NEW ALGORITHMS IN BIN PACKING

by

Andrew Chi-Chih Yao

STAN-CS-78-662
SEPTEMBER 1978

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY

New Algorithms in Bin Packing

Andrew Chi-Chih Yao

Computer Science Department
Stanford University
Stanford, California 94305

Abstract.

In the bin-packing -problem a list L of n numbers are to be packed into unit-capacity bins. For any algorithm S , let r(S) be the maximum ratio $S(L)/L^*$ for large L^* , where S(L) denotes the number of bins used by S and L^* denotes the minimum number needed. In this paper we give an on-line O(n log n) -time algorithm RFF with r(RFF) = 5/3 , and an off-line -polynomial-time algorithm RFFD with r(RFFD) = $(11/9)-\epsilon$ for some fixed $\epsilon > 0$. These are strictly better respectively than two prominent algorithms -- the First-Fit (FF) which is on-line with r(FF) = 17/10 , and the First-Fit-Decreasing (FFD) with r(FFD) = 11/9 . Furthermore, it is shown that any on-line algorithm S must have $r(S) \geq 3/2$. We also discuss the question "how well can an O(n) -time algorithm perform?", showing that, in the generalized d-dimensional bin-packing, any O(n) -time algorithm S must have $r(S) \geq d$.

Keywords: bin-packing, First-Fit, First-Fit-Decreasing, heuristic algorithm, NP-complete, on-line.

This research was supported in -part by National Science Foundation grants MCS-72-03752 A03 and MCS-77-05313.

1. Introduction.

Let $L = (x_1, x_2, \ldots, x_n)$ be a given list of real numbers in $(0,1]$, and $BIN_1, BIN_2, \ldots$, an infinite sequence of bins each of unit capacity. The bin packing problem is to assign each x_i into a unique bin, with the sum of numbers in each bin not exceeding 1, such that the total number of used bins is a minimum (denoted by L^*). As this problem is NP-complete [8], efficient algorithms that always generate packings using L^* bins are unlikely to exist. In the literature, heuristic algorithms with guaranteed bounds on performance have been studied extensively [5],[6],[7]. For any (heuristic) bin packing algorithm S, let S(L) denote the number of bins used for the input list L, and $R_S(k)$ the maximum ratio $S(L)/L^*$ for any list L with $L^* = k$. The performance ratio of S, denoted by $r(S)$, is defined as $\overline{\lim_{k \to \infty}} R_S(k)$. Informally, $(r(S)-1) \times 100\%$ is the percentage of excess bins used over the optimal packing in the worst case, for large lists. Two prominent algorithms are the First-Fit Algorithm (FF) and the First-Fit-Decreasing Algorithm (FFD) (see Section 2 for definitions). It is known [7] that $r(FF) = 17/10$ and $r(FFD) = 11/9$.

A natural question is, how good can any polynomial algorithm be? In this regard, two specific questions were raised by Johnson [6]:

Is there a polynomial on-line algorithm S better than First-Fit (i.e., with $r(S) < 17/10$)?

Is there any polynomial algorithm S better than First-Fit-Decreasing (i.e., with $r(S) < 11/9$)?

We call an algorithm on-line if the numbers in list L are available one at a time, and the algorithm has to assign each number before the next one becomes available [5],[6]. In this paper, we resolve both questions in the affirmative. It will also be shown that no on-line algorithm can have a performance ratio less than $3/2$.

Section 3 gives an $O(n \log n)$ -time on-line algorithm S with $r(S) = 5/3$. Section 4 explores the limitation to on-line algorithms, showing that no such algorithm S (polynomial-time or not) can have $r(S) < 3/2$. In Section 5, a general approach for seeking improvements over known heuristic algorithms is suggested and illustrated with an example. Based on this idea, a heuristic polynomial-time algorithm better than FFD is constructed in Section 6. We discuss in Section 7 the question "How well can an $O(n)$ -time algorithm perform?". It is shown that in a generalized version of bin packing, namely the d-dimensional bin packing [2], any $O(n)$ -time algorithm S must have $r(S) \geq d$.

2. Terminologies.

For standard definitions with regard to the bin packing problem, the reader is referred to [7]. We will mention below only a few terminologies for use in the present paper.

A list is a finite sequence of real numbers. Some numbers may have identical values, but are regarded as distinct item;. A set of real numbers in this paper is often in fact a multiset, in which some numbers may appear more than once (see [9]).

If $L_1 = (x_1, x_2, \ldots, x_n)$ and $L_2 = (y_1, y_2, \ldots, y_\ell)$ are two lists, their concatenation $L_1 L_2$ is the list $L = (x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_\ell)$. Let X be a bin used in a packing, the content of X, $\mathrm{cont}(X)$, is the sum of the numbers that are assigned to X. We shall say that a bin packing algorithm S has running time $O(p(n))$ if, when implemented on a random access machine [1], S takes at most $O(p(n))$ steps to produce the packing for a list with n numbers. We describe the two algorithms FF and FFD for easy reference:

First-Fit (FF). Given a list $L = (x_1, x_2, \ldots, x_n)$, the algorithm assigns x_j sequentially, for $j = 1, 2, \ldots, n$, to BIN_i with the smallest i whose current content does not exceed $1 - x_j$.

First-Fit-Decreasing (FFD). Given a list $L = (x_1, x_2, \ldots, x_n)$, the algorithm first sorts the x_j's into decreasing order, and then performs First-Fit.

Both FF and FFD can be implemented to have a running time $O(n \log n)$; for details, see [6].

3. A New On-line Algorithm.

We will present an on-line algorithm that processes a list of n numbers in $O(n \log n)$ time, and show that its performance ratio is $5/3 = 1.666\cdots$.

Any element x_j in a list L will be called an A-piece, B_1-piece, B_2-piece, or X-piece if x_j is in the interval $(1/2, 1]$, $(2/5, 1/2]$, $(1/3, 2/5]$, or $(0, 1/3]$, respectively.

Algorithm RFF (Refined First Fit).

Before packing, we divide the set of all bins into four infinite classes. The algorithm then proceeds as follows. Let $m \in \{6,7,8,9\}$ be a fixed integer. Suppose the first $j-1$ numbers in list L have been assigned, we process the next number x_j according to the following rules.

(a) We put x_j by first-fit into a bin in:

- class 1, if x_j is an A-piece,
- class 2, if x_j is a B_1-piece,
- class 3, if x_j is a B_2-piece, but not the (mi)-th B_2-piece seen so far for any integer $i \geq 1$,
- class 4, if x_j is an X-piece,

(b) If x_j is the (mi)-th B_2-piece seen so far for some integer $i \geq 1$, we put x_j into the first fitting bin containing an A-piece in class 1 if -possible, and put x_j in a new bin of class 1 otherwise.

Analysis of RFF. This algorithm can be implemented to run in $O(n \log n)$ tine, as it essentially performs a first-fit within each class of bins, which takes $O(\log n)$ time for each x_j (see [6]).

