximum value achieved by the ratio $\mathrm{FF}(L)/L^*$ over all lists with $L^* = k$, with $R_{\mathrm{BF}}(k)$, $R_{\mathrm{FFD}}(k)$ and $R_{\mathrm{BFD}}(k)$ being defined similarly. Our main results, the first two of which appeared in a preliminary version of this paper [6], can be summarized as follows:

$$\lim_{k\to\infty} R_{\mathrm{FF}}(k) = \tfrac{17}{10}, \tag{1}$$

$$\lim_{k\to\infty} R_{\mathrm{BF}}(k) = \tfrac{17}{10}, \tag{2}$$

$$\lim_{k\to\infty} R_{\mathrm{FFD}}(k) = \tfrac{11}{9}, \tag{3}$$

$$\lim_{k\to\infty} R_{\mathrm{BFD}}(k) = \tfrac{11}{9}. \tag{4}$$

All these ratios are achieved for small values of k, so that these asymptotic results actually reflect the performance for essentially all values of k. In addition, similar results are obtained for certain restricted lists L.

2. First-fit and best-fit. We begin with a simple example which illustrates the type of list for which these two algorithms behave poorly. For any n divisible by 18 and $0 < \delta < \frac{1}{84}$, let the list $L = (a_1, a_2, \cdots, a_n)$ be defined by

$$a_i = \begin{cases} \frac{1}{6} - 2\delta, & 1 \leqq i \leqq n/3, \\ \frac{1}{3} + \delta, & n/3 < i \leqq 2n/3, \\ \frac{1}{2} + \delta, & 2n/3 < i \leqq n. \end{cases}$$

Clearly $L^* = n/3$ since the elements can be packed perfectly by placing one element from each of the three regions in each bin. However, as the reader can easily verify, both first fit and best fit will obtain the packing which consists of $n/18$ bins, each containing six elements of size $\frac{1}{6} - 2\delta$, $n/6$ bins, each containing two elements of size $\frac{1}{3} + \delta$, and $n/3$ bins each containing a single element of size $\frac{1}{2} + \delta$. The two packings are illustrated in Fig. 1. Thus we have

$$\frac{\mathrm{FF}(L)}{L^*} = \frac{\mathrm{BF}(L)}{L^*} = \frac{(n/18) + (n/6) + (n/3)}{n/3} = \frac{5}{3}.$$

By slightly modifying the list L given in the example, we can force even worse behavior and prove the following theorem.

THEOREM 2.1. *For every $k \geqq 1$, there exists a list L, with $L^* = k$, such that* $\mathrm{FF}(L) = \mathrm{BF}(L) > 1.7L^* - 8$.

Proof. As in the previous example, the elements of the list L will belong to three regions, with sizes nearly equal to $\frac{1}{6}$, $\frac{1}{3}$ and $\frac{1}{2}$, respectively. The number of elements belonging to each region will be the same, and those of the first region will precede those of the second region which precede those of the third region in the list L.

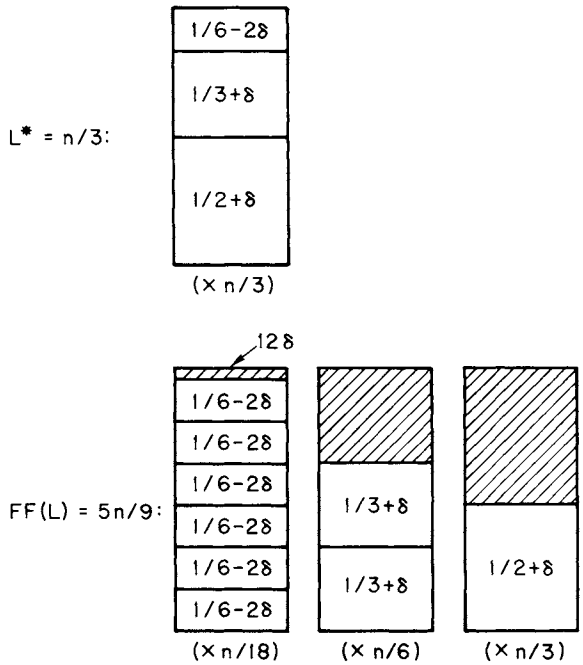

FIG. 1. *The 5/3 example*

Let N be a positive integer divisible by 17 and let δ be chosen so that $0 < \delta \ll 18^{-N/17}$. The first region will consist of $N/17$ *blocks* of ten numbers each. Let the numbers of the ith block of region 1 be denoted by $a_{0i}, a_{1i}, \cdots, a_{9i}$. These numbers are given by the following expressions, where $\delta_i = \delta \cdot 18^{(N/17)-i}$ for $1 \leqq i \leqq N/17$:

$$a_{0i} = \tfrac{1}{6} + 33\delta_i, \qquad a_{4i} = \tfrac{1}{6} - 13\delta_i,$$
$$a_{1i} = \tfrac{1}{6} - 3\delta_i, \qquad a_{5i} = \tfrac{1}{6} + 9\delta_i,$$
$$a_{2i} = a_{3i} = \tfrac{1}{6} - 7\delta_i, \qquad a_{6i} = a_{7i} = a_{8i} = a_{9i} = \tfrac{1}{6} - 2\delta_i.$$

Let the first $10N/17$ numbers in the list L be $a_{01}, a_{11}, \cdots, a_{91}, a_{02}, \cdots, a_{92}, \cdots, a_{0(N/17)}, \cdots, a_{9(N/17)}$. We notice that $a_{0i} + a_{1i} + \cdots + a_{4i} = \frac{5}{6} + 3\delta_i$, and $a_{5i} + a_{6i} + \cdots + a_{9i} = \frac{5}{6} + \delta_i$. Thus, for all i, the first five numbers of block i will fill up bin $2i - 1$, and the last five numbers of block i will fill up bin $2i$ when either the first-fit algorithm or the best-fit algorithm is applied to L. To see this we need only observe that a_{4i}, the smallest number in block i, will not fit in any of the previous bins, since the least filled of these, bin $2i - 2$, has contents totaling $\frac{5}{6} + \delta_{i-1} = \frac{5}{6} + 18\delta_i$. Also, the smallest of $a_{5i}, a_{6i}, \cdots, a_{9i}$, which is $\frac{1}{6} - 2\delta_i$, will not fit in bin $2i - 1$, which has contents totaling $\frac{5}{6} + 3\delta_i$. Thus the $N/17$ blocks in region 1 fill up $2N/17$ bins.

We now turn to region 2. Here the numbers are all about $\frac{1}{3}$, and they are again divided into $N/17$ blocks of ten numbers each. Let the ith block of region 2 be $b_{0i}, b_{1i}, \cdots, b_{9i}$. The numbers $b_{01}, b_{11}, \cdots, b_{91}, b_{02}, \cdots, b_{92}, \cdots, b_{0(N/17)}, \cdots, b_{9(N/17)}$ follow those of region 1 in the list L. The values of the numbers in block i

are given by:

$$b_{0i} = \tfrac{1}{3} + 46\delta_i, \qquad b_{4i} = \tfrac{1}{3} + 12\delta_i,$$
$$b_{1i} = \tfrac{1}{3} - 34\delta_i, \qquad b_{5i} = \tfrac{1}{3} - 10\delta_i,$$
$$b_{2i} = b_{3i} = \tfrac{1}{3} + 6\delta_i, \qquad b_{6i} = b_{7i} = b_{8i} = b_{9i} = \tfrac{1}{3} + \delta_i.$$

The numbers of block i fill bins $(2N/17) + 5i - 4$ through $(2N/17) + 5i$. These are filled with b_{0i} and b_{1i}, b_{2i} and b_{3i}, etc. To see this, we observe that the contents of the five bins filled by block i sum, respectively, to

$$\tfrac{2}{3} + 12\delta_i, \quad \tfrac{2}{3} + 12\delta_i, \quad \tfrac{2}{3} + 2\delta_i, \quad \tfrac{2}{3} + 2\delta_i, \quad \tfrac{2}{3} + 2\delta_i.$$

Thus $b_{5i} = \frac{1}{3} - 10\delta_i$ cannot fall into either of the first two bins, and $b_{1i} = \frac{1}{3} - 34\delta_i$ cannot fall into any of the bins for previous blocks since these are all filled to at least level $\frac{2}{3} + 2\delta_{i-1} = \frac{2}{3} + 36\delta_i$. Thus the $N/17$ blocks in region 2 fill up $5N/17$ bins.

The third region consists of $10N/17$ numbers, each equal to $\frac{1}{2} + \delta$. These complete the list L and fill one bin each. The total number of bins filled by either the first-fit algorithm or the best-fit algorithm applied to list L is thus $2N/17$ from region 1, $5N/17$ from region 2, and $10N/17$ from region 3, for a total of N bins.

However, the numbers on the list L can be packed into $(10N/17) + 1$ bins as follows. All but two of these bins contain one of the numbers $\frac{1}{2} + \delta$. The remaining space in each of these bins is filled with one of the following combinations:

(i) $a_{ji} + b_{ji}$ for some $2 \leqq j \leqq 9$ and $1 \leqq i \leqq N/17$,
(ii) $a_{0i} + b_{1i}$ for some $1 \leqq i \leqq N/17$,
(iii) $a_{1i} + b_{0(i+1)}$ for some $1 \leqq i \leqq N/17$.

This leaves b_{01}, $a_{1(N/17)}$, and one number $\frac{1}{2} + \delta$ which may be packed easily into the remaining two bins. We have thus shown that $L^* \leqq 1 + 10N/17$, so

$$\mathrm{FF}(L)/L^* \geqq 17N/(10N + 17) > 1.7 - 2/L^*$$

and, similarly

$$\mathrm{BF}(L)/L^* \geqq 17N/(10N + 17) > 1.7 - 2/L^*.$$

To obtain values of L^* not congruent to 1 (mod 10), we can form the list L' by adjoining to L m elements, each with size 1, where m is a fixed positive integer $\leqq 9$. The preceding arguments then show

$$\mathrm{FF}(L') = \mathrm{FF}(L) + m \quad \text{and} \quad L'^* = L^* + m$$

so that

$$\frac{\mathrm{FF}(L')}{L'^*} > \frac{17N + 17m}{10N + 17 + 17m} \geqq \frac{17}{10} - \frac{7m + 17}{10((10N/17) + m + 1)}$$
$$\geqq \frac{17}{10} - \frac{(7m + 17)/10}{L'^*} \geqq 1.7 - \frac{8}{L'^*},$$

since $m \leqq 9$. The same argument applies to $\mathrm{BF}(L)$. This proves Theorem 2.1.

We will now show that the examples constructed in the previous proof are essentially the worst possible, that is, 1.7 is the asymptotic least upper bound of the ratios $R_{\mathrm{FF}}(k)$ and $R_{\mathrm{BF}}(k)$.

THEOREM 2.2. *For every list* L, $\mathrm{FF}(L) \leqq 1.7L^* + 2$ *and* $\mathrm{BF}(L) \leqq 1.7L^* + 2$.

Proof. We use only the two following properties of the FF and BF algorithms.

(i) No element is placed in an empty bin unless it will not fit in *any* nonempty bin.

(ii) If there is a unique nonempty bin with lowest level, no element will be placed there unless it will *not* fit in any lower numbered bin.

