WORST-CASE ANALYSIS OF MEMORY ALLOCATION ALGORITHMS

M. R. Garey and R. L. Graham
Bell Telephone Laboratories, Incorporated
Murray Hill, New Jersey

J. D. Ullman†
Princeton University
Princeton, New Jersey

ABSTRACT

Various memory allocation problems can be modeled by the following abstract problem. Given a list $A = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ of real numbers in the range $(0,1]$, place these in a minimum number of "bins" so that no bin holds numbers summing to more than 1. We let A^* be the smallest number of bins into which the numbers of list A may be placed. Since a general placement algorithm for attaining A^* appears to be impractical, it is important to determine good heuristic methods for assigning numbers of bins. We consider four such simple methods and analyze the worst-case performance of each, closely bounding the maximum of the ratio of the number of bins used by each method applied to list A to the optimal quantity A^*.

I. INTRODUCTION

Given a list $A = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ of real numbers in the range $(0,1]$, find the minimum number N of "bins" for which there is a mapping $f: \{1,\ldots,n\} \rightarrow \{1,\ldots,N\}$ such that for all i, the sum of those α_j for which $f(\alpha_j) = i$ does not exceed 1. This least N is termed A^*.

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

(1) Table Formating. Let the "bins" be computer words of fixed size k. Suppose there are items of data (e.g., bit string of length 6, character string of 3 bytes, half word integer) requiring $k\alpha_1, \ldots, k\alpha_n$ bits, respectively. It is desirable to place the data in as few words as possible. The minimum number of words is A^*, where A is the list $(\alpha_1, \ldots, \alpha_n)$.

(2) Prepaging. Here, the bins are pages and the numbers in the list A represent segments of the program which should appear on a single page, e.g., inner loops, arrays.

(3) File Allocation. It is desired to place files of varying sizes on as few tracks of a disc as is possible.

The calculation of an optimal solution to the assignment problem mentioned is in general too time consuming to be considered a realistic goal. In practice one must find heuristics that are likely to yield a good assignment. Several simple algorithms have been suggested in [3,4]. In particular, we will consider the following four placement algorithms.

(1) First Fit. Initially, all bins are "filled to level" 0. Consider $\alpha_1, \ldots, \alpha_n$ in that order. To consider α_i, find the least j such that B_j is filled to a level $\beta \leq 1 - \alpha_i$. Place α_i in B_j. B_j is now filled to level $\beta + \alpha_i$.

(2) Best Fit. Initially, all bins are filled to level 0. Consider $\alpha_1, \ldots, \alpha_n$ in that order. To consider α_i, find that bin B_j such that B_j is filled to level $\beta \leq 1 - \alpha_i$ and $\beta + \alpha_i$ is as large as possible. Place α_i in B_j. B_j is now filled to level $\beta + \alpha_i$.

(3) First Fit Decreasing. Order $(\alpha_1, \ldots, \alpha_n)$ largest first, then apply (1).

(4) Best Fit Decreasing. Order $(\alpha_1, \ldots, \alpha_n)$ largest first, then apply (2).

Let A^{FF}, A^{BF}, A^{FFD}, and A^{BFD} be the number of bins filled to level greater than zero by the four algorithms above, respectively.

Our approach to evaluating the performance of these simple algorithms is to determine bounds upon the ratios A^{FF}/A^*, A^{BF}/A^*, A^{FFD}/A^*, and A^{BFD}/A^*. To this end, let $R^{FF}(k)$, $R^{BF}(k)$, $R^{FFD}(k)$, and $R^{BFD}(k)$ be the maxima, over all lists A such that $A^* = k$, of A^{FF}/A^*, A^{BF}/A^*, A^{FFD}/A^*, and A^{BFD}/A^*. The following is a summary of our main results.

(1) $\lim_{k\to\infty} R^{FF}(k) = 17/10.$

(2) $\lim_{k\to\infty} R^{BF}(k) \geq 17/10.$

(3) $11/9 \leq \lim_{k\to\infty} R^{FFD}(k) \leq 5/4.$

(4) $11/9 \leq \lim_{k\to\infty} R^{BFD}(k) \leq 5/4.$

One might question the validity of evaluating an algorithm by its worst-case performance, rather than, say, its average performance. However, since the algorithm may be used in a variety of applications, and since the probability distribution for

†Work of this author supported by NSF Grant GJ-1052 to Princeton University.

the lists A is not usually well known, such statistical evaluations have only limited usefulness. Intuitively, a "mechanism" which causes an algorithm to have a high worst-case ratio might well be expected to manifest itself at least partially in real applications. Some experiments with the case in which the elements of the lists are chosen with uniform distribution on $(0,1]$ have tended to confirm the hypothesis that worst-case analysis is a valid performance measure for the type of algorithms considered here [5]. In particular, FFD and BFD appeared equally good in the experiments and substantially better than FF and BF, as might be conjectured from our worst-case results.

II. FIRST FIT AND BEST FIT

Theorem 1: For any $\varepsilon > 0$, if k is sufficiently large, then $R^{FF}(k) \geq 17/10 - \varepsilon$ and $R^{BF}(k) \geq 17/10 - \varepsilon$.

Proof: A will consist of numbers in three regions. The elements in the regions will have sizes close to 1/6, 1/3 and 1/2, respectively. The number of elements in the three regions will be the same, and those of the first region precede those of the second which precede those of the third in the list A.