We shall now analyze the performance ratio of RFF. In general the resulting packing of a list L has the following structure (Figure 1). There are three types of bins in class 1. Let Z_{11} be the set of class 1-bins containing a single A-piece, Z_{12} the set of class 1-bins containing a single B_2-piece, and Z_{13} the set of class 1-bins containing both an A-piece and a B_2-piece. In class 2, every non-empty bins contain exactly 2 B_1-pieces, except possibly for the last one. Let Z_2 denote the set of all (non-empty) class 2-bins, Let Z_3 be the set of class T-bins, each clearly containing 2 B_2-pieces, except possibly for the last one. The set of class 4-bins, denoted by Z_4 , is simply the FF-packing of the sublist of L consisting of the X-pieces. We shall write $|Z_{11}|, |Z_{12}|, |Z_{13}|, |Z_2|, \ldots$ as $z_{11}, z_{12}, z_{13}, z_2, \ldots$, etc. The numbers of A-pieces, B_1-pieces, B_2-pieces, X-pieces are denoted by a , b_1 , b_2 , and x , respectively.

. We shall first prove an upper bound on $r(\mathrm{RFF})$.

<u>Lemma 1</u>. For any list L , $\mathrm{RFF}(L) \le \frac{5}{3} L^* + 5$.

<u>Proof</u>. Clearly,

$$\mathrm{RFF}(L) = a + z_{12} + z_2 + z_3 + z_4 . \qquad (1)$$

<u>Fact 1</u>. Every bin BIN_i in Z_4 , with the possible exception of two bins, has $\mathrm{cont}(\mathrm{BIN}_i) \ge 3/4$.

<u>Proof</u>. The set of bins Z_4 can be regarded as the First-Fit packing of a list of pieces in $(0, 1/3]$. Therefore, every bin except the last one has at least 3 pieces. If BIN_j is the first bin with $\mathrm{cont}(\mathrm{BIN}_j) \le 3/4$, then all the bins following it contains only pieces

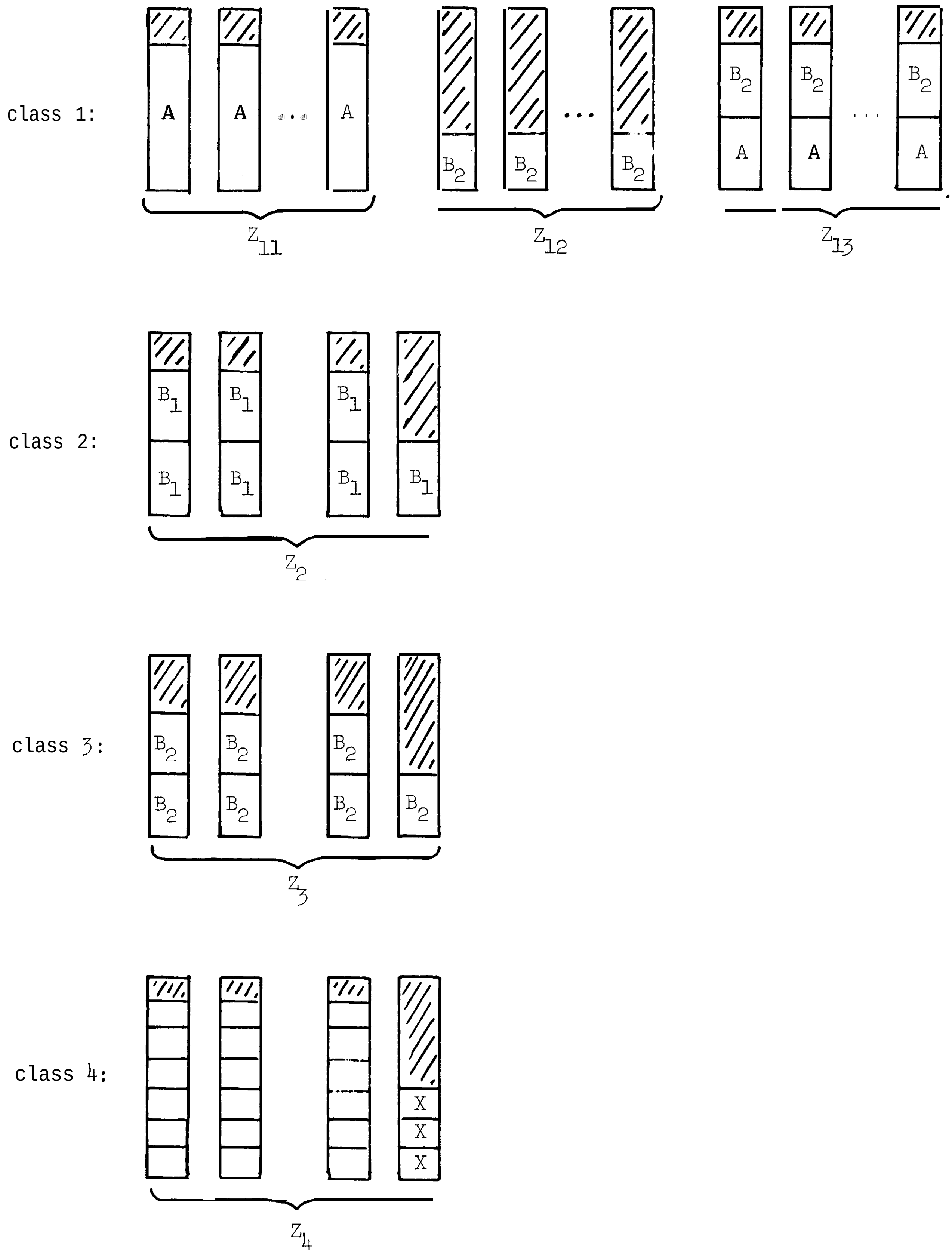

Figure 1. The structure of a packing using RFF. The ordering of bins and the relative positions of pieces within a bin are not necessarily represented faithfully.

greater than $1/4$. This means all bins following BIN_j , except the last one have contents exceeding $3/4$. □

Fact 1 has often been used in bin packing arguments (see [7, proof of Theorem 2.3]). Its proof is given here for convenience.

<u>Fact 2</u>. $z_{12} + z_{13} = \lfloor b_2/m \rfloor$, $z_2 = \lceil b_1/2 \rceil$, $\left| z_3 - \frac{1}{2}\left(1 - \frac{1}{m}\right) b_2 \right| \leq 2$.

<u>Proof</u>. The first two equations are obvious from the algorithm. The last one follows from $z_3 = \left\lceil \frac{1}{2}(b_2 - z_{12} - z_{13}) \right\rceil$. □

<u>Fact 3</u>. $a \leq L^*$.

<u>Proof</u>. No two A-pieces can be in the same bin in any packing. □

We shall find upper bounds on z_4 , and hence on $\mathrm{RFF}(L)$ via formula (1). There are several cases to consider.

<u>Case 1</u>. $z_{12} = 0$.