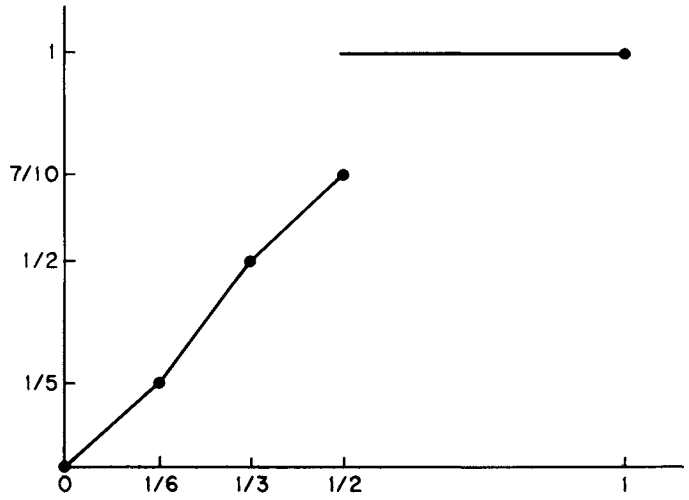

FIG. 2. *The function* $W(\alpha)$

Define W: $[0, 1] \to [0, 1]$ as follows (see Fig. 2):

$$W(\alpha) = \begin{cases} \frac{6}{5}\alpha & \text{for } 0 \leqq \alpha \leqq \frac{1}{6}, \\ \frac{9}{5}\alpha - \frac{1}{10} & \text{for } \frac{1}{6} < \alpha \leqq \frac{1}{3}, \\ \frac{6}{5}\alpha + \frac{1}{10} & \text{for } \frac{1}{3} < \alpha \leqq \frac{1}{2}, \\ 1 & \text{for } \frac{1}{2} < \alpha \leqq 1. \end{cases}$$

CLAIM 2.2.1. *Let some bin be filled with* $b_1, b_2, \cdots, b_m$. *Then* $\sum_{i=1}^{m} W(b_i) \leqq \frac{17}{10}$.

Proof. If $b \leqq \frac{1}{2}$, then $W(b)/b \leqq \frac{3}{2}$. The extreme ratio is reached only when $b = \frac{1}{3}$ and is less otherwise. Thus the lemma is immediate unless one b_i is greater than $\frac{1}{2}$. We may take this one to be b_1, and must show that if $\sum_{i=2}^{m} b_i < \frac{1}{2}$, then $\sum_{i=2}^{m} W(b_i) \leqq \frac{7}{10}$.

It should be noted that since the slope of $W(b)$ is the same in the regions $[0, \frac{1}{6}]$ and $[\frac{1}{3}, \frac{1}{2}]$, any b_i which is in the latter region can be replaced without loss of generality by two numbers of $\frac{1}{3}$ and $b_i - \frac{1}{3}$, respectively. We therefore assume that $b_i \leqq \frac{1}{3}$ for $2 \leqq i \leqq n$. Moreover, if b_j and b_k are both $\leqq \frac{1}{6}$, they can be combined into one and $\sum_i W(b_i)$ will not decrease; in fact it may increase. Thus we may assume that at most one of the b_i's, $i \geqq 2$, is in the range $(0, \frac{1}{6}]$, and the rest are in $(\frac{1}{6}, \frac{1}{3}]$.

We have consequently reduced the proof to the consideration of four cases:

Case 1. $m = 2, b_2 \leqq \frac{1}{3}$.

Case 2. $m = 3, \frac{1}{6} < b_2 \leqq b_3 \leqq \frac{1}{3}$.

Case 3. $m = 3, b_2 \leqq \frac{1}{6} < b_3 \leqq \frac{1}{3}$.

Case 4. $m = 4, b_2 \leqq \frac{1}{6} < b_3 \leqq b_4 \leqq \frac{1}{3}$.

Case 1 is immediate since $b_2 \leqq \frac{1}{2}$ implies $W(b_2) \leqq \frac{7}{10}$. In Case 2, $W(b_2) + W(b_3) = \frac{9}{5}(b_2 + b_3) - \frac{1}{5} \leqq \frac{9}{5} \cdot \frac{1}{2} - \frac{1}{5} = \frac{7}{10}$, since $b_2 + b_3 \leqq \frac{1}{2}$. For Case 3, $W(b_2) + W(b_3) = \frac{6}{5}b_2 + \frac{9}{5}b_3 - \frac{1}{10} \leqq \frac{1}{5} + \frac{3}{5} - \frac{1}{10} = \frac{7}{10}$. And finally, in Case 4, $W(b_2) + W(b_3) + W(b_4) \leqq \frac{6}{5}b_2 + \frac{9}{5}(b_3 + b_4) - \frac{1}{5} = \frac{9}{5}(b_2 + b_3 + b_4) - \frac{3}{5}b_2 - \frac{1}{5} \leqq \frac{9}{10} - \frac{1}{5} = \frac{7}{10}$, since $b_2 + b_3 + b_4 \leqq \frac{1}{2}$.

Let us define the *coarseness* of a bin to be the largest α such that some bin with smaller index is filled to level $1 - \alpha$. The coarseness of the first bin is 0.

CLAIM 2.2.2. *Suppose bins are filled according to either the* FF *or the* BF *algorithm, and some bin B has coarseness* α. *Then every member of B that was placed there before B was more than half full exceeds* α.

Proof. Until the bin has been filled to a level greater than $\frac{1}{2}$, it must be either empty or the unique nonempty bin of lowest level (by property (i) of the placement algorithm), so by constraints (i) and (ii), any element placed in the bin must not fit in any bin with lower index, and hence must exceed α.

CLAIM 2.2.3. *Let a bin of coarseness* $\alpha < \frac{1}{2}$ *be filled with numbers* $b_1 \geqq b_2 \geqq \cdots \geqq b_m$ *in the completed* FF*-packing* (BF*-packing*). *If* $\sum_{i=1}^{m} b_i > 1 - \alpha$, *then* $\sum_{i=1}^{m} W(b_i) \geqq 1$.

Proof. If $b_1 > \frac{1}{2}$, then the result is immediate since $W(b_1) = 1$. We therefore assume that $b_1 \leqq \frac{1}{2}$. If $m \geqq 2$, then the second element placed in the bin was placed before the bin was more than half full, so by Claim 2.2.2, at least two of the elements exceed α. In particular, we must have $b_1 \geqq b_2 \geqq \alpha$. We consider several cases depending on the range of α.

Case 1. $\alpha \leqq \frac{1}{6}$. Then $\sum_{i=1}^{m} b_i > 1 - \alpha \geqq \frac{5}{6}$. Since $W(\beta)/\beta \geqq \frac{6}{5}$ in the range $0 \leqq \beta \leqq \frac{1}{2}$, we immediately have $\sum_{i=1}^{m} W(b_i) \geqq \frac{6}{5} \cdot \frac{5}{6} = 1$.

Case 2. $\frac{1}{6} \leqq \alpha \leqq \frac{1}{3}$. We consider subcases (a)–(c), depending on the value of m.

(a) $m = 1$. Since $b_1 \leqq \frac{1}{2}$, we must have $1 - \alpha \leqq \frac{1}{2}$ or $\alpha \geqq \frac{1}{2}$, which contradicts our assumption that $\alpha \leqq \frac{1}{3}$.

(b) $m = 2$. If both b_1 and b_2 are $\geqq \frac{1}{3}$, then $W(b_1) + W(b_2) \geqq (\frac{6}{5} \cdot \frac{1}{3} + \frac{1}{10}) \cdot 2 = 1$. If both are $< \frac{1}{3}$, then $b_1 + b_2 < \frac{2}{3} < 1 - \alpha$, which contradicts our hypothesis. If $b_1 \geqq \frac{1}{3}$ and $b_2 < \frac{1}{3}$, then, since both must be greater than α, $\alpha < b_1 < \frac{1}{3} \leqq b_2 \leqq \frac{1}{2}$. Hence $W(b_1) + W(b_2) = \frac{9}{5}b_1 - \frac{1}{10} + \frac{6}{5}b_2 + \frac{1}{10} = \frac{6}{5}(b_1 + b_2) + \frac{3}{5}b_1$. Since $b_1 + b_2 \geqq 1 - \alpha$ and $b_1 > \alpha$, we thus have $W(b_1) + W(b_2) \geqq \frac{6}{5}(1 - \alpha) + \frac{3}{5}\alpha = 1 + \frac{1}{5} - \frac{3}{5}\alpha \geqq 1$, since $\alpha \leqq \frac{1}{3}$.

(c) $m \geqq 3$. As in the previous case, if two of the b_i are $\geqq \frac{1}{3}$, the result is immediate. If $b_1 \geqq \frac{1}{3} > b_2 \geqq \alpha$, then

$$W(b_1) + W(b_2) + \sum_{i=3}^{m} W(b_i)$$

$$\geqq \tfrac{6}{5}b_1 + \tfrac{1}{10} + \tfrac{9}{5}b_2 - \tfrac{1}{10} + \frac{6}{5}\sum_{i=3}^{m} b_i$$

$$= \frac{6}{5}\sum_{i=1}^{m} b_i + \tfrac{3}{5}b_2 \geqq \tfrac{6}{5}(1 - \alpha) - \tfrac{3}{5}\alpha = 1 + \tfrac{1}{5} - \tfrac{3}{5}\alpha \geqq 1.$$

If $\frac{1}{3} > b_1 \geqq b_2 > \alpha$, then

$$W(b_1) + W(b_2) + \sum_{i=3}^{m} W(b_i) \geqq \tfrac{9}{5}(b_1 + b_2) - \tfrac{1}{5} + \frac{6}{5}\sum_{i=3}^{m} b_i$$

$$\geqq \tfrac{6}{5}(1 - \alpha) + \tfrac{3}{5}(2\alpha) - \tfrac{1}{5} = 1 + \tfrac{6}{5}\alpha - \tfrac{6}{5}\alpha = 1.$$

Case 3. $\frac{1}{3} < \alpha < \frac{1}{2}$. If $m = 1$, we have $b_1 \geqq 1 - \alpha > \frac{1}{2}$, so $W(b_1) = 1$.

If $m \geqq 2$, then $b_1 \geqq b_2 > \frac{1}{3}$ and the result is immediate.

CLAIM 2.2.4. *If a bin of coarseness* $\alpha < \frac{1}{2}$ *is filled with* $b_1 \geqq \cdots \geqq b_m$ *and* $\sum_{i=1}^{m} W(b_i) = 1 - \beta$, *where* $\beta > 0$, *then either*

(i) $m = 1$ *and* $b_1 < \frac{1}{2}$, *or*

(ii) $\sum_{i=1}^{m} b_i \leqq 1 - \alpha - \frac{5}{9}\beta$.

Proof. If $m = 1$ and $b_1 > \frac{1}{2}$, it is impossible that $\beta > 0$. Therefore, if (i) does not hold, we may assume that $m \geqq 2$, and hence $b_1 \geqq b_2 \geqq \alpha$ by reasoning of the previous claim. Let $\sum_{i=1}^{m} b_i = 1 - \alpha - \gamma$. Then we may construct a bin filled with $b_3, b_4, \cdots, b_m$ and two other numbers δ_1 and δ_2, selected so that $\delta_1 + \delta_2 = b_1 + b_2 + \gamma$, $\delta_1 \geqq b_1$, $\delta_2 \geqq b_2$, and so that neither δ_1 nor δ_2 exceeds $\frac{1}{2}$. By the proof of Claim 2.2.3 and the fact that both δ_1 and δ_2 exceed α, $\sum_{i=3}^{m} W(b_i) + W(\delta_1) + W(\delta_2) \geqq 1$. But since the slope of W in the range $[0, \frac{1}{2}]$ does not exceed $\frac{9}{5}$, it follows that $W(\delta_1) + W(\delta_2) \leqq W(b_1) + W(b_2) + \frac{9}{5}\gamma$. Therefore $\gamma \geqq \frac{5}{9}\beta$, and (ii) holds.