Let N be a number 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 i^{th} block of region 1 be $a_{1i}, a_{2i}, \cdots, a_{10i}$. These numbers are given by the following formulas. Let δ_i be $\delta\ 18^{(N/17 - i)}$, for $1 \leq i \leq N/17$. Then:

$$a_{1i} = \frac{1}{6} + 33\delta_i$$

$$a_{2i} = \frac{1}{6} + 3\delta_i$$

$$a_{3i} = a_{4i} = \frac{1}{6} - 7\delta_i$$

$$a_{5i} = \frac{1}{6} - 13\delta_i$$

$$a_{6i} = \frac{1}{6} + 9\delta_i$$

$$a_{7i} = a_{8i} = a_{9i} = a_{10i} = \frac{1}{6} - 2\delta_i$$

Let the first 10 N/17 numbers in the list A be $a_{11}, \cdots, a_{10,1}, a_{12}, \cdots, a_{10,2}, \cdots$. We notice that $a_{1i} + \cdots + a_{5i} = 5/6 + 3\delta_i$, and $a_{6i} + \cdots + a_{10i} = 5/6 + \delta_i$. 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 used to fill bins. To make this observation, we need only note that a_{5i}, 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 $5/6 + \delta_{i-1} = 5/6 + 18\delta_i$. Also, the smallest of $a_{6i}, \cdots, a_{10i}$, which is $1/6 - 2\delta_i$ will not fit in bin 2i - 1, which has contents totaling $5/6 + 3\delta_i$.

Thus, the N/17 blocks in region 1 fill up 2N/17 cells. We now turn to region 2. Here, the numbers are all about 1/3, and they are again divided into N/17 blocks. Let the i^{th} block of region 2 be $b_{1i}, \cdots, b_{10i}$. The numbers $b_{11}, \cdots, b_{10,1}, b_{12}, \cdots, b_{10,2}, \cdots$ follow those of region 1 in the list A. The values of the numbers in block i are given by:

$$b_{1i} = \frac{1}{3} + 46\delta_i$$

$$b_{2i} = \frac{1}{3} - 34\delta_i$$

$$b_{3i} = b_{4i} = \frac{1}{3} + 6\delta_i$$

$$b_{5i} = \frac{1}{3} + 12\delta_i$$

$$b_{6i} = \frac{1}{3} - 10\delta_i$$

$$b_{7i} = b_{8i} = b_{9i} = b_{10i} = \frac{1}{3} + \delta_i$$

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

$$\frac{2}{3} + 12\delta_i \quad \frac{2}{3} + 12\delta_i \quad \frac{2}{3} + 2\delta_i$$

$$\frac{2}{3} + 2\delta_i \quad \frac{2}{3} + 2\delta_i$$

Thus, $b_{6i} = 1/3 - 10\delta_i$ cannot fall into either of the first two bins, and $b_{2i} = 1/3 - 34\delta_i$ cannot fall into any of the bins for previous blocks, since these are all filled to at least level $2/3 + 2\delta_{i-1} = 2/3 + 36\delta_i$.

The third region consists of 10N/17 numbers, each $1/2 + \delta$. These complete the list A and clearly fill one bin each. The total number of bins filled by the first fit algorithm is thus 2N/17 from region 1, 5N/17 from region 2 and 10N/17 from region 3, a total of N bins.

However, we may use the list A to fill 10N/17 + 1 bins as follows. All but two of these bins have one of the elements $1/2 + \delta$. These bins are then filled with one of the following combinations:

(1) $a_{ji} + b_{ji}$ for some $3 \leq j \leq 10$ and $1 \leq i \leq \frac{N}{17}$.

(2) $a_{ij} + b_{2i}$ for some $1 \leq i \leq \frac{N}{17}$.

(3) $a_{2i} + b_{1(i+1)}$ for some $1 \leq i \leq \frac{N}{17}$.

This leaves b_{11}, $a_{2(N/17)}$ and one number $1/2 + \delta$ which may fill the remaining two bins in several ways. We have thus shown that

$A^* \leq 10N/17 + 1$, so $A^{FF}/A^* \geq 17N/(10N+17)$ and $A^{BF}/A^* \geq 17N/(10N+17)$. By selecting sufficiently large N, we can make this ratio be bounded below by 17/10 - ε for any positive ε.

We will now show that 17/10 is the asymptotic least upper bound of the ratio $R^{FF}(k)$. The general strategy will be to consider the two packings of bins governed by (i) the optimal algorithm and (ii) the first fit algorithm. We will arrange a system of "payments," where each bin in the optimal assignment "pays" a certain amount for each number it contains, in such a way that no bin pays more than 17/10 units. Each bin in the first fit assignment is "paid" for each number it contains, and it will be shown that with exceptions totaling at most 3 units, each bin is paid at least one unit.

A "conservation of payments" argument allows us to conclude that $17/10\ A^* \geq A^{FF} - 3$, and thus that $A^{FF}/A^* \leq 17/10 + O(1/A^*)$.

Lemma 1: If numbers are assigned to bins by the first fit algorithm, there is at most one non-empty bin that is not more than half full.

Proof: If not, then the one to the right was filled illegally.