The total contents of class-4 bins is at most $L^* - \frac{1}{2}a - \frac{2}{5}b_1 - \frac{1}{3}b_2$. Thus, by Fact 1, we have

$$z_4 \leq 2 + \frac{4}{3}\left(L^* - \frac{1}{2}a - \frac{2}{5}b_1 - \frac{1}{3}b_2\right) . \tag{2}$$

Combining (1) and (2), one obtains

$$\mathrm{RFF}(L) \leq \frac{4}{3}L^* + \frac{1}{3}a + \left(z_2 - \frac{8}{15}b_1\right) + \left(z_3 - \frac{4}{9}b_2\right) + 2 . \tag{3}$$

Making use of Fact 2, Fact 3, and the fact $m \leq 9$, we have

$$z_2 - \frac{8}{15}b_1 \leq 1 ,$$

$$z_3 - \frac{4}{9}b_2 \leq 2 ,$$

$$a < L^* .$$

Formula (3) then implies

$$RFF(L) \leq \frac{5}{3} L^* + 5 .$$

<u>Case 2</u>. $z_{12} > 0$.

<u>Fact 4</u>. In this case, $\mathrm{cont}(BIN_i) + \mathrm{cont}(BIN_j) > 1$ for each $BIN_i \in Z_{11}$, $BIN_j \in Z_{12}$. In particular, $\mathrm{cont}(BIN_i) > 3/5$ for each $BIN_i \in Z_{11}$.

<u>Proof</u>. Otherwise, the A-piece in BIN_i should have shared the same bin with some B_2 -piece during the packing. □

<u>Case 2.1</u>. $z_{11} \geq z_{12}$.

The total sum of all A, B_i-pieces is at least

$$z_{12} + \frac{3}{5}(z_{11} - z_{12}) + \frac{5}{6} z_{13} + \frac{2}{5}(2z_2 - 1) + \frac{1}{3}(2z_3 - 1)$$

$$> \frac{3}{5} a + \frac{2}{5} z_{12} + \frac{7}{30} z_{13} + \frac{4}{5} z_2 + \frac{2}{3} z_3 - 1 ,$$

where we have used Fact 4 and the equation $z_{11} = a - z_{13}$. From Fact 1, we obtain

$$z_4 \leq 2 + \frac{4}{3}\left(L^* - \frac{3}{5} a - \frac{2}{5} z_{12} - \frac{7}{30} z_{13} - \frac{4}{5} z_2 - \frac{2}{3} z_3 + 1\right) . \qquad (4)$$

Combining (1) and (4), and noticing that $z_{13} \geq 0$ and $z_2 \geq 0$, we obtain

$$RFF(L) \leq \frac{4}{3} L^* + \frac{1}{5} a + \frac{7}{15} z_{12} + \frac{1}{9} z_3 + 4 . \qquad (5)$$

We now make use of Fact 2 to derive from (5)

$$RFF(L) \leq \frac{4}{3} L^* + \frac{1}{5} a + \left(\frac{1}{18} + \frac{37}{90} \frac{1}{m}\right) b_2 + 5 . \qquad (6)$$

Fact 5. $\frac{1}{5}a + \left(\frac{1}{18} + \frac{37}{90}\frac{1}{m}\right)b_2 \le \frac{1}{3}L^*$.

Proof. In an optimal packing of L , each bin with an A-piece can contain at most 1 B_2-piece, and any other bin at most 2 B_2-pieces. Thus $b_2 \le a + 2(L^* - a) = 2L^* - a$. Therefore

$$\frac{1}{5}a + \left(\frac{1}{18} + \frac{37}{90}\frac{1}{m}\right)b_2 \le \left(\frac{1}{18} + \frac{37}{90}\frac{1}{m}\right)2L^* + \left(\frac{1}{5} - \frac{1}{18} - \frac{37}{90}\frac{1}{m}\right)a .$$

As the second term on the R.H.S. is non-negative and $a \le L^*$, we have

$$\frac{1}{5}a + \left(\frac{1}{18} + \frac{37}{90}\frac{1}{m}\right)b_2 \le \left(\frac{23}{90} + \frac{37}{90}\frac{1}{m}\right)L^* \le \frac{1}{3}L^* ,$$

for $m \ge 37/7$. □

Formula (6) and Fact 5 lead to $RFF(L) \le \frac{5}{3}L^* + 5$.

Case 2.2. $z_{11} < z_{12}$.

Total sum of all A, B_i-pieces is at least

$$z_{11} + \frac{1}{3}(z_{12} - z_{11}) + \frac{5}{6}z_{13} + \frac{2}{5}(2z_2 - 1) + \frac{1}{3}(2z_3 - 1)$$

$$> \frac{2}{3}a + \frac{1}{3}z_{12} + \frac{1}{6}z_{13} + \frac{4}{5}z_2 + \frac{2}{3}z_3 - 1 .$$

By Fact 1,

$$z_4 \le 2 + \frac{4}{3}\left(L^* - \frac{2}{3}a - \frac{1}{3}z_{12} - \frac{1}{6}z_{13} - \frac{4}{5}z_2 - \frac{2}{3}z_3 + 1\right) .$$

It follows that

$$RFF(L) = a + z_{12} + z_2 + z_3 + z_4$$

$$< \frac{4}{3}L^* + \frac{1}{9}a + \frac{5}{9}z_{12} - \frac{2}{9}z_{13} - \frac{1}{15}z_2 + \frac{1}{9}z_3 + 4$$

$$\le \frac{4}{3}L^* + \frac{1}{9}a + \frac{5}{9}z_{12} + \frac{1}{9}z_3 + 4 .$$

Using Fact 2, we have

$$\begin{aligned} RFF(L) &\leq \frac{4}{3} L^* + \frac{1}{9} a + \frac{5}{9}\left(\frac{1}{m} b_2\right) + \frac{1}{9}\left(\frac{1}{2}\left(1 - \frac{1}{m}\right) b_2 + 2\right) + 4 \\ &\leq \frac{4}{3} L^* + \frac{1}{9} a + \left(\frac{1}{18} + \frac{1}{2}\,\frac{1}{m}\right) b_2 + 5 \quad . \qquad (7) \end{aligned}$$

Fact 6. $\frac{1}{9} a + \left(\frac{1}{18} + \frac{1}{2}\,\frac{1}{m}\right) b_2 \leq \frac{1}{3} L^*$.

Proof. $$\begin{aligned} \text{L.H.S.} &\leq \frac{1}{9} a + \left(\frac{1}{18} + \frac{1}{2}\,\frac{1}{m}\right)(2L^* - a) \\ &= \left(\frac{1}{9} + \frac{1}{m}\right) L^* + \left(\frac{1}{18} - \frac{1}{2m}\right) a \quad . \end{aligned}$$

The second term is never positive (as $m \leq 9$) , thus

$$\text{L.H.S.} \leq \left(\frac{1}{9} + \frac{1}{m}\right) L^* \leq \frac{1}{3} L^*$$

as $m \geq 6$. □

Formula (7) and Fact 6 lead to $RFF(L) \leq \frac{5}{3} L^* + 5$ for Case 2.2.