We are now prepared to complete the proof. Let $L = (a_1, a_2, \cdots, a_n)$ and $\overline{W} = \sum_{i=1}^{n} W(a_i)$. By Claim 2.2.1, $L^* \cdot \frac{17}{10} \geqq \overline{W}$.

Suppose that in the FF (BF) algorithm bins $B'_1, B'_2, \cdots, B'_m$ are all the bins that receive at least one element and for which $\sum_j W(a_j) = 1 - \beta_i$ with $\beta_i > 0$, where j ranges over all elements in bin B'_i. We assume that $1 \leqq i < j \leqq m$ implies that B'_i had a smaller index than B'_j in the original indexing of all bins. Let α_i be the coarseness of B'_i. Since B'_i contains no element exceeding $\frac{1}{2}$, we must have each $\alpha_i < \frac{1}{2}$. By Claim 2.2.4 and the definition of coarseness,

$$\alpha_i \geqq \alpha_{i-1} + \tfrac{5}{9}\beta_{i-1} \quad \text{for } 1 < i \leqq m.$$

Thus

$$\sum_{i=1}^{m-1} \beta_i \leqq \frac{9}{5}\sum_{i=2}^{m} (\alpha_i - \alpha_{i-1}) = \tfrac{9}{5}(\alpha_m - \alpha_1) \leqq \tfrac{9}{5} \cdot \tfrac{1}{2} < 1.$$

Since β_m cannot exceed 1, we have $\sum_{i=1}^{m} \beta_i \leqq 2$. Applying Claim 2.2.3, we obtain

$$\mathrm{FF}(L) \leqq \overline{W} + 2 \leqq (1.7)L^* + 2 \quad \text{and} \quad \mathrm{BF}(L) \leqq \overline{W} + 2 \leqq (1.7)L^* + 2,$$

completing the proof.

As a consequence of Theorems 2.1 and 2.2, we have a corollary.

COROLLARY. (i) $\lim_{k\to\infty} R_{\mathrm{FF}}(k) = 1.7$,

(ii) $\lim_{k\to\infty} R_{\mathrm{BF}}(k) = 1.7$.

It is interesting to note that for several values of k, the ratio $\frac{17}{10}$ can actually be attained. In particular, there is a list L with $L^* = 10$ and $\mathrm{FF}(L) = \mathrm{BF}(L) = 17$. The two packings, with all quantities in units of $\frac{1}{101}$, are shown in Fig. 3. The list L is in nondecreasing order. There is also a list L having $L^* = 20$ and $\mathrm{FF}(L) = \mathrm{BF}(L) = 34$. It may be true, however, that $R_{\mathrm{FF}}(k) < 1.7$ and $R_{\mathrm{BF}}(k) < 1.7$ for $k > 20$.

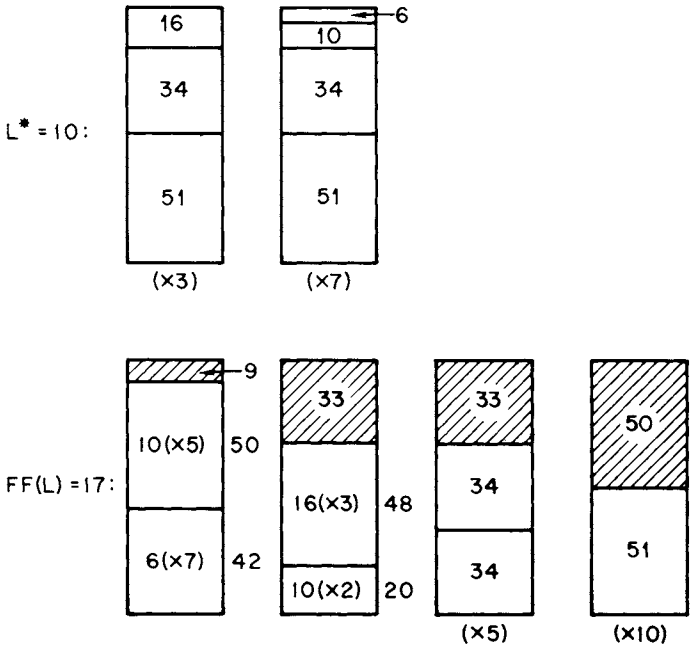

FIG. 3. *An example with* FF(L) = 17 *and* L^* = 10

If the list L is such that $a_i \leqq \alpha \leqq \frac{1}{2}$ for all i, the worst-case behavior of the two placement algorithms is not as extreme. We then have the following result.

THEOREM 2.3. *For any positive* $\alpha \leqq \frac{1}{2}$, *let* $m = \lfloor \alpha^{-1} \rfloor$. *Then we have*

(*i*) *for each* $k \geqq 1$, *there exists a list*[2] $L \subseteq (0, \alpha]$ *with* $L^* = k$ *such that* $\mathrm{FF}(L) \geqq [(m+1)/m]L^* - (1/m)$; *and*

(*ii*) *for any list* $L \subseteq (0, \alpha]$, $\mathrm{FF}(L) \leqq [(m+1)/m]L^* + 2$.

Both (*i*) *and* (*ii*) *hold with* FF *replaced by* BF.

Proof. We first describe how one constructs lists L, with no element exceeding α, for which

$$\frac{\mathrm{FF}(L)}{L^*} = \frac{\mathrm{BF}(L)}{L^*} = \frac{m+1}{m} - \frac{1}{mL^*}.$$

Let k be any positive integer. The list L is composed of elements which are all very close to $1/(m+1)$. The elements are of two types, described as follows:

$$b_j = 1/(m+1) - m^{2j+1}\delta, \qquad j = 1, 2, \cdots, k-1;$$

$$a_{1j} = a_{2j} = \cdots = a_{mj} = 1/(m+1) + m^{2j}\delta, \quad j = 1, 2, \cdots, k,$$

where $\delta > 0$ is chosen suitably small. The list L has the a-type elements occurring in nonincreasing order and the b-type elements occurring in strictly increasing order interspersed so that each successive pair, b_j and b_{j-1}, of b-type elements has

[2] Strictly speaking, L is not a set, but a sequence. However, the use of set terminology is convenient and should cause no confusion. Other instances will follow.

precisely m a-type elements occurring between them. The list L is then completely specified by the property that b_{k-1} occurs as the second element. We leave it for the reader to verify that

$$\mathrm{FF}(L) = \mathrm{BF}(L) = \left\lceil \frac{k(m+1)-1}{m} \right\rceil.$$

It is easy to see that the elements of L can be packed optimally by placing $b_j, a_{1j}, a_{2j}, \cdots, a_{mj}$ in a single bin for each $j = 1, \cdots, k-1$ and placing $a_{1k}, a_{2k}, \cdots, a_{mk}$ in one additional bin. This gives $L^* = k$. We then have

$$\frac{\mathrm{FF}(L)}{L^*} = \frac{\mathrm{BF}(L)}{L^*} \geqq \frac{k(m+1)-1}{mk} = \frac{m+1}{m} - \frac{1}{mL^*}.$$

The upper bound is also easily proved. Suppose that the list L contains no element exceeding $1/m$, m an integer.

Consider an FF packing of L. Every bin, except possibly the last bin, contains at least m elements. Disregarding the last bin, suppose two bins B_i and B_j, $i < j$, each contain elements totaling less than $m/(m+1)$. Then, since B_j contains m elements, B_j must contain an element with size less than $1/(m+1)$. But this element would have fit in B_i and thus could not have been placed in B_j by FF, a contradiction. Thus all but at most two bins must contain elements totaling at least $m/(m+1)$. Thus, letting $w(L)$ denote the sum of all elements on L, we have

$$L^* \geqq w(L) \geqq \frac{m}{m+1}(\mathrm{FF}(L) - 2),$$

so that

$$\mathrm{FF}(L) \leqq (m+1)L^*/m + 2.$$

A similar, but slightly more complicated, argument can be used to prove this for BF.

If we let $R^{\alpha}_{\mathrm{FF}}(k)$ and $R^{\alpha}_{\mathrm{BF}}(k)$ be defined analogously to $R_{\mathrm{FF}}(k)$ and $R_{\mathrm{BF}}(k)$ for lists $L \subseteq (0, \alpha]$, we have the following immediate corollary:

COROLLARY. $\lim_{k\to\infty} R^{\alpha}_{\mathrm{FF}}(k) = \lim_{k\to\infty} R^{\alpha}_{\mathrm{BF}}(k) = 1 + \lfloor \alpha^{-1} \rfloor^{-1}$.

3. First-fit decreasing and best-fit decreasing. The main results about FFD and BFD are the following.

THEOREM 3.1. *For each $k \geqq 1$, there exists a list L with $L^* = k$ such that* $\mathrm{FFD}(L) = \mathrm{BFD}(L) > \frac{11}{9}L^* - 2$.

THEOREM 3.2. *For all lists L,* $\mathrm{FFD}(L) \leqq \frac{11}{9}L^* + 4$, *and* $\mathrm{BFD}(L) \leqq \frac{11}{9}L^* + 4$. From these it follows that $\lim_{k\to\infty} R_{\mathrm{FFD}}(k) = \lim_{k\to\infty} R_{\mathrm{BFD}}(k) = \frac{11}{9}$.

The proof of Theorem 3.1 consists of a simple construction. Let ε satisfy $0 < \varepsilon < \frac{1}{12}$, $N = \lfloor k/9 \rfloor$, $\bar{k} \equiv k \pmod 9$ with $0 \leqq \bar{k} < 9$, $n = 30N + \bar{k}$, and consider the list $L = (a_1, \cdots, a_n)$ formed as follows:

$$a_i = \begin{cases} .5 + \varepsilon & \text{for } 1 \leqq i \leqq 6N, \\ .25 + 2\varepsilon & \text{for } 6N < i \leqq 12N, \\ .25 + \varepsilon & \text{for } 12N < i \leqq 18N, \\ .25 - 2\varepsilon & \text{for } 18N < i \leqq 30N, \\ 1.0 & \text{for } 30N < i \leqq n. \end{cases}$$

When L is put in decreasing order, the elements of size 1 will head the list, and BFD and FFD will yield the same packing. Figure 4 shows both this and the optimal packings. We have $L^* = 9N + \bar{k} = k$, and $\mathrm{FFD}(L) = \mathrm{BFD}(L) = 11N + \bar{k} > \frac{11}{9}L^* - 2$.