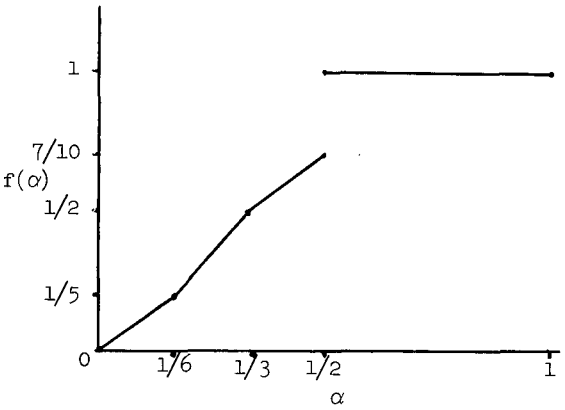

Fig. 1 The Function $f(\alpha)$

Formally:

$$f(\alpha) = \frac{6}{5}\alpha, \qquad \text{for } 0 \leq \alpha \leq \frac{1}{6}$$

$$f(\alpha) = \frac{9}{5}\alpha - \frac{1}{10}, \qquad \text{for } \frac{1}{6} \leq \alpha \leq \frac{1}{3}$$

$$f(\alpha) = \frac{6}{5}\alpha + \frac{1}{10}, \qquad \text{for } \frac{1}{3} \leq \alpha \leq \frac{1}{2}$$

$$f(\alpha) = 1, \qquad \text{for } \frac{1}{2} < \alpha \leq 1$$

Lemma 2: Let some bin be filled with $(\alpha_1, \alpha_2, \ldots, \alpha_n)$. Then $\sum_{i=1}^{n} f(\alpha_i) \leq 17/10$.

Proof: If $\alpha \leq 1/2$, then $f(\alpha)/\alpha \leq 3/2$; the extreme ratio is reached only when $\alpha = 1/3$ and is less otherwise. Thus, the lemma is immediate unless one α_i is greater than 1/2. We may take this one to be α_1, and must now show that if $\sum_{i=2}^{n} \alpha_i < 1/2$, then $\sum_{i=2}^{n} f(\alpha_i) \leq 7/10$.

It should be noted that since the slope of $f(\alpha)$ is the same in the region [0,1/6] and [1/3, 1/2], any α_i which is in the second regions can be replaced without loss of generality by two numbers of 1/3 and α_i - 1/3, respectively. We therefore assume that $\alpha_i \leq 1/3$ for $2 \leq i \leq n$. Moreover, if α_j and α_k are both equal to or less than 1/6, they can be combined into one, and $\sum_i f(\alpha_i)$ will not decrease; in fact it may increase. Thus, we assume that all but at most one of the α_i's are in the range (1/6, 1/3].

We have thus reduced the proof to the consideration of two cases:

(1) $n = 2$; $\frac{1}{6} \leq \alpha_2 \leq \frac{1}{3}$, and

(2) $n = 3$; $\alpha_1 \leq \frac{1}{6} \leq \alpha_2 \leq \alpha_3 \leq \frac{1}{3}$.

In case (1), $f(\alpha_1) + f(\alpha_2) = 9/5\ (\alpha_1+\alpha_2) - 1/5$. Since $\alpha_1 + \alpha_2 \leq 1/2$, we have $f(\alpha_1) + f(\alpha_2) \leq 7/10$, as desired. In case (2), $f(\alpha_1) + f(\alpha_2) + f(\alpha_3) \leq 6/5\ \alpha_1 + 9/5\ (\alpha_2+\alpha_3) - 1/5 = 9/5\ (\alpha_1+\alpha_2+\alpha_3) - 3/5\ \alpha_1 - 1/5$. Since $\alpha_1 + \alpha_2 + \alpha_3 \leq 1/2$ and $\alpha_1 > 0$, we have $f(\alpha_1) + f(\alpha_2) + f(\alpha_3) \leq 7/10$.

Let us define the coarseness of a bin to be the largest α such that some bin to its left is filled to level $1 - \alpha$. The coarseness of the left-most bin is 0.

Lemma 3: If bins are filled by first fit, and some bin has coarseness α, then every number in it exceeds α.

Proof: If not, a violation of the first fit algorithm is immediate.

Lemma 4: Let a bin of coarseness $\alpha < 1/2$ be filled with numbers $\alpha_1, \alpha_2, \ldots, \alpha_n$. If $\sum_{i=1}^{n} \alpha_i \geq 1 - \alpha$, then $\sum_{i=1}^{n} f(\alpha_i) \geq 1$.

Proof: If $\alpha_i > 1/2$ for any i, then the result is immediate, since $f(\alpha_i) = 1$. We therefore assume that $\alpha_i \leq 1/2$ for all i. We consider several cases, depending on the range of α.

Case 1: $\alpha \le 1/6$. Then $\sum_{i=1}^{n} \alpha_i \ge 1 - \alpha \ge 5/6$. Since $f(\beta)/\beta \ge 6/5$ in the range $0 \le \beta \le 1/2$, we immediately have $\sum_{i=1}^{n} f(\alpha_i) \ge 1$.

Case 2: $1/6 < \alpha \le 1/3$. We consider subcases, depending on the value of n.

n = 1: Then since $\alpha_1 \le 1/2$, we must have $\alpha \ge 1/2$. But we assume $\alpha \le 1/3$.

n = 2: If both α_1 and α_2 are equal to or greater than 1/3, then $f(\alpha_1) + f(\alpha_2) \ge 1/2 + 1/2 \ge 1$. If both are less than 1/3, then $\alpha_1 + \alpha_2 < 2/3 \le 1 - \alpha$, which is impossible. Therefore, we may assume that $\alpha \le \alpha_1 < 1/3 \le \alpha_2 \le 1/2$. Then $f(\alpha_1) + f(\alpha_2) = 9/5\ \alpha_1 + 6/5\ \alpha_2 = 3/5\ \alpha_1 + 6/5\ (\alpha_1+\alpha_2)$. We know that $\alpha_1 + \alpha_2 \ge 1 - \alpha$, and $\alpha_1 \ge \alpha$, so $f(\alpha_1) + f(\alpha_2) \ge 3/5\ \alpha + 6/5\ (1-\alpha) = 6/5 - 3/5\ \alpha$. Since we assume $\alpha \le 1/3$, we have $f(\alpha_1) + f(\alpha_2) \ge 1$.