This completes the proof of Lemma 1. □

Lemma 1 implies that the performance ratio of RFF does not exceed 5/3 . We shall show that it is in fact exactly 5/3 .

Theorem 1. $r(RFF) = 5/3$.

Proof. We need only exhibit lists L with arbitrary large L^* such that $RFF(L) = \frac{5}{3} L^* + o(1)$.

Let $\delta_j = 4^{-(j+2)}$ for $j \geq 1$, and n an integer of the form $6k+1$ for some $k \geq 1$. Define $p_j = \frac{1}{2} + \delta_j$, $u_j = \frac{1}{4} + \delta_j$, $t_j = \frac{1}{4} - 2\delta_j$ for $1 \leq j \leq n$. Consider the list $L = L_1 L_2$, where

$$L_1 = (u_1, t_2, t_3, u_3, t_4, t_5, \ldots, u_{2j-1}, t_{2j}, t_{2j+1}, \ldots, u_{n-2}, t_{n-1}, t_n) \quad ,$$

and

$$L_2 = (u_2, u_4, \ldots \text{[illegible]}, p_1, p_2, \ldots \text{[illegible]} \ldots, t_1, u_n) \quad .$$

Clearly $L^* = n$ (Figure 2a). Now, using the easily verified fact that $(u_{2j\,1} + t_{2j} + t_{2j+1}) + \min\{t_k, u_i\} > 1$ for every $k > 2j+1$ and any i, the packing resulting from RFF is as shown in Figure 2b. Thus, $RFF(L) = \frac{5}{3} L^* + o(1)$. This -proves the lemma. □

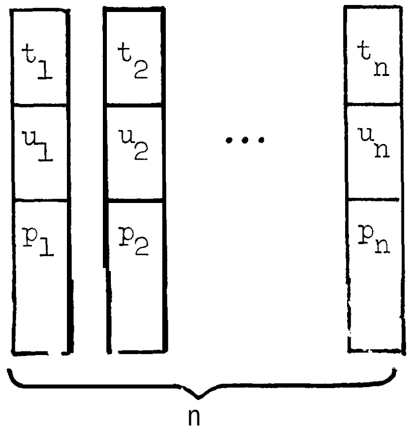

Figure 2a. An optimal packing of L in the proof of Theorem 1.

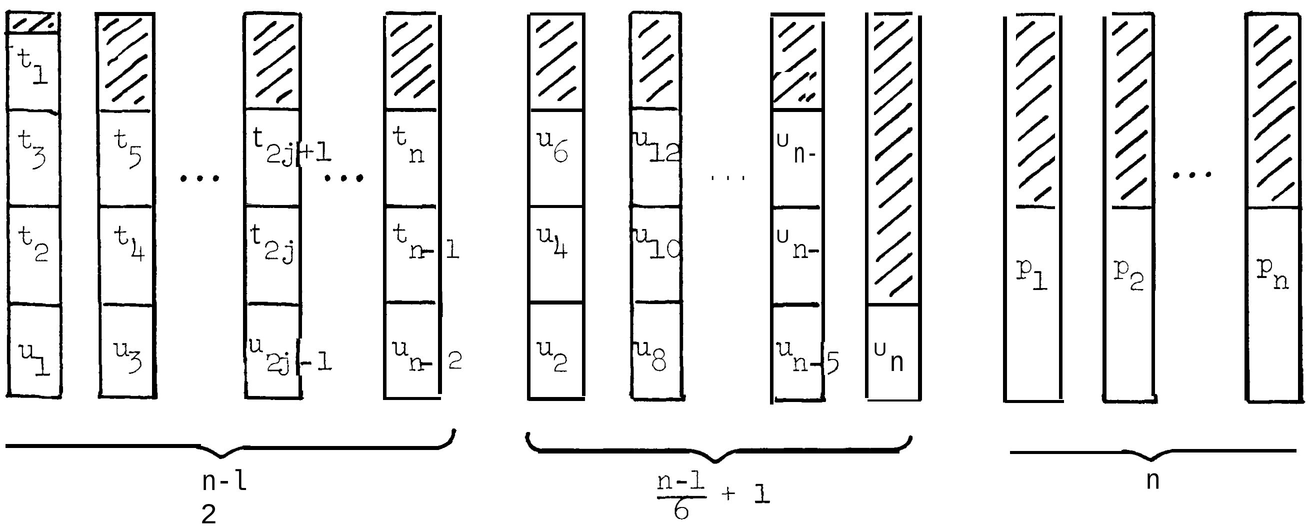

Figure 2b. The RFF packing of L.

4. A Lower Bound to r(S) for On-line Algorithms.

In this section we will show that one cannot expect to find on-line algorithms as good as, say, FFD , even if an arbitrary amount of computation is allowed.

Theorem 2. For any on-line bin packing algorithm S , $r(S) \geq 3/2$.

Proof. Let $0 < \epsilon < 0.01$ be a fixed number, and $x = \frac{1}{6} - 2\epsilon$, $y = \frac{1}{3} + \epsilon$, $z = \frac{1}{2} + \epsilon$. For any $n = 12k$ (k a -positive integer), define a list $L = L_1L_2L_3$, where L_1 consists of n x 's, L_2 consists of n y's, and L_3 consists of n z 's.

Clearly,

$$L_1^* = \frac{1}{6} n \ , \ (L_1L_2)^* = \frac{1}{2} n \ , \text{ and } (L_1L_2L_3)^* = n \ .$$

Given any on-line algorithm S , let $r_1(n) = S(L_1)/L_1^*$, $r_2(n) = S(L_1L_2)/(L_1L_2)^*$, and $r_3(n) = S(L_1L_2L_3)/(L_1L_2L_3)^*$. We shall prove that

$$\max\{r_1(n), r_2(n), r_3(n)\} \geq 3/2 \ . \qquad (8)$$

This immediately implies that $r(S) \geq 3/2$ and hence the theorem,

Consider the packing of L under algorithm S . We shall gather information about $r_j(n)$ $(1 \leq j \leq 3)$ by examining the resulting packing configurations at points when jn items have been assigned.

Consider the -packing of the first n items (i.e., L_1). Let α_i $(1 \leq i \leq 6)$ be the number of bins containing i pieces of x (Figure 3), then

$$\begin{cases} S(L_1) = \sum_{1 \le i \le 6} \alpha_i \ , \\ n = \sum_{1 \le i \le 6} i\alpha_i \ . \end{cases} \tag{9}$$

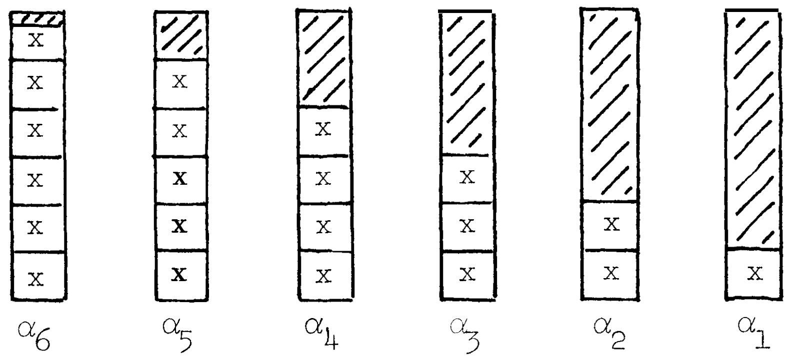

Figure 3. The packing of L_1 by S .