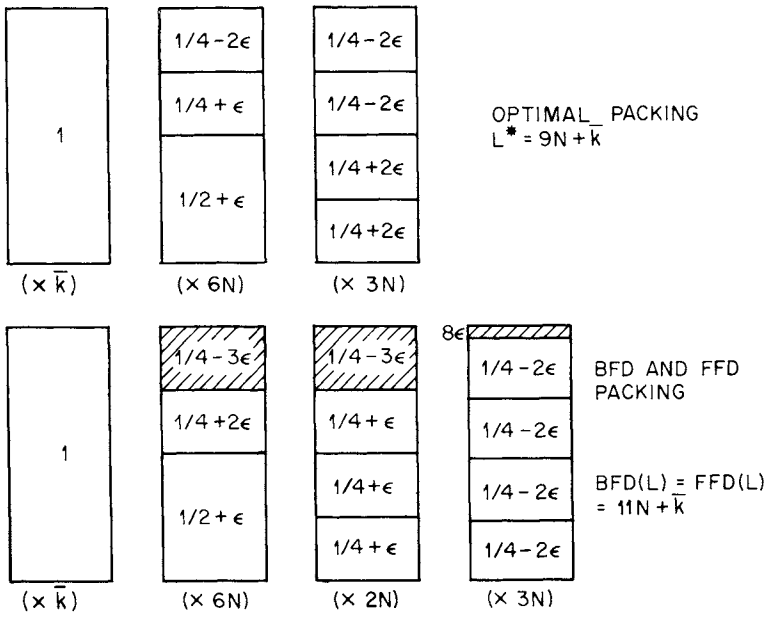

FIG. 4. *An 11/9 example*

The proof of Theorem 3.2 is considerably more complicated than the upper bound proofs in § 2, although some of the same ideas are involved. In this section we will show that we need only prove the result for the algorithm FFD and a restricted class of lists. In § 4 we will indicate how we go about proving the simplified, though still very difficult, result.

LEMMA 3.3. *Suppose L is a list such that* $\mathrm{FFD}(L) > rL^* + d$, *with* $r, d \geqq 1$. *Then the list L' obtained from L by deleting all elements not exceeding $(r-1)/r$ also has* $\mathrm{FFD}(L') > rL'^* + d$. *The same holds if* FFD *is replaced by* BFD.

Proof. Let P be the packing of L and P' the packing of L', using K and K' bins, respectively. If $K > K'$, then no element in the last bin of P can be larger than $(r-1)/r$, and hence all but the last must have levels exceeding $1/r$, since neither BF nor FF will start a new bin with an element which would fit in a previous bin. Thus $L^* \geqq \sum a_i > (1/r)(K-1)$ and so $K < rL^* + 1$, contrary to hypothesis. Hence $K' \geqq K$ and the lemma is proved.

Thus to prove Theorem 3.2 we need only consider lists $L \subset (\frac{2}{11}, 1]$. We would also like to use a single proof that would simultaneously yield the desired result for both FFD and BFD. For instance, in § 2 we simultaneously proved Theorem 2.2 for both FF and BF by only using properties the two algorithms have in common. Although *this* approach has not been successful for the current theorem, we could still use just one proof for both BFD and FFD if it could be shown that the result for one were just a simple corollary of the result for the other.

For instance, if for all lists L, $\mathrm{FFD}(L) \leqq \mathrm{BFD}(L)$, the result for FFD would follow immediately from that for BFD, or vice versa. Unfortunately, as the examples in Figs. 5 and 6 show, there are both lists L with $\mathrm{FFD}(L) < \mathrm{BFD}(L)$ and ones with $\mathrm{BFD}(L) < \mathrm{FFD}(L)$. Figure 5 presents packings of lists L with $\mathrm{BFD}(L) = \frac{10}{9}\mathrm{FFD}(L)$, and Fig. 6 presents L with $\mathrm{FFD}(L) = \frac{11}{10}\mathrm{BFD}(L)$. However, observe that the first example contains numbers less than $\frac{1}{6}$, whereas any list $L \subset (\frac{2}{11}, 1]$ contains no such numbers. Thus if these numbers less than $\frac{1}{6}$ are *essential* for examples like those in Fig. 5, we will still have $\mathrm{BFD}(L) \leqq \mathrm{FFD}(L)$ for all lists L we need to consider for Theorem 3.2, and so the result for BFD would follow from that for FFD. This is indeed the case, and we devote the remainder of this section to the details of the proof.

THEOREM 3.4. *Suppose* $L \subseteq [\frac{1}{6}, 1]$. *Then* $\mathrm{BFD}(L) \leqq \mathrm{FFD}(L)$.

Proof. Let $L = (a_1, \cdots, a_n)$ be ordered so that $a_i \geqq a_{i+1}$, $1 \leqq i < n$, with $a_n \geqq \frac{1}{6}$. We assume we have a copy of the FFD packing of L, denoted by PF, and are now proceeding to construct the BFD packing, element by element. At each step we will show that there is a way to extend the current packing to a packing of all of L using no more than $\mathrm{FFD}(L)$ bins.

For each i, $0 \leqq i \leqq n$, let L_i be the final segment of L consisting of all elements with index exceeding i. Thus $L_0 = L$. Let P_0 be the empty packing of $L - L_0 = \varnothing$, with which we begin the generation of the BFD packing, and let $f_0 : L_0 \to N \times N$ be defined as follows: if $a_i \in L$ is the kth largest element in bin j in PF (with ties broken according to the ordering of L), then $f_0(a_i) = (j, k)$.

The ordered pair (j, k) may be thought of as representing the kth *position* in bin j. We let k_j denote the number of elements in bin j in PF. Then we can define

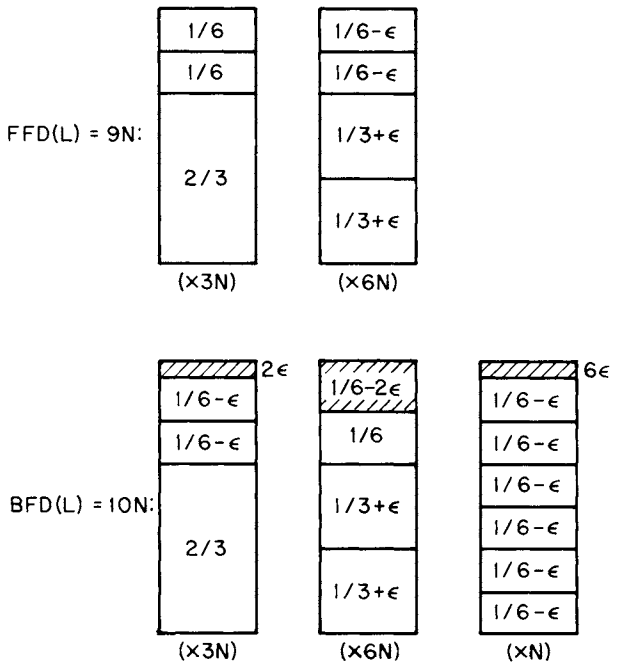

FIG. 5. *An example with* L^* *large and* $\mathrm{BFD}(L)/\mathrm{FFD}(L) = 10/9$

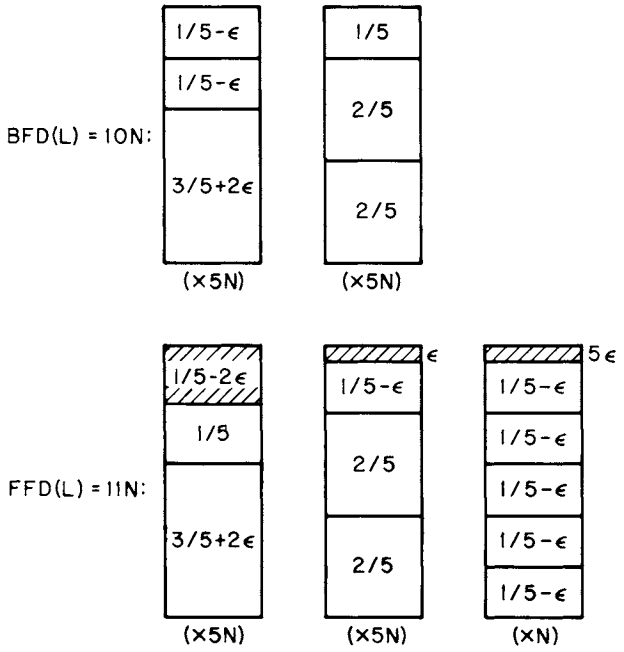

FIG. 6. *An example with L* large and* FFD(*L*)/BFD(*L*) = 11/10

the *empty positions of* P_0 to be

$$S_0 = \{(j,k): 1 \leqq j \leqq \mathrm{FFD}(L), 1 \leqq k \leqq k_j\},$$

that is, the positions which are *filled* in PF but not in P_0. This makes f_0 a 1–1 map from L_0, the elements of L remaining to be packed, to S_0, the positions remaining to be filled in P_0. Thus, in essence, f_0 shows us how to extend P_0 to a packing of all of L, by placing each remaining a_i in position $f_0(a_i)$, with none of the elements going in bins which were not used in PF.

The properties of f_0, P_0, and S_0 which allow this to happen can be summarized as follows, with $i = 0$:

(A) $f_i: L_i \to S_i$ is 1–1.

(B) $S_i \subseteq \{(j,h): (j,h)$ is filled in PF but empty in $P_i\}$.

(C) For each $j \geqq 1$, the sum of the elements in bin j in P_i, plus the sum of the elements which *map* to bin j under f_i, does not exceed 1.

As the generation of the BF packing proceeds, we construct P_i, S_i and f_i: $L_i \to S_i$ for $1 \leqq i \leqq n$ as follows: the packing P_i is obtained by adding element a_i to packing P_{i-1}. The bin it goes in is chosen according to the BFD placement rule. If that bin is the jth, the *position* a_i fills is that (j,k) which was unfilled in P_{i-1} and has minimal k. (It is possible that we may have to take $k > k_j$, if the bin already contains k_j elements.) We then set $S_i = S_{i-1} - \{(j,k)\}$. f_i is identical with f_{i-1} restricted to L_i (that is, with a_i deleted from its domain), with one possible exception. If $a_{i'}$, with $i' > i$, has $f_{i-1}(a_{i'}) = (j,k)$, we set $f_i(a_{i'}) = f_{i-1}(a_i)$. This insures that $f_i(a_{i'}) \in S_i$ and that f_i remains 1–1. In fact, a very elementary induction will establish

CLAIM 3.4.1. (A) *and* (B) *hold for all* i, $0 \leqq i \leqq n$.

Thus each f_i will provide us with a way to extend P_i to a packing of the entire list L with each element of L_i going in a bin which was used in PF, if only we can show that (C) also holds. What is more, this will prove the theorem, for we have the following claim.

CLAIM 3.4.2. *If for all* $i, 0 \leqq i \leqq n$, (A), (B) *and* (C) *hold, then* $\mathrm{BFD}(L) \leqq \mathrm{FFD}(L)$.

Proof. Suppose $\mathrm{BFD}(L) > \mathrm{FFD}(L)$. Let a_{m+1} be the first element assigned to bin $\mathrm{FFD}(L) + 1$ under BFD. By (A) and (B), $f_m(a_{m+1})$ is a position in one of the first $\mathrm{FFD}(L)$ bins of P_m, and by (C), a_{m+1} will fit in that bin. Thus a_{m+1} could not have gone in an empty bin (bin $\mathrm{FFD}(L) + 1$) to the right of that bin without violating the BFD placement rule. Thus bin $\mathrm{FFD}(L) + 1$ can never become nonempty, and so $\mathrm{BFD}(L) \leqq \mathrm{FFD}(L)$.

The proof of Theorem 3.4 is thus reduced to showing that (C) holds for $1 \leqq i \leqq n$. Rather than prove this directly by induction, we shall use two slightly more technical induction hypotheses which, together with (A) and (B), imply (C).