n = 3: As in the previous case, if two of α_1, α_2 and α_3 exceed 1/3, the result is immediate. If exactly one, say α_3, does so, then $f(\alpha_1) + f(\alpha_2) + f(\alpha_3) = 9/5\ (\alpha_1+\alpha_2) + 6/5\ (\alpha_3) - 1/10 = 6/5\ (\alpha_1+\alpha_2+\alpha_3) + 3/5\ (\alpha_1+\alpha_2) - 1/10$. We have $\alpha_1 + \alpha_2 + \alpha_3 \ge 1 - \alpha$ and $\alpha_1 + \alpha_2 \ge 2\alpha$. Therefore, $f(\alpha_1) + f(\alpha_2) + f(\alpha_3) \ge 11/10$.

If, on the other hand, none exceed 1/3, then $f(\alpha_1) + f(\alpha_2) + f(\alpha_3) = 9/5\ (\alpha_1+\alpha_2+\alpha_3) - 3/10 \ge 3/2 - 9/5\ \alpha$. If $\alpha \le 5/18$, then $3/2 - 9/5\ \alpha \ge 1$, so we have our desired result. If $5/18 < \alpha \le 1/3$, then $\alpha_1 + \alpha_2 + \alpha_3 \ge 5/6$, so $f(\alpha_1) + f(\alpha_2) + f(\alpha_3) \ge 9/5\ (5/6) - 3/10 = 6/5$.

n = 4: We may again restrict ourselves to the cases where none of $\alpha_1, \ldots, \alpha_4$ are greater than 1/3 or one of these, say α_4, is. In the latter case, $\sum_{i=1}^{4} f(\alpha_i) = 9/5\ (\alpha_1+\alpha_2+\alpha_3) + 6/5\ \alpha_4 - 1/5 \ge 1 - 6/5\ \alpha + 3/5\ (\alpha_1+\alpha_2+\alpha_3) \ge 1 + 3/5\ \alpha \ge 1$. If all are less than 1/3, then $\sum_{i=1}^{4} f(\alpha_i) = 9/5\ (\alpha_1+\alpha_2+\alpha_3+\alpha_4) - 2/5 \ge 7/5 - 9/5\ \alpha$. If $\alpha \le 2/9$, we have our desired result. If $2/9 < \alpha \le 1/3$, then $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 \ge 8/9$, so $\sum_{i=1}^{4} f(\alpha_i) \ge 9/5\ (8/9) - 2/5 = 6/5$.

n ≥ 5: This case is immediate, as $f(\alpha_i) \ge 1/5$ for $\alpha_i \ge 1/6$.

Case 3: $1/3 < \alpha < 1/2$. Then $n \le 2$. If n = 2, then since α_1 and α_2 are each greater than 1/3, the result is immediate. If n = 1, we have $\alpha_1 \ge 1 - \alpha > 1/2$, so $f(\alpha_1) = 1$.

Corollary: If a bin of coarseness $\alpha < 1/2$ is filled with $\alpha_1, \ldots, \alpha_n$, and $\sum_{i=1}^{n} f(\alpha_i) = 1 - \beta$, where $\beta > 0$, then either:

(1) n = 1 and $\alpha_1 \le \frac{1}{2}$, or

(2) $\sum_{i=1}^{n} \alpha_i \le 1 - \alpha - \frac{5}{9}\beta$.

Proof: If n = 1 and $\alpha_1 > 1/2$, it is impossible that $\beta > 0$. Therefore, if (1) does not hold, we may assume that $n \ge 2$. Let $\sum_{i=1}^{n} \alpha_i = 1 - \alpha - \gamma$. Then we may construct a bin filled with $\alpha_3, \alpha_4, \ldots, \alpha_n$ and two other numbers δ_1 and δ_2, selected so that $\delta_1 + \delta_2 = \alpha_1 + \alpha_2 + \gamma$, $\delta_1 \ge \alpha_1$, $\delta_2 \ge \alpha_2$, and neither δ_1 nor δ_2 exceeds 1/2. By Lemma 4, $\sum_{i=3}^{n} f(\alpha_i) + f(\delta_1) + f(\delta_2) \ge 1$. But since the slope of f in the range [0,1/2] does not exceed 9/5, it follows that $f(\delta_1) + f(\delta_2) \le f(\alpha_1) + f(\alpha_2) + 9/5\ \gamma$. Therefore, $\gamma \ge 5/9\ \beta$, and (2) holds.

Theorem 2: For all $\varepsilon > 0$ there exists N such that if $k \ge N$, then $R^{FF}(k) \le 17/10 + \varepsilon$.

Proof: Let $A = (\alpha_1, \ldots, \alpha_r)$, and let $W = \sum_{i=1}^{r} f(\alpha_i)$. By Lemma 2, $A^* \ge 10/17\ W$. Suppose that in the first fit algorithm, bins $B_1, B_2, \ldots, B_m$ are all the bins that receive at least one number, but for which $\sum_i f(\alpha_i)$, where i ranges over all α_i in the bin, sums to $1 - \beta_i$, for $\beta_i > 0$. Let γ_i be the coarseness of B_i. If $\gamma_i \ge 1/2$, then by Lemma 1, $B_{i+1}, \ldots, B_m$ are each more than half full, and must hold one number greater than 1/2.