Next we examine the configuration after 2n items are packed (i.e., $L_1 L_2$ has been assigned), A bin is called type (i,ℓ) if there are i x's and ℓ y's in the bin. Let β_1 , β_2 , α_1' , α_1'' , α_1''' , α_2' , α_2'' , α_2''' , α_3' , α_3'' , α_4' , α_4'' be the number of bins of type (0,1) , (0,2) , (1,0) , (1,1) , (1,2) , (2,0) , (2,1) , (2,2) , (3,0) , (3,1) , (4,0) , (4,1) , respectively (see Figure 4). Clearly,

$$
\begin{cases}
\alpha_1 = \alpha_1' + \alpha_1'' + \alpha_1''' , \\
\alpha_2 = \alpha_2' + \alpha_2'' + \alpha_2''' , \\
\alpha_3 = \alpha_3' + \alpha_3'' , \\
\alpha_4 = \alpha_4' + \alpha_4'' .
\end{cases}
\qquad (10)
$$

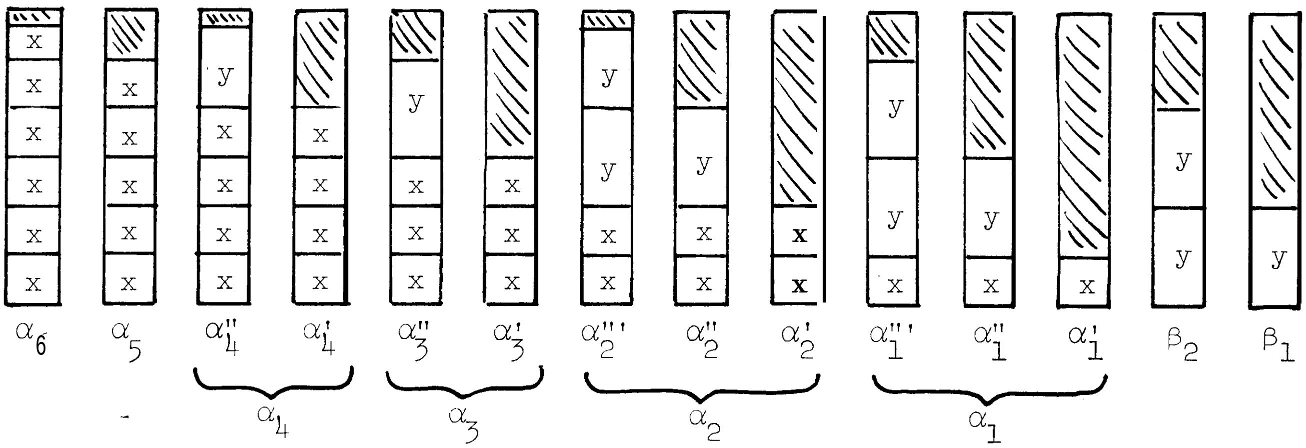

Figure 4. The packing of L_1L_2 by S .

It is easy to see that the only other possible types are (6,0) and (5,0) and there are respectively α_6 and a_5 such bins. The analogue of (9) is

$$
\begin{cases}
S(L_1L_2) = (\alpha_1' + \alpha_1'' + \alpha_1''') + (\alpha_2' + \alpha_2'' + \alpha_2''') + (\alpha_3' + \alpha_3'') + (\alpha_4' + \alpha_4'') + \alpha_5 + \alpha_6 + \beta_1 + \beta_2 , \\
n = (\alpha_1'' + 2\alpha_1''') + (\alpha_2'' + 2\alpha_2''') + \alpha_3'' + \alpha_4'' + \beta_1 + 2\beta_2 ,
\end{cases}
\qquad (11)
$$

where the second equation counts the number of y's.

A lower bound to $S(L_1L_2L_3)$ can be obtained by observing that no z-piece can go into a bin of type (1,2), (2,1), (2,2), (3,1), (4,0), (4,1), (5,0), (6,0), or (0,2), and that no two z-pieces can occupy the same bin. Thus

$$S(L_1L_2L_3) \geq \alpha_1''' + \alpha_2'' + \alpha_2''' + \alpha_3'' + \alpha_4' + \alpha_4'' + \alpha_5 + \alpha_6 + \beta_2 + n \quad . \tag{12}$$

We now define a new set of variables:

$$\begin{cases} \bar{\alpha}_1 = \alpha_1'' \\ \bar{\alpha}_2 = \alpha_1''' + \alpha_2''' \\ \alpha_3 = \alpha_1' + a'_2 + \alpha_3' \\ \bar{\alpha}_4 = \alpha_2'' + \alpha_3'' + \alpha_4'' \\ \bar{\alpha}_6 = \alpha_4' + \alpha_5 + \alpha_6 \quad . \end{cases} \tag{13}$$

Making use of (10) and the positivity of all quantities involved, we obtain from (9), (11) and (12) the following constraints.

$$\begin{cases} S(L_1) = \bar{\alpha}_1 + \bar{\alpha}_2 + \bar{\alpha}_3 + \bar{\alpha}_4 + \bar{\alpha}_6 \\ n \leq \bar{\alpha}_1 + 2\bar{\alpha}_2 + 3\bar{\alpha}_3 + 4\bar{\alpha}_4 + 6\bar{\alpha}_6 \end{cases} \tag{9)'$$

$$\begin{cases} S(L_1L_2) = \bar{\alpha}_1 + \bar{\alpha}_2 + \bar{\alpha}_3 + \bar{\alpha}_4 + \bar{\alpha}_6 + \beta_1 + \beta_2 \\ n = \bar{\alpha}_1 + 2\bar{\alpha}_2 + \bar{\alpha}_4 + \beta_1 + 2\beta_2 \end{cases} \tag{11)'$$

and

$$S(L_1L_2L_3) \geq \alpha_2 + \bar{\alpha}_4 + \bar{\alpha}_6 + \beta_2 + n \quad . \tag{12)'$$

In terms of $r_i(n)$, the above systems can be rewritten as follows.

$$\begin{cases} \frac{1}{6}\, n \cdot r_1(n) = \bar{\alpha}_1 + \bar{\alpha}_2 + \bar{\alpha}_3 + \bar{\alpha}_4 + \bar{\alpha}_6 \\ \frac{1}{2}\, n \cdot r_2(n) = \bar{\alpha}_1 + \bar{\alpha}_2 + \bar{\alpha}_3 + \bar{\alpha}_4 + \bar{\alpha}_6 + \beta_1 + \beta_2 \\ n \cdot r_3(n) \geq \bar{\alpha}_2 + \bar{\alpha}_4 + \bar{\alpha}_6 + \beta_2 + n \\ -\frac{n}{2} \geq -\frac{1}{2}\bar{\alpha}_1 - \bar{\alpha}_2 - \frac{3}{2}\bar{\alpha}_3 - 2\bar{\alpha}_4 - 3\bar{\alpha}_6 \\ -n = -\bar{\alpha}_1 - 2\bar{\alpha}_2 - \bar{\alpha}_4 - \beta_1 - 2\beta_2 \quad . \end{cases} \tag{14}$$