(D) If $(j, k) \in S_i \cap \mathrm{Range}(f_i)$, then $\mathrm{index}[f_i^{-1}(j, k)] \geqq \mathrm{index}[f_0^{-1}(j, k)]$.

(E) If a_r fills position (j, k), $1 \leqq k < k_j$, in P_i, then $r \geqq \mathrm{index}[f_0^{-1}(j, k)]$.

CLAIM 3.4.3. *If* (A), (B), (D) *and* (E) *hold for* i, *then* (C) *also holds for* i.

Proof. Let us consider an arbitrary bin j. We wish to show that the sum of the elements in bin j in P_i, plus the sum of the elements mapping to bin j under f_i, does not exceed 1. If no element of L_i maps to bin j, the result is immediate. If any element *does*, then by (A) and (B), position (j, k_j) must be empty in P_i, so that (E) will apply to all elements in the bin. If we were to place all elements that map to bin j in the positions to which they map, there would still be at most one element per position, because by (A) f_i is 1–1, and by (A) and (B) no element is mapped to a position which is already filled in P_i. By (D) and (E), each element would have index no smaller, and hence size no larger, than the element which filled its position in PF. Since the sum of the elements in any bin in PF does not exceed 1, the claim is proved.

CLAIM 3.4.4. (D) *holds for all* $i, 0 \leqq i \leqq n$.

Proof. (D) holds trivially for $i = 0$. If (D) holds for $i - 1$, the only opportunity for it to fail for i would be an $a_{i'}$, $i' > i$, for which $f_i(a_{i'}) \neq f_{i-1}(a_{i'})$. This can only happen if $f_i(a_i') = f_{i-1}(a_i)$. However, by (D) for $i - 1$, $\mathrm{index}[f_0^{-1}(f_{i-1}(a_i))] \leqq i < i'$, so (D) continues to hold for i. The claim follows by induction.

The proof of Theorem 3.4 is thus reduced to showing that (E) holds for all i, $0 \leqq i \leqq n$. We do this in two steps. First let $h = \max\{i : a_i > \frac{1}{3}\}$, where h is taken to be 0 if the set is empty.

CLAIM 3.4.5. *Each* a_i, $1 \leqq i \leqq h$, *is placed by* BFD *in position* $f_0(a_i)$.

Proof. Since the first elements placed in each of the bins under BFD form a nonincreasing sequence from left to right, the first time that the BFD choice could differ from the FFD choice can only have occurred when some bin B to the right of the FFD choice already contained two or more elements. But if this happens before a_h is assigned, the right-hand bin B would have had a level exceeding $\frac{2}{3}$ and would not have room for any additional elements exceeding $\frac{1}{3}$. Thus, until a_h is assigned, each a_i goes under BFD into the position it filled in PF, that is, position $f_0(a_i)$.

COROLLARY. (E) *holds for all* $i, 0 \leqq i \leqq h$.

Now let us order the positions by letting $(j, k) \leqq (j', k')$ mean that either $j < j'$ or $j = j'$ and $k \leqq k'$. The following fact about f_0 will be useful in showing that (E) holds for $i > h$.

CLAIM 3.4.6. *If* $(j, k), (j', k') \in S_h, k < k_j$, *and* $(j, k) \leqq (j', k')$, *then* $\text{index}[f_0^{-1}(j, k)] \leqq \text{index}[f_0^{-1}(j', k')]$.

Proof. Let a_i be the element in position (j, k) in PF, $a_{i'}$ the element in (j', k'). Since the positions are empty in P_h, we must have $\frac{1}{6} \leqq a_i, a_{i'} \leqq \frac{1}{3}$. If $j = j'$, the result is immediate, as position (j, k'), $k' \geqq k$, cannot have been filled before (j, k) under FF and so we must have $i' \geqq i$ as desired. So assume $j < j'$. Since $k < k_j$, position (j, k_j) must have been unfilled when a_i was to be assigned under FF, and so until a_i was assigned, the gap in bin j, was at least $2 \cdot \frac{1}{6} = \frac{1}{3}$. If $i' < i$, then $a_{i'}$ was assigned before a_i was, and so would have fit in bin j, contradicting our assumption that the FF rule assigned $a_{i'}$ to bin j', which is to the right of bin j. Thus we must have $i' \geqq i$ in this case also, and the claim is proved.

We are now ready to conclude the proof of Theorem 3.4 with the following claim.

CLAIM 3.4.7. (E) *holds for all* i, $h \leqq i \leqq n$.

Proof. By Claim 3.4.5, we know that (E) holds for $i = h$. Suppose it holds for $i - 1$. We shall show it holds for i, and the claim will follow by induction. Consider element a_i. Let (j, k) be the position it fills in P_i, and let $(j', k') = f_{i-1}(a_i)$. If $k \geqq k_j$, then (E) does not apply to the position filled by a_i, and so automatically continues to hold. So we may assume $k < k_j$. If $(j, k) \leqq (j', k')$, then by Claim 3.4.6 and (D),

$$\text{index}[f_0^{-1}(j, k)] \leqq \text{index}[f_0^{-1}(j', k')] \leqq \text{index}[f_{i-1}^{-1}(j', k')] = i,$$

and (E) would not be violated.

The only other possibility is $(j, k) > (j', k')$ and $k < k_j$. We shall show that in fact this cannot happen. Since both positions (j, k) and (j', k') must be empty in P_{i-1} and a_i goes in the bottom-most unfilled position in bin j, $(j, k) > (j', k')$ implies $j' < j$, and so bin j is to the right of bin j'. By (D) and (E) for $i - 1$ and Claim 3.4.3, a_i would have fit in bin j' of P_{i-1}. Since it went to the right of bin j' under the BFD rule, the level of bin j must have exceeded that of bin j' in P_{i-1}. However, since $k < k_j$, (E) applies to the elements in bin j in P_{i-1}, and so they take up no more space than the corresponding elements in PF. Consequently the gap in bin j, which we shall write gap(j), is at least as large as $f_0^{-1}(j, k) + f_0^{-1}(j, k_j) \geqq \frac{1}{6} + \frac{1}{6} = \frac{1}{3}$. Thus gap($j'$), the gap in bin j' in P_{i-1}, must *exceed* $\frac{1}{3}$. Now we cannot have $k = 1$, as then gap(j) $= 1 \geqq$ gap(j'), and we must have gap(j') $>$ gap(j). Thus $k > 1$, and consequently bin j must contain a bottom element b_1 with $b_1 >$ gap(j') $> \frac{1}{3}$ (see Fig. 7). Moreover, since list L is in decreasing order, the bottom elements in the bins form a nonincreasing sequence from left to right, and so if $k = 2$ we would again have gap(j) $\geqq$ gap(j'). Hence $k > 2$ and there is a second element in bin j in P_{i-1} (call it b_2) with $b_2 >$ gap(j') $> \frac{1}{3}$. But by (E) for $i - 1$, this means that the sum of the bottom two elements in bin j in PF also exceeds $\frac{2}{3}$, and hence the bin could have contained at most one additional element greater than or equal to $\frac{1}{6}$, so that $k_j \leqq 3$. Since $k > 2$ we thus have $k \geqq k_j$, the desired contradiction. So this case is impossible, (E) cannot be violated, and the claim is proved.

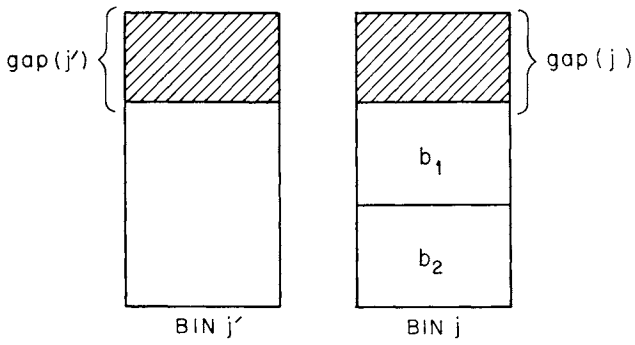

FIG. 7

Thus by Claims 3.4.5 and 3.4.7, (E) holds for all i, $0 \leqq i \leqq n$. Claim 3.4.1 and 3.4.4 tell us that (A), (B) and (D) also hold for such i. Thus (C) holds by Claim 3.4.3, and the theorem follows by Claim 3.4.2. Q.E.D.

A similar argument [6], [8] can be used to show the following result (which, however, will not be needed in our main task of completing the proof of Theorem 3.2).

THEOREM 3.5. *If* $L \subseteq [\frac{1}{5}, 1]$, *then* $\mathrm{BFD}(L) = \mathrm{FFD}(L)$.

Notice that the examples given in Figs. 5 and 6 show that the lower bounds of $\frac{1}{6}$ and $\frac{1}{5}$ in Theorems 3.4 and 3.5 are best possible.

4. First-fit decreasing upper bounds. In the previous section we reduced the proof of Theorem 3.2 to the task of showing that if $L \subset (\frac{2}{11}, 1]$, then $\mathrm{FFD}(L) \leqq \frac{11}{9}L^* + 4$. In this section we shall indicate how this can be done, and prove a simpler upper bound for the case when $L \subset (0, \frac{1}{2}]$.

The strategy behind such proofs is basically the same as the one used in § 2 for FF and BF upper bounds. Essentially, a "weighting function" is defined which assigns real number values or "weights" to the elements of L, depending on their size, in such a way that

(i) The total "weight" of all the elements in the list L is no less than a fixed constant c short of the number of bins used in the particular packing under consideration (e.g., FF or FFD).

(ii) The total weight of any legally packed bin must be less than some fixed constant r.

For FF we had $r = \frac{17}{10}$ and $c = 2$; for FFD we shall have $r = \frac{11}{9}$ and $c = 4$.

However, the actual details of the proof for FFD are considerably more complex than for FF and BF, requiring the introduction of a number of new concepts. Rather than burden the reader with this long[3] and detailed proof, we shall attempt to illustrate the basic ideas involved by describing in detail the major techniques used in the proof of a slightly simpler result, followed by an indication of the method for extending that proof to a proof of Theorem 3.2. Complete details can be found in Johnson [8].

THEOREM 4.1. *For all lists* $L \subset (0, \frac{1}{2}]$, $\mathrm{FFD}(L) \leqq \frac{71}{60}L^* + 5$.

[3] Exceeding 75 pages.

Remark. Theorem 4.1 gives the best bound possible, as can be seen from Fig. 8, which gives optimal and FFD packings of lists L for which $\mathrm{FFD}(L) = \frac{71}{60}L^*$, even though all elements in L are less than $\frac{1}{2}$, in fact, less than $\frac{1}{3}$. The ε in the figure must satisfy $0 < \varepsilon \leqq \frac{5}{87}$. We thus will be able to conclude that

$$\lim_{k\to\infty} R_{\mathrm{FFD}}^{1/2}(k) = \lim_{k\to\infty} R_{\mathrm{FFD}}^{1/3}(k) = \frac{71}{60},$$

in the terminology of § 2.