If $i < m$ and $\gamma_i \ge 1/2$, then $B_m > 0$ is impossible, since B_k must hold one number α, for which $f(\alpha) = 1$. Thus, i = m. We may conclude that $\gamma_i < 1/2$ for $1 \le i < m$. By Lemma 3 and the corollary to Lemma 4, we have $\gamma_i \ge \gamma_{i-1} + 5/9\ \beta_{i-1}$ for all $1 < i < m$. Thus $\sum_{i=1}^{m-2} \beta_i \le 9/5\ (\gamma_{m-1} - \gamma_1) < 1$.

Since β_{m-1} and β_m cannot exceed 1 (an argument which can be improved greatly), we have $\sum_{i=1}^{m} \beta_i \leq 3$.

We therefore find that by Lemma 4, $A^{FF} \leq W + 3$. Hence $A^{FF}/A^* \leq 17/10 + 51/10W$. Since $W \geq A^{FF} - 3$, it must be that $W \geq A^* - 3$. Given $\varepsilon > 0$, choose $N \geq 51/10\varepsilon + 3$. Then $A^{FF}/A^* \leq 17/10 + \varepsilon$.

As a consequence of Theorems 1 and 2, we know that $\lim_{k\to\infty} R^{FF}(k) = 17/10$.

It is interesting to note that for several values of k the ratio 17/10 can actually be attained. In particular, there is a list A with $A^* = 10$ and $A^{FF} = 17$. The packing of bins, with all quantities in units of 1/101 is shown in Fig. 2. There is also a list A with $A^* = 20$ and $A^{FF} = 34$. Perhaps $R^{FF}(k) < 17/10$ for $k \geq 30$.

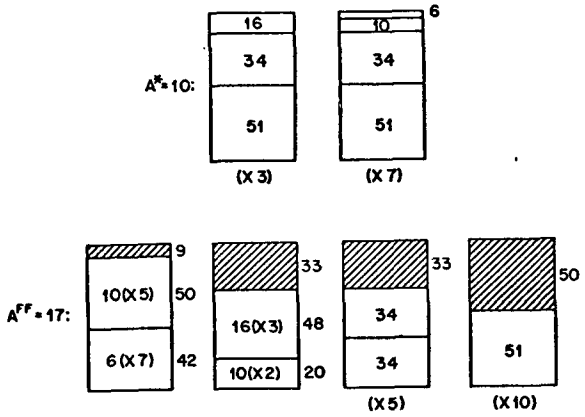

Fig. 2. An example with $A^{FF} = 17$ and $A^* = 10$.

In order for the ratios $R^{FF}(k)$ and $R^{BF}(k)$ to achieve relatively large values, it is necessary for some of the α_i to be relatively large. In fact, when all $\alpha_i \leq \alpha \leq 1/2$, we have the following result, stated without proof.

Theorem 3: For any $\alpha \leq 1/2$ and any $\varepsilon > 0$, there exists N such that for any list $A = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ with $\max \alpha_i \leq \alpha$ and $A^* > N$, we have

$$A^{FF}/A^* - \varepsilon \leq 1 + \lfloor \alpha^{-1} \rfloor^{-1}$$

and

$$A^{BF}/A^* - \varepsilon \leq 1 + \lfloor \alpha^{-1} \rfloor^{-1}.\dagger$$

†Where $\lfloor x \rfloor$ denotes the greatest integer $\leq x$. Note that $\lfloor \alpha^{-1} \rfloor^{-1}$ is piecewise continuous and flat in the ranges $1/i < \alpha \leq 1/(i-1)$, for integers $i \geq 3$.

Furthermore, this bound cannot be replaced by any smaller function of α.

III. FIRST FIT DECREASING AND BEST FIT DECREASING

The exact asymptotic values of $R^{FFD}(k)$ and $R^{BFD}(k)$ are not yet known. However, they each can be bounded below by 11/9 and above by 5/4. We begin with an extremely useful lemma.

Lemma 1: Given any list A, let α be the last number placed in a previously empty bin when applying the FFD method to A. Then

$$A^{FFD}/A^* < 1/(1-\alpha) + 1/A^*.$$

The same result holds with FFD replaced by BFD.

Proof: Since α could not be placed in any of the first $A^{FFD} - 1$ bins, each of them must be filled to a level greater than $1 - \alpha$. But since each of the A^* bins in the optimal placement is filled to level at most 1,

$$(A^{FFD}-1)\cdot(1-\alpha) + \alpha < A^*,$$

or, rewriting,

$$A^{FFD} < (A^*+1-2\alpha)/(1-\alpha).$$

We then have

$$A^{FFD}/A^* < 1/(1-\alpha) + (1-2\alpha)/(A^*(1-\alpha))$$
$$< 1/(1-\alpha) + 1/A^*.$$

We now use Lemma 1 to obtain an upper bound for $R^{BFD}(k)$.

Theorem 4: For any $\varepsilon > 0$, there exists an N such that, for all $k > N$,

$$R^{BFD}(k) < 5/4 + \varepsilon.$$

Proof: We merely sketch the lengthy proof. Choose $N = 2/\varepsilon$, suppose we have a list A which exceeds the bound. By Lemma 1, we can assume that every number on A is larger than 1/5, since all numbers smaller than the α of Lemma 1 can be eliminated without reducing A^{BFD}/A^*.

Consider any optimal placement for A. We apply the BFD method to successively relocate the numbers, eventually transforming the optimal placement into the BFD placement, in such a way that the number of extra bins required can be bounded.