We are now ready to prove (8). If (8) is not true, then we have

$$\begin{cases} \frac{1}{4}\, n > \frac{1}{6}\, n \cdot r_1(n) \quad , \\ \frac{3}{4}\, n > \frac{1}{2}\, n \cdot r_2(n) \quad , \\ \frac{3}{2}\, n > n \cdot r_3(n) \quad . \end{cases} \tag{15}$$

Now adding up all the equations in (14) and (15), we obtain $0 > \frac{1}{2}\bar{\alpha}_1 + \frac{1}{2}\bar{\alpha}_3$, a contradiction. This completes the proof of (8), and hence Theorem 2. □

5. The Technique of ϵ-Improvement.

Given several simple heuristic algorithms in an optimization problem, a practical method to obtain a good solution is to run each algorithm and then select the best solution produced. For example, in the traveling salesman problem, one may produce tours using several heuristic algorithms (see, e.g. [10]) and select the shortest tour. It is hoped that the quality of solution obtained will be much better than using a single fixed algorithm. Implicitly, the success of this idea depends on the hypothesis that different algorithms "favor" different regions in the input space. An interesting research area, so far not much explored, is to analyze the performance (worst case or average case) of such "compound-algorithms". Trying to obtain a better heuristic algorithm than FFD, one possibility is to try such compound algorithms.

There are two difficulties in a direct approach, however. Firstly, there are many algorithms sharing the same worst-case input (e.g. the almost-any-fit algorithms in [5][6]). This eliminates some natural compound algorithms (running FFD and BFD will not improve the worst-case bound). Secondly, the ratio $11/9 = 1.22\cdots$ is very close to 1, and the analysis has to be rather precise to beat this bound. As the analysis for a relatively simple FFD is already complicated, it is likely to be hard to analyze more sophisticated algorithms. We will circumvent these difficulties by focusing on a specific goal -- to find an algorithm with bound $\frac{11}{9} - \epsilon$ for any positive ϵ.

The idea is to locate the part of input space for which FFD may realize its worst-case performance. If the characterization is simple enough, we may be able to design a heuristic algorithm S that has a better performance in this bad region. The compound-algorithm of FFD and S then has a bound better than $\frac{11}{9} - \epsilon$. It turns out that, for many bin-packing algorithms, one can give simple descriptions of small regions covering all the "bad" inputs, as a result of the weight-function type argument used. Thus the bin-packing problem provides an ideal opportunity to try out this idea of "e-improvement".

In this section, we shall illustrate the idea by proving a simpler result about FFD. Consider the restricted problem of bin packing, in which each number in list L is in the range $(0, 1/2]$. It is known [7] that FFD has a -performance ratio 71/60 for this restricted -problem. We shall show that there is a better heuristic algorithm.

We first state a useful lemma.

Lemma 2. Let λ , λ' , μ , ν be constants such that $0 < \lambda < \lambda' \leq 1$, $\mu \geq (1-\lambda)^{-1}$, and $\nu > 1$. Suppose there is a bin-packing algorithm S with running time $O(p(n))$ such that, for any list L consisting of numbers in $(\lambda,\lambda']$, $S(L) \leq \mu L^* + \nu$. If -p(n) is a non-decreasing function of n , then there is an algorithm S' with running time $O(p(n) + n \log n)$ such that $S'(L) \leq \mu L^* + \nu$ for any list L consisting of numbers in $(0,\lambda']$.

Proof. Given an arbitrary list L , the algorithm S' works as follows. In O(n) time, one divides the items into two lists L_1 and L_2 , consisting of numbers in $(\lambda,\lambda']$ and $(0,\lambda]$, respectively, The algorithm S is applied to L_1 to produce a packing using, say N_1 bins. One finishes the packing by -performing a first-fit algorithm on list L_2 , The algorithm clearly works in time $O(p(n) + n \log n)$. We now show that $S'(L) \leq \mu . L^* + \nu$. By assumption, $N_1 \leq \mu . L_1^* + \nu$. If $S'(L) \leq N_1$, then the result follows immediately since $L_1^* \le L^*$, If $S'(L) > N_1$, then in the final packing, all except possibly the last bin must have content greater than $1-\lambda$. This implies that $L^* \ge (1-\lambda)(S'(L)-1)$, and hence $S'(L) \le \frac{1}{1-\lambda} L^* + 1 \le \mu L^* + \nu$. □

The above line of argument appears often in bin-packing analysis (e.g. [7, Lemma 3.3]).

The rest of this section is devoted to proving the following result, based on the general idea outlined earlier.

Theorem 3. Let $\epsilon = 10^{-6}$. There is an $O(n \log n)$-time algorithm S for bin-packing such that, if a list L has all numbers in $(0, 1/2]$, then $S(L) \le \left(\frac{71}{60} - \epsilon\right) \cdot L^* + 5$.

Let $\lambda = 1/7$, $\lambda' = 1/2$, $\mu = 71/60 - \epsilon$, and $\nu = 5$. ByLemma2, we need only prove the theorem assuming that the lists L have all numbers in $(1/7, 1/2]$. For the rest of this section, we restrict ourselves to such lists, although some statements also apply to general lists. The first step is to locate the "bad" input lists.

A Review of the Proof for $FFD(L) \le \frac{71}{60} L^* + 5$.

The proof [5][7] proceeds by defining a function $W(S) \ge 0$ for any finite set S of numbers in $(0, 1/2]$, such that the following properties are satisfied.

Property A1. W is subadditive -- $W\left(\bigcup_i S_i\right) \le \sum_i W(S_i)$.

Property A2. If all elements in L are in $(1/N, 1/2]$, $N \ge 4$, then

$$W(L) \ge FFD(L) - N + 2 .$$

Property A3. If $S = \{x_1, x_2, \ldots, x_m\}$ with $x_i \in (1/7, 1/2]$ and $\sum_i x_i \leq 1$, then

$$W(S) \leq 71/60 .$$

Let X_i be the i-th bin in an optimal packing of L . Properties A1- A3 imply the desired result

$$FFD(L) - 5 \leq W(L) \leq \sum_i W(X_i) \leq \frac{71}{60} L^* . \qquad (16)$$

A Strengthened Analysis,

We have seen from (16) that,

$$FFD(L) \leq \frac{71}{60} L^* + 5 . \qquad (17$$

Notice that we would obtain a bound better than (17), except in the case when almost all X_i have $W(X_i) = 71/60$. Actually, $W(X_i) = 71/60$ only under very special conditions.