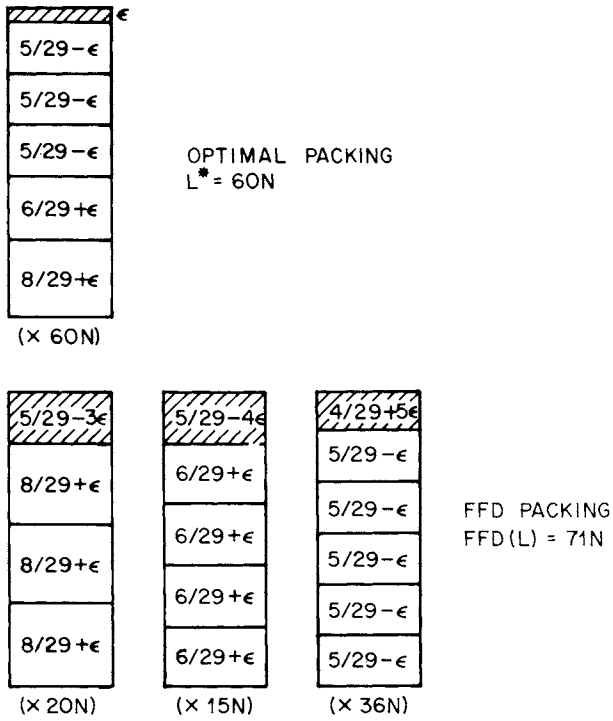

FIG. 8. *A 71/60 example*

We first give an overview of the proof, and then proceed to fill in many of the details. To begin with, the weighting function W which will be used is not really a function of *elements*, but rather a function of *sets* of elements,

$$W: 2^L \to Q \text{ (the rational numbers)}.$$

W will be defined in terms of two auxiliary functions.

$$w_1: L \to Q \quad \text{and} \quad w_2: L \times L \to Q.$$

The definitions of w_1 and w_2 will be made precise when we give the details of the proof.

Given a set of elements $X \subseteq L$, we obtain the weight $W(X)$ as follows: for any partition π of X into one- and two-element sets, let

$\pi(1) = \{x : \{x\} \in \pi\}$, (the set of elements which are in one-element sets in the partition π)

$\pi(2) = \{(x, y) : \text{index}(x) < \text{index}(y)$ and $\{x, y\} \in \pi\}$, (a set of ordered pairs corresponding to the two-element sets in π).

Then let

$$w_{12}(\pi) = \sum_{x \in \pi(1)} w_1(x) + \sum_{(x,y) \in \pi(2)} w_2(x, y).$$

Finally we define $W(X) = \min_\pi \{w_{12}(\pi)\}$, where π ranges over all partitions of X into one- and two-element sets. An elementary consequence of the definition as given so far is that W is subadditive, i.e.,

$$W\left(\bigcup_{i=1}^{k} X_i\right) \leqq \sum_{i=1}^{k} W(X_i).$$

Having defined the weighting function W, we can then divide the remainder of the proof of Theorem 4.1 into the two parts indicated at the beginning of the section. We shall state each part as a lemma, with the first given in sufficient generality so that it can also be used in the proof of the $\frac{11}{9}$ result.

LEMMA 4.2. *For any integer* $N \geqq 4$ *and* $L \subset (1/N, \frac{1}{2}]$,

$$W(L) \geqq \text{FFD}(L) - N + 2.$$

LEMMA 4.3. *If* $X \subseteq (\frac{1}{7}, \frac{1}{2}]$ *is any set of elements whose sum does not exceed* 1, *then* $W(X) \leqq \frac{71}{60}$.

By combining these two lemmas with the subadditivity of W, we can conclude that for any $L \subset (\frac{1}{7}, \frac{1}{2}]$, and X_i the set of elements in the ith bin of a given optimal packing of L, we have

$$\text{FFD}(L) - 5 \leqq W(L) \leqq \sum_{i=1}^{L^*} W(X_i) \leqq \tfrac{71}{60} L^*.$$

Thus Theorem 4.1 will be proved, since by Lemma 3.3 we *can* restrict our attention to lists $L \subseteq (\frac{1}{7}, \frac{1}{2}]$ when proving this upper bound.

We now begin a proof of Lemma 4.2, during which we will provide the remaining details of the definition of W. Let us call $x \in L$ a *k-piece* if $x \in (1/(k+1), 1/k]$. We shall also refer to 2-pieces as B-pieces, 3-pieces as C-pieces, etc. By a *k-bin* we mean a bin whose largest element is a k-piece.

Define $w_1(x) = \lfloor 1/x \rfloor^{-1}$. Thus if x is a k-piece, $w_1(x) = 1/k$. Let BASIC denote the set $\{x \in L$: for some k, x is a k-piece and is in a k-bin in the FFD packing of $L\}$.

CLAIM 4.2.1. *If* $L \subset (1/N, \frac{1}{2}]$, *then*

$$\sum_{x \in \text{BASIC}} w_1(x) \geqq \text{FFD}(L) - \sum_{j=2}^{N-1} \frac{j-1}{j}.$$

Proof. Note that for each k, $2 \leqq k < N$, all k-bins, except possibly for the last (rightmost) k-bin, must contain k k-pieces. For instance, every B-bin, that is, every 2-bin, must contain 2 B-pieces, except possibly for the rightmost one, which may contain only one B-piece. Thus, with the possible exception of the rightmost k-bin, all k-bins must contain elements of BASIC whose total w_1-weight is at least $k(1/k) = 1$. Since even the last k-bin must contain one k-piece belonging to BASIC, its deficiency can be at most $(k-1)/k$. This proves the claim.

As a consequence of Claim 4.2.1, we could satisfy Lemma 4.2 by merely defining $W(X) = w_1(X) = \sum_{x \in X} w_1(x)$. The reason we do *not* do this, but instead introduce w_2, is in order that we can prove Lemma 4.3. There are many sets of elements X whose sum does not exceed 1 and yet for which $w_1(x) > \frac{71}{60}$. This is not surprising in light of the fact that there are probably many elements from L which are not in BASIC, so that in fact $w_1(L)$ is probably much larger than w_1(BASIC) and hence much larger than FFD(L). Thus w_1 is in a sense "overcharging" the bins of the FFD packing.

The elements of SURPLUS = L − BASIC can thus be considered excess baggage in the weight calculated by w_1. The purpose of w_2 is to enable us to avoid counting this unneeded contribution to the total weight by SURPLUS. Given a pair of elements, w_2 will do this by "discounting" the w_1-weight of the second element by an appropriate amount if certain *discounting relations* are satisfied by the members of the pair. The relations can be generally described as follows: (x, y) is said to obey *relation* k if x is a k-piece and $kx + y \leqq 1$. We define w_2 by

$$w_2(x, y) = \begin{cases} w_1(x) + [(k-1)/k]w_1(y) & \text{if } (x, y) \text{ obeys relation } k, \\ w_1(x) + w_1(y) & \text{otherwise.} \end{cases}$$

Another way of looking at w_2 is to note that if (x, y) obeys relation k, then $w_1(\{x, y\}) - w_2(x, y) = w_1(y)/k$, and y has been discounted by a factor of $1/k$.

As a concrete example, suppose x is a B-piece (2-piece), y a C-piece, and $2x + y \leqq 1$. Then $w_1(\{x, y\}) = \frac{1}{2} + \frac{1}{3} = \frac{5}{6}$ and $w_2(x, y) = \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{3} = \frac{2}{3} = \frac{5}{6} - \frac{1}{2} \cdot \frac{1}{3}$. If $2x + y > 1$, then $w_2(x, y) = w_1(\{x, y\}) = \frac{5}{6}$.

We are now going to show that the "discount" due to using w_2 instead of w_1 on a pair of elements actually corresponds to an element in SURPLUS, or at least a portion of one, so that, modulo certain edge effects, the lower bound proven for w_1(BASIC) also holds for $w_{12}(\pi)$, where π is *any* partition of L into one- and two-element sets. Formally, we have Claim 4.2.2.

CLAIM 4.2.2. *If* $N \geqq 4$ *and* π *is a partition of* $L \subseteq (1/N, \frac{1}{2}]$ *into one- and two-element sets, then*

$$w_{12}(\pi) \geqq w_1(\text{BASIC}) - \sum_{j=3}^{N-1} \frac{1}{j}.$$

Lemma 4.2 will follow from Claims 4.2.1 and 4.2.2, since together they tell us that for all such π,

$$w_{12}(\pi) \geqq \text{FFD}(L) - \sum_{j=2}^{N-1} \frac{j-1}{j} - \sum_{j=3}^{N-1} \frac{1}{j} \geqq \text{FFD}(L) - N + 2,$$

and so $W(L) = \min_\pi w_{12}(\pi)$ must also exceed that lower bound.

Proof of Claim 4.2.2. Let R_k, $2 \leqq k \leqq N - 2$, be the set of all pairs in the partition π which obey relation k. Then, since no pairs in π can obey any relation $k' > N - 2$, $R = \bigcup_{k=2}^{N-2} R_k$ is the set of all pairs in π obeying discounting relations. If we let DISCOUNT$(x, y) = w_1(x) + w_1(y) - w_2(x, y)$, and DISCOUNT$(X)$ $= \sum_{(x,y) \in X}$ DISCOUNT(x, y) for any set of pairs X, we then have

$$w_{12}(\pi) = w_1(\text{BASIC}) + w_1(\text{SURPLUS}) - \text{DISCOUNT}(R),$$

and consequently all we need prove is that

$$w_1(\text{SURPLUS}) \geqq \text{DISCOUNT}(R) - \sum_{j=3}^{N-1} \frac{1}{j}. \tag{4.2.2a}$$

To show that (4.2.2a) holds, we construct a system of "billing" so that for each $(x, y) \in R$, DISCOUNT(x, y) is billed to some specific element of SURPLUS (or perhaps divided among a number of such elements), and no element of SURPLUS is billed for a total of more than its w_1-weight. A small number of DISCOUNTS will go unbilled and these will account for the $\sum(1/j)$ term. More formally, we construct a billing map BILL$: R \times \text{SURPLUS} \to (0, \frac{1}{6}]$ and a set UNBILLED such that

$$\text{DISCOUNT(UNBILLED)} \leqq \sum_{j=2}^{N-2} 1/(j+1). \tag{4.2.2b}$$

For all $(x, y) \in R - \text{UNBILLED}$,

$$\sum_{z \in \text{SURPLUS}} \text{BILL}((x, y), z) \geqq \text{DISCOUNT}(x, y). \tag{4.2.2c}$$

For all $z \in \text{SURPLUS}$,

$$\sum_{(x,y) \in R} \text{BILL}((x, y), z) \leqq w_1(z). \tag{4.2.2d}$$

Inequality (4.2.2a), and hence Claim 4.2.2 will then follow from (4.2.2b)–(4.2.2d).

To keep this paper to a reasonable length, we shall not present the billing procedure in all its intricacies [8]. However, we will present the basic idea behind it and an indication of why additional intricacies are necessary.

Initially we set BILL$((x, y), z) = 0$ for all $(x, y) \in R$, $z \in \text{SURPLUS}$. As we proceed, some of these values will be reset. At any given point in time, $z \in \text{SURPLUS}$ will have been *charged* the current value of $\sum_{(x,y) \in R} \text{BILL}((x, y), z)$. For $(x, y) \in R$, we will say that DISCOUNT(x, y) has been *billed* if

$$\sum_{z \in \text{SURPLUS}} \text{BILL}((x, y), z) \geqq \text{DISCOUNT}(x, y).$$

We shall treat each R_k in turn, defining a set UNBILLED$_k \subseteq R_k$ and then billing the DISCOUNT for each pair in $R_k - \text{UNBILLED}_k$ in such a way that no element of SURPLUS will have been charged more than its w_1-weight. UNBILLED will be defined as $\bigcup_{k=2}^{N-2} \text{UNBILLED}_k$. And we will have DISCOUNT (UNBILLED$_k$) $\leqq 1/k + 1$, $2 \leqq k \leqq N - 2$. The basic idea involved in the processing of R_k can be explained as follows: Assume that

(G1) no $z \in \text{SURPLUS}$ which is in a k'-bin, $k' \geqq k$, in the FFD-packing has yet been charged more than 0,

(G2) no member of any pair in R_k is in a k'-bin, $k' < k$.