The BFD relocation proceeds in n stages, corresponding to the BFD placement of $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_n$. Initially, all the α_i are "unfixed", meaning that each can still be moved to a different bin. During stage k, the number α_k is "fixed" by placing it permanently into the bin B_j into which it fits best, ignoring all unfixed weights. If α_k was not in B_j at the beginning of stage k, then the largest unfixed α_i from B_j is placed in

the location just vacated by α_k (α_i must fit in that space since the fact that α_i has not yet been fixed implies $\alpha_i \leq \alpha_k$). At this point, the total amount contained in B_j may exceed 1, since it may still contain some unfixed numbers. If this occurs, some of the unfixed α_i are relocated so that no bin contains a total more than 1. The detailed description of this relocation process (entailing some 20 cases) will not be included here. During this relocation, it may be necessary to remove one unfixed α_i to a special list of displaced numbers, rather than simply moving it from one bin to another. However, whenever a number is displaced in this way, certain rather complicated properties must be satisfied by both the displaced number and the bin from which it is displaced. These properties are designed to guarantee that no displaced numbers will be larger than 1/3 and that the list of displaced numbers never contains more than $A^* - \ell$ numbers, where ℓ of the displaced numbers are larger than 1/4. Furthermore, these properties ensure than once bin $A^* + 1$ is started, no more α_i will be displaced. Since the number of additional bins required by the BFD method is essentially determined by the number and sizes of the α_i on the displaced list during the stage that bin $A^* + 1$ is started, and since both have been suitably restricted by the relocation technique, one can show that the list A could not have exceeded the bound of the theorem, a contradication.

The same upper bound holds for R^{FFD}.

Theorem 5: For any $\varepsilon > 0$, there exists an N such that, for all $k > N$,

$$R^{FFD}(k) < 5/4 + \varepsilon.$$

Proof: The proof follows from Lemma 1, Theorem 4, and the following, somewhat surprising, result which shows that FFD and BFD require the same number of bins whenever the list A contains no element smaller than 1/5.

Theorem 6: For any list $A = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, if $\min \alpha_i \geq 1/5$, then $A^{FFD} = A^{BFD}$.

Proof: Let $A = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ be a list with $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_n \geq 1/5$. We are going to imagine assigning two copies of A. One copy will be assigned to the bins B_i using the first fit decreasing (FFD) algorithm. The other copy will be assigned to the bins B_i' using the best fit decreasing (BFD) algorithm. We shall make these two assignments simultaneously, maintaining certain relationships between the contents of various B_i and B_i' as we proceed.

More precisely, at each stage of the assignment, the k^{th} stage starting with the assignment of α_k, we shall associate to each B_i a unique bin B_j' which will be denoted by $B_i'(k)$. Although $B_i'(k)$ may vary as k increases, this will cause no difficulties since the ordering of the bins is irrelevant when applying the BFD algorithm. For notational convenience we shall let $B_i(k) = B_i$ for all k. Also, $|B_i(k)|$ will denote the sum of the α_j which have been assigned to B_i after the elements $\alpha_1, \alpha_2, \ldots, \alpha_k$ have all been assigned, with $|B_i'(k)|$ defined similarly. Finally, let the current unused capacity $1 - |B_i(k)|$ in $B_i(k)$ be denoted by $g_i(k)$, with $g_i'(k)$ defined similarly. If $g_i(k) < 1/5$, we shall say that $B_i(k)$ is closed, since no further α_j can be assigned to $B_i(k)$. If $g_i(k) < 2/5$, we shall say that $B_i(k)$ is nearly closed. Of course, the same terminology applies to $g_i'(k)$.

Initially, before α_i is assigned, to each B_i is assigned the box $B_i'(0) = B_i'$. Since at this stage, all the bins are empty, this assignment is arbitrary. For $k = 0, 1, \ldots, n$, consider the following statement (which applies for all $i \geq 1$).

S(k): (i) If $g_i(k) > g_i'(k)$ then $g_i(k) < 1/5$.

(ii) If $g_i(k) < g_i'(k)$ then $g_i'(k) < 2/5$.

(iii) If $g_i(k) < g_i'(k)$ and $g_i'(k) \geq 1/5$ then $\alpha_{k+1} \leq g_i(k)$.

(iv) $g_i(k) = 1$ if and only if $g_i'(k) = 1$.

S(0) is certainly true since $g_i(0) = g_i'(0) = 1$ for all i. For a fixed $k < r$, assume S(k) has been established. We shall show that S(k+1) also holds. Once this is done then by induction, condition (iv) of S(n) implies the desired result.

Thus, we begin the $(k+1)^{st}$ stage by assigning α_{k+1} to one of the $B_i(k)$, as well as to one of the $B_j'(k)$. Suppose in the FFD algorithm, α_{k+1} is assigned to $B_i(k)$ while in the BFD algorithm, α_{k+1} is assigned to $B_j'(k)$. There are several possibilities.

(I) $i = j$: In this case conditions (i), (ii) and (iv) are immediate for S(k+1). To see that (iii) holds for S(k+1), it is sufficient to note that if $g_i(k+1) < g_i'(k+1)$ then we must have $g_i(k) < g_i'(k)$. By (ii) for S(k), $g_i'(k) < 2/5$. Since $\alpha_{k+1} \geq 1/5$ then $g_i'(k+1) = g_i'(k) - \alpha_{k+1} < 1/5$ so that (iii) in S(k+1) holds vacuously for index i and by induction for the other indices.