Definition. A number x_j in L is called an A, B, C, D , E , or F-piece if x_j is in $1/2, 1]$, $(1/3, 1/2]$, $(1/4, 1/3]$, $(1/5, 1/4]$, $(1/6, 1/5]$, or $(1/7, 1/6]$. We shall use notations such as S = {CCDE} to express the situation $S = \{x_1, x_2, x_3, x_4\}$ with x_1 , x_2 , x_3 , x_4 being a C , C , D , and E-piece, respectively. In a packing, a bin containing a set {CCDE} will be called a CCDE-bin. The notation generalizes obviously to other configurations.

Property A3'. [5] [7]. If $S = \{x_1, x_2, \ldots, x_m\}$ with $x_i \in (1/7, 1/2]$, and $\sum_i x_i \leq 1$, then

$$W(S) \leq 71/60 \quad , \quad \text{if } S = \{BBEF\} \text{ or } \{CDEEE\} \quad ,$$

and

$$W(S) \leq 7/6 \quad , \quad \text{otherwise.}$$

A strengthened form of (17) can now be derived as follows. Let P^* be an optimal packing of L , and X_i the i-th bin in P* $(1 \leq i \leq L^*)$. Assume that there are α bins in P* of the form {BBEF} or {CDEEE} .

Lemma 3. If $a \leq (1 - 60\epsilon)L^*$, then $\mathrm{FFD}(L) \leq \left(\frac{71}{60} - \epsilon\right)L^* \quad 5$.

Proof. From Properties A1, A2, and A3', we have

$$\mathrm{FFD}(L) - 5 \leq W(L) \leq \sum_i W(X_i) \leq \frac{71}{60}\alpha + \frac{7}{6}(L^* - \alpha) \quad .$$

Therefore,

$$\begin{aligned} \mathrm{FFD}(L) &\leq \frac{7}{6} L^* + \frac{1}{60}\alpha + 5 \\ &\leq \left(\frac{71}{60} - \epsilon\right)L^* + 5 \quad . \quad \square \end{aligned}$$

We shall call a list L severe, if in every optimal packing P^* of L , there are more than $(1 - 60\epsilon)L^*$ bins of the form {BBEF} or {CDEEE} . Lemma 3 states that, if a list L is not severe, then the packing produced by FFD has a bound at most $\frac{71}{60} - \epsilon$, strictly less than 71/60 , This concludes the step of identifying "bad" lists, We can finish the proof of Theorem 3, if we can design a heuristic algorithm S such that $S(L) \leq \left(\frac{71}{60} - \epsilon\right)L^* + 5$ for all severe lists L . We shall presently give an algorithm M with running time O(n log n), and prove that $S = M$ has the desired property.

Algorithm M.

Step 1. Sort the input list L; let $(b_1 \leq b_2 \leq \cdots)$, $(c_1 \leq c_2 \leq \cdots)$, $(d_1 \leq d_2 \leq \cdots)$, $(e_1 \leq e_2 \leq \cdots)$, and $(f_1 \leq f_2 \leq \cdots)$ be the sublists of B-pieces, C-pieces, D-pieces, E-pieces, and F-pieces, respectively.

Step 2. For $j = 1,2,\ldots$, put $\{c_j, d_j, e_{3j-2}, e_{3j-1}, e_{3j}\}$ into BIN_j, as long as such a set can fit into one bin and enough pieces are available. [We shall abbreviate the latter clause below as "as long as it is feasible".] Assume that m such bins are formed.

Step 3. For $j = 1,2,\ldots$, put $\{c_{m+j}, d_{m+j}, e_{3m+2j-1}, e_{3m+2j}\}$ into BIN_{m+j}, as long as there are enough pieces available. Assume that k such bins are formed. [Note that a set {CDEE} has sum $\leq \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{5} < 1$, and thus can always fit into a bin.

Step 4. Suppose there are h F-pieces. For $j = 1,2,\ldots$, put $\{b_{2j-1}, b_{2j}, f_j, f_{h-j}\}$ into BIN_{m+k+j} as long as it is feasible. Assume that q such bins are formed.

Step 5. For $j = 1,2,\ldots$, put $\{b_{2q+2j-1}, b_{2q+2j}, e_{3m+2k+j}\}$ into $BIN_{m+k+q+j}$, as long as it is feasible. Assume that ℓ such bins are formed.

Step 6. Pack the remaining E-pieces and F-pieces, respectively, by themselves into new bins using first-fit. Let p be the number of bins formed this way.

Step 7. Pack all the remaining pieces by themselves into new bins using first-fit. Suppose t new bins are used.

End of Algorithm M.

Figure 5 shows a packing produced by Algorithm M.

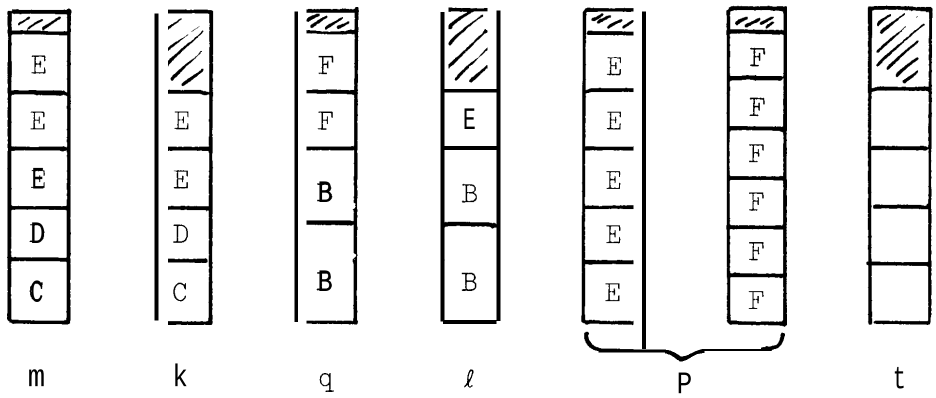

Figure 5. The packing produced by Algorithm M.

Analysis of Algorithm M.

It is easy to implement M so that it runs in O(n log n) time. To complete the proof of Theorem 3, it remains to prove the following result.

Lemma 4. If L is severe, then $M(L) \le \left(\frac{71}{60} - \epsilon\right)L^* + 5$.