All the billing we shall do in this case will be to elements of SURPLUS$_k$, the members of SURPLUS which are in k-bins.

First take all pairs in R_k and relabel them (x_i, y_i) in order of increasing index (with respect to the original list) of their second components. We will thus have index$(y_1) <$ index$(y_2) < \cdots <$ index(y_m), where $m = |R_k|$, and hence $y_1 \geqq$

$y_2 \geqq \cdots \geqq y_m$. Note that the x_i's and y_i's are all distinct since the $\{x_i, y_i\}$'s form a partition of R_k and hence are disjoint.

Suppose we can construct a 1–1 map

$$g:\{y_{kj}:1 \leqq j \leqq \lceil m/k \rceil\} \to \text{SURPLUS}_k$$

such that for all $y_i \in \text{Domain}(g)$,

(G3) $\text{index}(g(y_i)) \leqq \text{index}(y_i)$.

We can then use g to define BILL for elements of R_k. For $1 \leqq j \leqq \lceil m/k \rceil$ and $0 \leqq i \leqq k-1$, let

$$\text{BILL}((x_{kj+i}, y_{kj+i}), g(y_{kj})) = \text{DISCOUNT}(x_{kj+i}, y_{kj+i}).$$

(For $j = \lceil m/k \rceil$ and $i > m - k\lceil m/k \rceil$ the definition will be vacuous.)

If we let $\text{UNBILLED}_k = \{(x_i, y_i):1 \leqq i < k\}$ we thus have for all $(x, y) \in R_k - \text{UNBILLED}_k$ that $\text{DISCOUNT}(x, y)$ has been billed. Moreover,

$$\text{DISCOUNT}(\text{UNBILLED}_k) = \sum_{i=1}^{k-1} \text{DISCOUNT}(x_i, y_i) \leqq (k-1)\frac{1}{k} \cdot \frac{1}{k+1} < \frac{1}{k+1}.$$

Finally, the only elements charged are $g(y_{kj})$, and since g is 1–1, the most $g(y_{kj})$ is charged is

$$\begin{aligned}\sum_{(x,y)\in R_k} \text{BILL}((x, y), g(y_{kj})) &= \sum_{i=0}^{k-1} \text{DISCOUNT}(x_{kj+i}, y_{kj+i}) \\ &= \frac{1}{k}\sum_{i=0}^{k-1} w_1(y_{kj+i}) \\ &\leqq \frac{1}{k}[k \cdot w_1(y_{kj})] = w_1(y_{kj}) \leqq w_1(g(y_{kj}))\end{aligned}$$

by (G3) because $\text{index}(y_{kj+k-1}) > \cdots > \text{index}(y_{kj}) \geqq \text{index}(g(y_{kj}))$ and L is in decreasing order.

If the above held for all k, $2 \leqq k \leqq N-2$, and if g were 1–1 throughout the composite range $[y_{kj}:1 \leqq j \leqq \lceil m/k \rceil, 2 \leqq k \leqq N-2\}$, we would have properties (4.2.2b) through (4.2.2d), and hence Claim 4.2.2 would be proved.

How might we define g so that the above *does* hold? In the case when both (G1) and (G2) hold, as they must trivially for $k = 2$, the process is fairly straightforward.

Observe that since all $(x_i, y_i) \in R_k$ obey relation k, we have $y_i + kx_i \leqq 1$, $1 \leqq i \leqq m$, and hence, by the indexing of the pairs, $y_i + kx_j \leqq 1$ for $1 \leqq j \leqq i$.

Let us now look at the FFD-packing again, in particular the k-bins. (See Fig. 9.) There must be at least $\lceil m/k \rceil$ k-bins, since by assumption (G2) there are at least $|R_k| = m$ k-pieces in k'-bins for $k' \geqq k$ and hence in k-bins. We label the bottom k elements in each k-bin from top to bottom and right to left, as shown in Fig. 9. Then b_k must be a k-piece, as are all the labeled elements with higher index. The remaining elements in the bin containing b_k (the *b_k-bin*) need not all be k-pieces. Indeed, some may not even exist, if this is the last bin in the packing, in

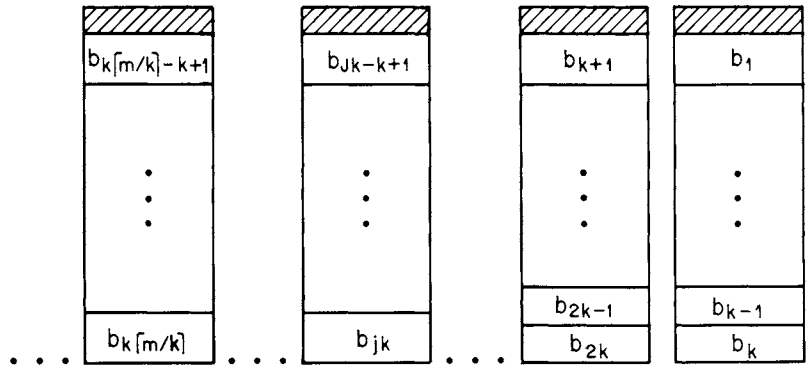

FIG. 9. *K-bins of* FFD *packing labeled for Lemma* 4.2.2

which case we set the corresponding $b_i = 0$. By the way FFD is executed, we must have

$$b_{k\lceil m/k\rceil} \geqq b_{k\lceil m/k\rceil-1} \geqq \cdots \geqq b_3 \geqq b_2 \geqq b_1.$$

Now all the x_i's occur in this list; thus for no $j \leqq k\lceil m/k\rceil$ can we have $b_j > x_i$, for all i, $1 \leqq i \leqq j$, as that would put one too many x_i's to the right of b_j in the sequence. Thus for each j there exists an $i(j) \leqq j$ such that $x_{i(j)} \geqq b_j$. We can thus conclude that for all h, j, $1 \leqq j \leqq h \leqq \lceil m/k\rceil$,

$$y_{kh} + \sum_{i=1}^{k} b_{kj-k+i} \leqq y_{kh} + kb_{kj} \leqq y_{kh} + kx_{i(kj)} \leqq 1.$$

Consequently y_{kh} would fit as the $(k+1)$st element in any of the bins to the right of and including the b_{kh}-bin. Using this fact we can define our function g as follows.

If y_k is in a k-bin in the FFD-packing, then we must have $y_k \in \text{SURPLUS}_k$. Let $g(y_k) = y_k$. If not, then y_k must by assumption be in some bin to the right of the b_k-bin. Since y_k would have fit in that bin unless it already contained $k+1$ elements, the b_k-bin must have contained that many elements when y_k was assigned, one of which must be an element of SURPLUS_k and have lower index than y_k. Let $g(y_k)$ be the largest such element.

Note that in both cases, $g(y_k) \in \text{SURPLUS}_k$ and obeys (G3) for $i = k$, i.e., $\text{index}(g(y_k)) \leqq \text{index}(y_k)$.

Continuing by induction, assume that values for $g(y_{kh})$, $1 \leqq h \leqq j \leqq [m/k]$, have been assigned and are all distinct elements of SURPLUS_k obeying (G3). If y_{kj} is in a k-bin, it cannot be $g(y_{kh})$ for any $h < j$, since by (G3) and the labeling of the y_i's we have $\text{index}(g(y_{kh})) \leqq \text{index}(y_{kh}) < \text{index}(y_{kj})$. So in this case we again define $g(y_{kj}) = y_{kj}$. If y_{kj} is not in a k-bin, then it must have gone to the right of the b_k- through b_{kj}-bins, into each of which it would have fit as the $(k+1)$st element. Hence all j bins must contain elements of SURPLUS with index lower than that of y_{jk}. Since at most $j-1$ of them can have yet been assigned to the range of g, there is at least one still unassigned and we can let such an element of SURPLUS be $g(y_{kj})$. This maintains the 1–1 property of g and insures that (G3) will hold for $i = kj$.

Thus by induction we have defined our map

$$g:\{y_{kj}:1 \leqq j \leqq \lceil m/k \rceil\} \to \text{SURPLUS}$$

obeying property (G3) throughout its domain.

The above analysis depended on assumptions (G1) and (G2), which, as we have said, clearly hold for $k = 2$. We might thus hope to proceed by induction. In our billing procedure, only elements of SURPLUS_k received new charges, so (G1) will continue to hold when we begin to process R_{k+1}.

However, there is no guarantee that (G2) will hold for any $k > 2$. This is what gives rise to complications. If $(x, y) \in R_k$ and x is not in a k-bin, then it must be in a k'-bin, $k' < k$, and hence a member of SURPLUS. If x has not yet been charged, we can bill DISCOUNT(x, y) to x. If it has been charged more than 0, then x must be $g(z)$ for some z, with $z \leqq x$ by (G1), and z may be a k-piece in a k-bin. The more intricate argument here omitted shows how to modify our billing procedure to take advantage of such possibilities and still guarantee that (4.2.2b)–(4.2.2d) hold, and hence Claim 4.2.2, will hold. Q.E.D.

We have already seen that Lemma 4.2 follows from Claim 4.2.1 and 4.2.2. To complete the proof of Theorem 4.1, we must now turn our attention to Lemma 4.3, which says that for any set $X \subseteq (\frac{1}{7}, \frac{1}{2}]$ whose sum does not exceed 1, $W(X) \leqq \frac{71}{60}$. Since $W(X)$ depends just on the types of the elements in X (e.g., B, C, $\cdots$, etc.) and which discounting relations are satisfied (not on the precise *values* of the elements), it is easy to see that there are a relatively small finite number of possible configurations to consider. We shall illustrate the type of calculation necessary by treating several typical cases, leaving the remaining 70-odd, more or less routine, cases to the ambitious reader.[4]

(i) $X = \{B_1, C_2, C_3\}$, i.e., X consists of one B-piece and two C-pieces with $C_2 \geqq C_3$. Then

$$W(X) \leqq w_1(X) = w_1(B_1) + w_1(C_2) + w_1(C_3) = \tfrac{1}{2} + \tfrac{1}{3} + \tfrac{1}{3} = \tfrac{7}{6} < \tfrac{71}{60}$$

(ii) $X = \{B_1, B_2, E_3, F_4\}$. Here $w_1(X) = \frac{1}{2} + \frac{1}{2} + \frac{1}{5} + \frac{1}{6} = \frac{41}{30} > \frac{71}{60}$ so the partition of X into 1-element sets is not adequate for determining W. Note, however, that

$$B_1 + E_3 \leqq 1 - B_2 - F_4 < 1 - \tfrac{1}{3} - \tfrac{1}{7} = \tfrac{11}{21},$$

which implies

$$2B_1 + E_3 \leqq \tfrac{22}{21} - \tfrac{1}{6} < 1,$$

so that (B_1, E_3) obeys relation 2. Hence we get a discount of $\frac{1}{2} \cdot \frac{1}{5} = \frac{1}{10}$ here. Similarly (B_2, F_4) obeys relation 2 and we get an additional discount of $\frac{1}{12}$. Thus

$$W(X) \leqq \tfrac{41}{30} - \tfrac{1}{10} - \tfrac{1}{12} = \tfrac{71}{60},$$

as required.