(II) $i > j$: Since α_{k+1} is assigned to $B_j'(k)$ then $1/5 \leq \alpha_{k+1} \leq g_j'(k)$. Thus, by (i) we have $g_j(k) \leq g_j'(k)$. If $g_j(k) = g_j'(k)$ then $\alpha_{k+1} \leq g_j'(k) = g_j(k)$ and α_{k+1} could have been assigned to $B_j(k)$

by the FFD algorithm which is a contradiction. Hence, we may assume $g_j(k) < g'_j(k)$. By (iii), $\alpha_{k+1} \le g_j(k)$ which again implies the same contradiction above.

(III) $\underline{i < j}$: Define $B'_i(k+1) = B_j(k)$, $B'_j(k+1) = B_i(k)$ and $B'_m(k+1) = B_m(k)$ for $m \neq i,j$. Thus,

$$g_i(k+1) = g_i(k) - \alpha_{k+1}, \quad g'_i(k+1) = g'_j(k) - \alpha_{k+1},$$

$$g_j(k+1) = g_j(k), \quad g'_j(k+1) = g_i(k).$$

We must verify the conditions for S(k+1). These are, for all i,

(i') If $g_i(k+1) > g'_i(k+1)$ then $g_i(k+1) < 1/5$.

(ii') If $g_i(k+1) < g'_i(k+1)$ then $g'_i(k) < 2/5$.

(iii) If $g_i(k+1) < g'_i(k+1)$ and $g'_i(k) \ge 1/5$ then $\alpha_{k+2} \le g_i(k)$.

(iv) $g_i(k) < 1$ if and only if $g'_i(k) < 1$.

Since only the bins $B_i(k)$, $B'_i(k)$, $B_j(k)$, $B'_j(k)$ have been affected by the transformations of the $(k+1)^{st}$ stage then it suffices to verify the following conditions:

(a) $g_i(k) - \alpha_{k+1} < 1/5$; $g'_j(k) - \alpha_{k+1} < 1/5$.

(b) If $g_j(k) < g'_i(k)$ then $g'_i(k) < 2/5$.

(c) If $g_j(k) < g'_i(k)$ and $g'_i(k) \ge 1/5$ then $\alpha_{k+1} \le g_j(k)$.

(d) $g'_i(k) < 1$ if and only if $g_j(k) < 1$.

Since $\alpha_{k+1} \le g_i(k)$ then by (i) we have

$$\alpha_{k+1} \le g_i(k) \le g'_i(k).$$

Thus, $\alpha_{k+1} \le g'_i(k)$ so that by the definition of the BFD algorithm, $g'_j(k) \le g'_i(k)$. But if $g'_j(k) = g'_i(k)$ then we could just as well have assigned α_{k+1} to $B'_i(k)$ with no change in the set of values of the $g_i(k+1)$. Hence we may assume

$$g'_j(k) < g'_i(k).$$

(a) If $g_i(k) \neq g'_i(k)$ then by (i) and (ii), $g'_i(k) < 2/5$. Thus, $g'_i(k) - \alpha_{k+1} < 1/5$ and by above

$$g_i(k) - \alpha_{k+1} < 1/5, \quad g'_j(k) - \alpha_{k+1} < 1/5.$$

If $g_i(k) = g'_i(k)$ then $g_i(k) > g'_j(k)$. But $g'_j(k) \ge g_j(k)$ by (i) so that $g_i(k) > g_j(k)$. Thus, by the definition of the FFD algorithm, at least two α's must be assigned to $B_j(k)$. But by (iii), $\alpha_{k+1} \le g_j(k)$ so that $g_j(k) \ge 1/5$. Hence, at least one α_t assigned to $B_j(k)$ must satisfy $\alpha_t \le 1/2\,(1-g_j(k)) \le 2/5$. Thus, by the definition of the FFD algorithm, $g_i(k) < \alpha_t \le 2/5$. This implies

$$g_i(k) - \alpha_{k+1} < 1/5 \text{ and thus, } g'_j(k) - \alpha_{k+1} < 1/5.$$

(b) Suppose $g_j(k) < g'_i(k)$. If $g'_i(k) \ge 2/5$ then by (ii), $g_i(k) = g'_i(k)$. But in this case the argument in (a) applies to show that $g'_i(k) < 2/5$ which is a contradiction. Hence, we must have $g'_i(k) < 2/5$.

(c) If $g_j(k) = g'_j(k)$ then certainly $\alpha_{k+1} \le g_j(k)$ since $\alpha_{k+1} \le g'_j(k)$. If $g_j(k) \neq g'_j(k)$ then by (i), $g_j(k) < g_j(k)$. By (iii), $\alpha_{k+1} \le g_j(k)$.

(d) If $g_j(k) = 1$ then $g_j(k) = 1$ by (iv). If $g'_i(k) < 1$ then by the rules for the BFD algorithm α_{k+1} should have been assigned to $B'_i(k)$ before being put into $B'_j(k)$. Since it was not then $g'_i(k) = 1$. If $g'_i(k) = 1$ then $g_i(k) = 1$ by (iv). Thus, by the rules for the FFD algorithm, $g_j(k) = 1$.

These conditions establish S(k+1). By induction, S(n) holds. Condition (iv) of S(n) implies the desired result.

It is interesting to note [5] that extensive experimentation with randomly generated lists including elements of all sizes has failed to uncover a single instance in which $A^{FFD} \neq A^{BFD}$. However, such examples do exist. For example, Fig. 3 shows how to construct arbitrarily large examples with $A^{FFD}/A^{BFD} = 11/10$. Note that the smallest pieces in that example are of size $1/5 - \varepsilon$, so the constant 1/5 in Theorem 6 cannot be improved upon.