(iii) $X = \{C_1, C_2, E_3, E_4, E_5\}$. This configuration is impossible since $C_i \in (\frac{1}{4}, \frac{1}{3}]$ and $E_j \in (\frac{1}{6}, \frac{1}{5}]$ imply $C_1 + C_2 + E_3 + E_4 + E_5 > 1$.

(iv) $X = \{C_1, D_2, E_3, E_4, E_5\}$. Then $W(X) \leqq w_1(X) = \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{5} + \frac{1}{5} = \frac{71}{60}$. Note that this is the configuration which occurs in the construction used to prove the lower bound for Theorem 4.1.

[4] Complete details may be found in [8].

When all possible cases are finally checked (there are only two which yield a bound of $\frac{71}{60}$; all other cases are bounded by $\frac{7}{6}$), the proof of Theorem 5 follows at once from the preceding remarks.

It is not difficult to show that if no element of L exceeds $\frac{8}{29}$, then the preceding analysis proves $W(X) \leqq \frac{7}{6}$ for all sets X of elements in a legally packed bin. (Case (ii) cannot occur since there are now no B-pieces; in case (iv), additional discounting now becomes possible.) This observation leads us to the following corollary.

COROLLARY. (a) *For* $\alpha \in (\frac{8}{29}, \frac{1}{2}]$,

$$\lim_{k \to \infty} R^{\alpha}_{\mathrm{FFD}}(k) = \tfrac{71}{60};$$

(b) *for* $\alpha \in (\frac{1}{4}, \frac{8}{29}]$,

$$\lim_{k \to \infty} R^{\alpha}_{\mathrm{FFD}}(k) = \tfrac{7}{6}.$$

The lower bound for (b) is proved with a construction similar to those used earlier, with the list $L = (a_1, a_2, \cdots, a_n)$ composed of $24N$ elements, half equal to $.25 + \varepsilon$ and half equal to $.25 - \varepsilon$, where $0 < \varepsilon < \frac{1}{12}$.

A similar (but more complicated) analysis can be given to establish Theorem 4.4.

THEOREM 4.4. *For* $\alpha \in (\frac{1}{5}, \frac{1}{4}]$,

$$\lim_{k \to \infty} R^{\alpha}_{\mathrm{FFD}}(k) = \tfrac{23}{20}.$$

The reader is referred to Johnson [8] for a much more complete discussion of the details needed to complete the proofs of these and other similar results.

As proposed at the beginning of this section, the techniques used in the proof of Theorem 4.1 form a significant part of the proof of Theorem 3.2, whose crucial assertion is that $\mathrm{FFD}(L) \leqq \frac{11}{9}L^* + 4$ for all lists $L \subseteq (\frac{2}{11}, 1]$.

The inclusion of A-pieces, that is, elements which exceed $\frac{1}{2}$, causes rather severe problems for the relatively simple calculations we were able to perform in proving Theorem 4.1. We are now forced to resort to somewhat more subtle ideas. The basic strategy is as follows.

(a) We assume we have a list $L = (a_1 \geqq a_2 \geqq \cdots \geqq a_n)$, consisting of A-, B-, C-, D- and E-pieces, where the E-pieces lie in $(\frac{2}{11}, \frac{1}{5}]$. Let P^* denote some optimal packing of L and let P_{FFD} denote the FFD-packing of L. Define $\mathscr{B}$ to be the set of non-A-pieces in L and

$$\mathscr{O} \equiv \{a_i \in \mathscr{B} : a_i \text{ is not in an A-bin in } P^*\},$$

$$\mathscr{F} \equiv \{a_i \in \mathscr{B} : a_i \text{ is not in an A-bin in } P_{\mathrm{FFD}}\}.$$

(b) A key idea now is to note that a packing of the non-A-bins in P_{FFD} is the same as if we had applied FFD directly to $\mathscr{F}$ alone. And since $\mathscr{F}$ does not contain any A-pieces, all the facts we proved before about W will hold for that packing. In particular, from Lemma 4.2 with $N = 6$ we have

$$(*) \qquad W(\mathscr{F}) \geqq \mathrm{FFD}(L) - |\mathrm{A}| - 4$$

where $|\mathrm{A}|$ denotes the number of A-pieces in L.

(c) However, we still face the problem that $\mathcal{O}$ need not contain the same elements as $\mathcal{F}$. Some elements that are not in A-bins in one packing may be in A-bins in the other. In addition, there is the fact that the number of A-bins is the same in both packings and must somehow be counted when we try to put a bound on FFD(L). To take care of these two problems, we introduce two functions:

$$f: L \to 2^{\mathcal{F}} \quad \text{and} \quad g: L \to Q.$$

They satisfy the two properties: (i) $\mathcal{F} = \bigcup_{i=1}^{n} f(a_i)$ and (ii) $|A| \geqq \sum_{i=1}^{n} g(a_i)$.

(d) f and g can be extended to set functions on subsets $X \subseteq L$ by

$$f(X) = \bigcup_{a \in X} f(a), \qquad g(X) = \sum_{a \in X} g(a).$$

We can then use a case analysis to establish the following critical inequality for the set of pieces X in any legally filled bin:

$$(**) \qquad W(f(X)) + g(X) \leqq \tfrac{11}{9}(y(X) + g(X)),$$

where

$$y(X) = \begin{cases} 1 & \text{if } X \text{ contains no A-pieces,} \\ 0 & \text{otherwise.} \end{cases}$$

We can say that the left-hand side of the inequality counts bins in P_{FFD} and the right-hand side does the same for P^* (and multiplies the result by $\frac{11}{9}$), since W counts non-A-bins in P_{FFD}, y counts them in P^*, and g counts the A-bins in both packings.

(e) Given this intuitive way of looking at (**), we observe that, if f and g satisfy (i) and (ii), respectively, and if property (**) holds for all possible X_i (the set of elements in the ith bin of P^*), Theorem 3.2 then follows by summation. For in this case we would have

$$\begin{aligned} W(\mathcal{F}) + g(L) &\leqq \bigcup_{i=1}^{L^*} W(f(X_i)) + \sum_{i=1}^{L^*} g(X_i) && \text{by subadditivity of } W \text{ and (i) and (ii)} \\ &\leqq \sum_{i=1}^{L^*} \tfrac{11}{9} y(X_i) + \tfrac{11}{9} g(L) && \text{by (**)} \\ &= \tfrac{11}{9}(L^* - |A| + g(L)) && \text{by definition of } y. \end{aligned}$$

Thus by (*),

$$\text{FFD}(L) - |A| - 4 + g(L) \leqq \tfrac{11}{9}(L^* - |A| + g(L)),$$

implying that

$$\begin{aligned} \text{FFD}(L) &\leqq \tfrac{11}{9} L^* - \tfrac{2}{9}(|A| - g(L)) + 4 \\ &\leqq \tfrac{11}{9} L^* + 4 \qquad \text{by (ii)}, \end{aligned}$$

which is just Theorem 3.2.

Unfortunately, the amount of space required to present the details necessary to establish the preceding remarks prohibit us from giving them here. The definitions of f and g are also somewhat complicated, although intuitively f assigns to

each piece a_i in L a subset of $\mathscr{F}$ for which a_i is, in a rough sense, "uniquely responsible", while g serves to count those A-bins which "collaborated" in this "responsibility". Needless to say, the actual arguments are considerably more complex than those given for Theorem 4.1. The interested reader is referred to Johnson [8] in which the complete details of these proof techniques may be found.

5. Concluding remarks. The four bin-packing algorithms studied in this paper are actually special cases of more general classes of algorithms which have been considered in some detail by Johnson [8], [9]. We mention here several relatively unexplored directions in this area which seem to be of some interest.

(i) What is the worst-case behavior for BFD and FFD when the lists L are restricted from above and below, i.e., $L \subseteq (a, \beta)$ for fixed $0 < \alpha \leqq \beta < 1$. The corresponding results for FF and BF are known and can be found in Johnson [8]. It appears that the precise bounds on this behavior will probably be rather complicated functions of α and β, depending on certain of their number-theoretic properties.

(ii) How do these various algorithms compare among themselves? For example, we have seen that $\mathrm{BFD}(L) \leqq \mathrm{FFD}(L)$ for $L \subseteq [\frac{1}{6}, 1]$. On the other hand, $\mathrm{BFD}(L)/\mathrm{FFD}(L) \geqq \frac{10}{9}$ can occur for lists L with arbitrarily large L^*. How large can this ratio be for large L^*? How small can it be? The same questions can be asked for other pairs of algorithms.

(iii) What is the trade-off between the effectiveness of an algorithm and the efficiency of implementation of the algorithm? For example, for a list L with n elements $\mathrm{FFD}(L)$ and $\mathrm{BFD}(L)$ are both bounded above by $\frac{11}{9}L^*$ and both can be implemented using $O(n \log n)$ operations. How well can an $O(n)$ algorithm perform? If we are willing to use an $O(n^2)$ algorithm, how close to L^* can we be guaranteed of coming?

(iv) It is possible to consider bins with differing capacities. How does the ordering of the bin sizes affect the number of bins required by the various algorithms under consideration? For example, by how much can ordering the bins *largest first* differ from the optimal ordering when FFD is applied?

(v) All the questions raised so far also apply (with suitable modifications) to *two-dimensional* bin packing. In view of potential applications, this direction would seem to warrant further investigation.

(vi) What is the *expected* behavior of these algorithms? For example, if the elements of L are chosen uniformly from $[0, 1]$, what is the expected value of $\mathrm{FF}(L)/L^*$? $\mathrm{FFD}(L)/L^*$? Simulation results on FF, BF, FFD, BFD [4], [8] indicate that $\mathrm{FFD}(L)$ and $\mathrm{BFD}(L)$ are almost always better than $\mathrm{FF}(L)$ and $\mathrm{BF}(L)$ for a random L, with BF occasionally slightly better than FF.

REFERENCES

[1] A. R. BROWN, *Optimum Packing and Depletion*, American Elsevier, New York, 1971.

[2] R. W. CONWAY, W. L. MAXWELL AND L. W. MILLER, *Theory of Scheduling*, Addison-Wesley, Reading, Mass., 1967.

[3] S. A. COOK, *The complexity of theorem-proving procedures*, Proc. 3rd Annual ACM Symp. on the Theory of Computing, 1971, pp. 151–158.

[4] J. CUNY, Private communication.

[5] S. EILON AND N. CHRISTOFIDES, *The loading problem*, Management Sci., 17 (1971), pp. 259–268.

[6] M. R. GAREY, R. L. GRAHAM AND J. D. ULLMAN, *Worst-case analysis of memory allocation algorithms*, Proc. 4th Annual ACM Symp. on the Theory of Computing, 1972, pp. 143–150.

[7] P. C. GILMORE AND R. E. GOMORY, *A linear programming approach to the cutting stock problem II*, Operations Res., 11 (1963), pp. 863–888.

[8] D. S. JOHNSON, Near-optimal bin packing algorithms, Doctoral thesis, Mass. Inst. of Tech., Cambridge, Mass., 1973.

[9] ———, *Fast algorithms for bin packing*, J. Comput. Systems Sci., 8 (1974), pp. 272–314.

[10] R. M. KARP, *Reducibility among combinatorial problems*, Complexity of Computer Computations, R. E. Miller and J. W. Thatcher, eds., Plenum Press, New York, 1972, pp. 85–104.