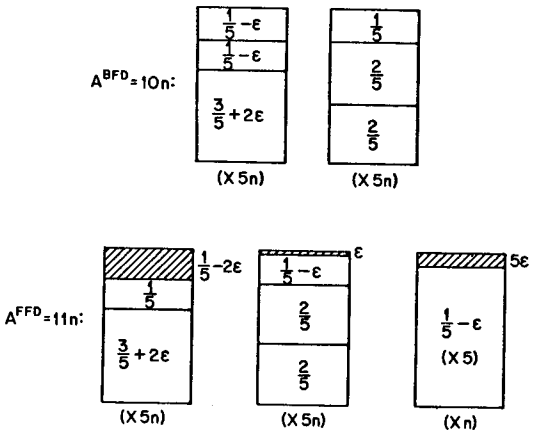

Fig. 3 An example with $A^{FFD}/A^{BFD} = 11/10$ and A* = 10n.

It is also possible that A^{BFD} can be substantially greater than A^{FFD}. Figure 4 shows that A^{BFD}/A^{FFD} can be as large as 10/9.

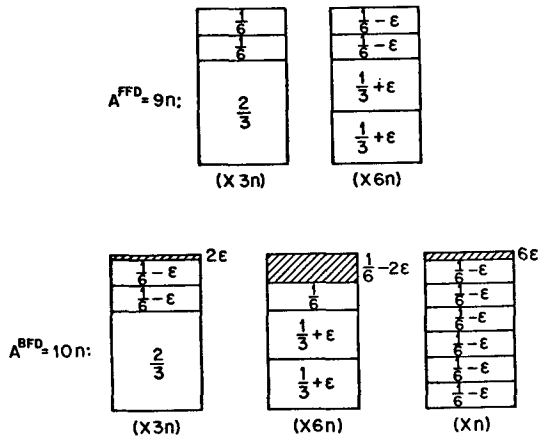

Fig. 4 An example with $A^{BFD}/A^{FFD} = 10/9$ and $A^* = 9n$.

We now turn to lower bounds on R^{FFD} and R^{BFD}.

Theorem 7:

$$\lim_{k\to\infty} R^{FFD}(k) \geq 11/9, \quad \lim_{k\to\infty} R^{BFD} \geq 11/9.$$

Proof: Figure 5 exhibits the method for obtaining arbitrarily large examples having

$$A^{FFD}/A^* = A^{BFD}/A^* = 11/9.$$

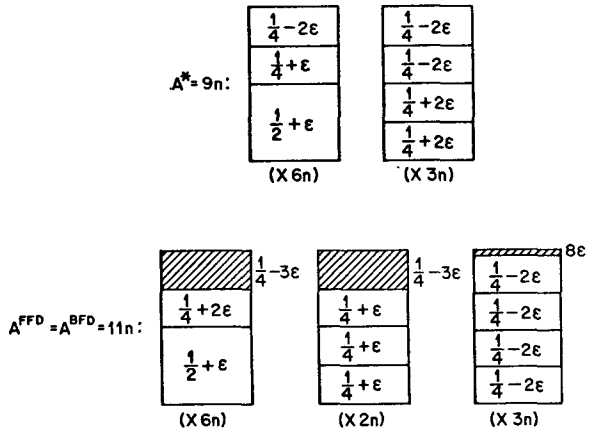

Fig. 5 An example with $R^{FFD}(k) = R^{BFD}(k) = 11/9$ for $k = 9n$.

IV. CONCLUSIONS AND OPEN PROBLEMS

We have considered four heuristic algorithms for the problem of bin packing. Of these, only for first fit do we know the asymptotic worst-case behavior exactly (17/10). For best fit, we know the ratio to be at least 17/10, and for first fit decreasing and best fit decreasing the ratio is bounded between 11/9 and 5/4. Some additional results on these and other algorithms can be found in [6,7].

The following open problems suggest themselves.

(1) Find the asymptotic values of R^{BF}, R^{BFD}, and R^{FFD}. It is conjectured that

$$\lim_{k\to\infty} R^{FFD}(k) = \lim_{k\to\infty} R^{BFD}(k) = 11/9.$$

(2) Is it true that $9/10 \leq A^{FFD}/A^{BFD} \leq 11/10$, for all lists A? If not, find the correct bounds on A^{FFD}/A^{BFD}. Also, it is natural to ask for the bounds on A^{FF}/A^{BF}.

(3) Lemma 1 provides some information on the performance of FFD and BFD when element sizes are restricted to being no greater than α, however, not nearly as much information as is given for FF and BF in Theorem 3. What more can be said about FFD and BFD applied to such restricted lists?

Additional open problems are described in [6].

References

1. P. C. Gilmore and R. E. Gomory, A linear programming approach to the cutting stock problem II, Oper. Res. 11 (1963), pp. 863-888.

2. R. W. Conway, W. L. Maxwell and L. W. Miller, Theory of Scheduling, Addison-Wesley, Reading, Mass. (1967).

3. S. Eilon and N. Christofides, The loading problem, Manag. Sci. 17, no. 5 (1971), pp. 259-268.

4. R. C. Prim, private communication.

5. J. Curry, private communication.

6. R. L. Graham, Bounds on multiprocessing anomalies and related packing algorithms, to appear in Proc. of 1972 SJCC.

7. A. Demers and J. D. Ullman, (in preparation).