Proof. Let P^* be an optimal packing of L . Assume that there are in P^* β bins of the type {BBEF} and γ bins of the type {CDEEE} . As L is severe, we have

$$\beta + \gamma > (1 - 60\epsilon)L^* . \tag{18}$$

We wish to find bounds on the various terms in

$$M(L) = m + k + q + \ell + p + t . \tag{19}$$

In Step 2, for $1 \leq j \leq \lfloor \gamma/5 \rfloor$,

$$c_j + d_j + e_{3j-2} + e_{3j-1} + e_{3j} \leq \text{the } (5j\text{-}4)\text{-th smallest content in all CDEEE-bins in } P^* .$$

Thus, at least $\lfloor \gamma/5 \rfloor$ bins are formed in this step, i.e.,

$$m \geq \lfloor \gamma/5 \rfloor . \qquad (20)$$

Bounds on m+k can be obtained by considering the total available CD-pairs. This gives

$$\gamma + 60\epsilon L^* \times 3 \geq m+k \geq y . \qquad (21)$$

In the last formula, the term $60\epsilon L^* \times 3$ is an upper bound on the number of C-pieces not contained in CDEEE-bins. In Step 4, for $1 \leq j \leq \lfloor \beta/3 \rfloor$,

$$b_{2j-1} + b_{2j} + f_j + f_{h-j} \leq \text{the } (3j\text{-}2)\text{-nd smallest content in all BBEF-bins in } P^* .$$

Therefore,

$$q \geq \lfloor \beta/3 \rfloor . \qquad (22)$$

By considering the number of all F-pieces, we find the following upper bound on q ,

$$\frac{\beta}{2} + 60\epsilon L^* \times 3 \geq q . \qquad (23)$$

To derive bounds on ℓ , we first observe that each B-piece in a BBEF-bin (in P^*) is less than $1 - \frac{1}{3} - \frac{1}{6} - \frac{1}{7} = \frac{5}{14}$. For any two such B-pieces, one can add any E-piece to form a BBE-bin. Thus, a lower bound to ℓ is the minimum of $(\#B)/2$ and $\#E$, where #B and $\#E$ are the numbers of such B-pieces and any E-pieces, respectively, at the start of Step 5. As $\#B \geq 2\beta - 2q$, and $\#E \geq (\beta + 3\gamma) - 3(m+k) \geq \beta - 540\epsilon L^*$ using (21), we obtain

$$\ell \geq \beta - q - 540\epsilon L^* . \tag{24}$$

The total number of B-pieces available gives an upper bound,

$$\beta - q + 60\epsilon L^* \geq \ell . \tag{25}$$

We will now estimate p and t by calculating the number of various -pieces not contained in the first $m+k+q+\ell$ bins. The total number of B-pieces in L is at most $2\beta + (60\epsilon L^* \times 2)$; by (24), at least $2(\beta - 540\epsilon L^*)$ of them are in the first $m+k+q+\ell$ bins. Thus, denoting by $N[Y]$ the number of Y-pieces in the last $p+t$ bins, we have

$$N[B] \leq 1200\epsilon L^* . \tag{26}$$

Similarly, one can show that

$$N[C] \leq 180\epsilon L^* , \tag{27}$$

$$N[D] \leq 240\epsilon L^* , \tag{28}$$

Also one has, using (22),

$$N[F] \leq \frac{1}{3}\beta + 360\epsilon L^* + 2 . \tag{29}$$

The number $N[E]$ satisfies

$$N[E] \leq (\beta + 3\gamma + 300\epsilon L^*) - (3m + 2k + \ell) . \tag{30}$$

Now, using (20),(21),(23), and (24), one has

$$\begin{aligned} 3m + 2k + \ell &= m + 2(m+k) + \ell \\ &\geq \left\lfloor \frac{\gamma}{5} \right\rfloor + 2\gamma + \frac{\beta}{2} - 720\epsilon L^* \\ &\geq \frac{1}{2}\beta + \frac{11}{5}\gamma - 720\epsilon L^* - 1 . \end{aligned} \tag{31}$$

From (30) and (31), we have

$$N[E] \leq \frac{1}{2}\beta + \frac{4}{5}\gamma + 1020\epsilon L^* + 1 . \tag{32}$$

We can now estimate p and t . Using (29) and (32)

$$p \leq \frac{1}{5} N[E] + \frac{1}{6} N[F] + 2 \leq \frac{7}{45} \beta + \frac{4}{25} y + 264 \epsilon L^* + 3 . \qquad (33)$$

From (26) - (28),

$$t \leq N[B] + N[C] + N[D] \leq 1620 \epsilon L^* . \qquad (34)$$

Making use of (21), (25), (33), (34) in (19), we obtain

$$M(L) \leq \frac{52}{45} \beta + \frac{29}{25} y + 2124 \epsilon L^* + 3 .$$

As $\beta + \gamma \leq L^*$, we have

$$M(L) \leq \left(\frac{29}{25} + 2124 \epsilon \right) L^* + 3 .$$

Observing that $\frac{29}{25} + 2124\epsilon < \frac{71}{60} - \epsilon$, we have finally,

$$M(L) \leq \left(\frac{71}{60} - \epsilon \right) L^* + 5 .$$

This proves Lemma 4. □

The proof of Theorem 3 is now complete. □

6. A Polynomial-time Algorithm Better Than FFD.

This section is devoted to proving the following result.

Theorem 4. Let $\epsilon = 10^{-9}$. There is a polynomial-time heuristic algorithm RFFD for bin-packing such that, for any list L,

$$\mathrm{RFFD}(L) \leq \left(\frac{11}{9} - \epsilon\right) L^* + 8 \quad .$$

We shall use the notations $\epsilon = 10^{-9}$, $\delta = 3 \times 10^{-5}$, $\eta = 10^{-4}$, and $\lambda = 1 - \left(\frac{11}{9} - \epsilon\right)^{-1}$. Clearly, $\epsilon = \frac{1}{3} \delta \eta$ and $0 < \lambda < 2/11$.

Although more complicated, the proof of Theorem 4 follows the same pattern as that of Theorem 3. By Lemma 2, it suffices to show the theorem considering only lists L with all elements in $(\lambda, 1]$. We will first prove that, for all such lists, except those of a special type, FFD produces a packing within the desired $\frac{11}{9} - \epsilon$ bound. We then construct a heuristic algorithm EPSI that performs well (below $\frac{11}{9} - \epsilon$) for the exceptional "critical" lists. The compound-algorithm S of FFD and EPSI clearly satisfies $S(L) \leq \left(\frac{11}{9} - \epsilon\right) L^* + 8$ for any list with elements in $(\lambda, 1]$, completing the argument.

A Review of the 11/9 Bound for FFD.

We review below the proof of [5][7] for $\mathrm{FFD}(L) \leq \frac{11}{9} L^* + 4$, if L obeys the following Assumptions 1 and 2. As Assumption 1 can be justified by Lemma 2, and it can be shown directly [5, p. 277, Reduction 3] that any list L violating Assumption 2 has $\mathrm{FFD}(L) \leq \frac{6}{5} L^* + 1$, this would prove $\mathrm{FFD}(L) \leq \frac{11}{9} L^* + 4$ for any list L.

Assumption 1. Let L be a list of numbers in $(2/11, 1]$.

Let P^* be any optimal packing, and P_F the packing produced by FFD. We use X_i to denote the i-th bin in P^*, $1 \leq i \leq L^*$. In any packing, a bin containing an A-piece is called an A-bin, otherwise it is a non-A bin. The number of A-bins in any packing of L is equal to the number of A-pieces in L, which we shall denote as $|A_L|$. Let $\mathcal{F} = \{x \mid x \in L, x \text{ is in a non-A bin in } P_F\}$.

Assumption 2. $\mathcal{F}$ contains at least a C-piece or a D-piece.

Let the function W be defined as in Section 5. The analysis proceeds to define two functions f and g, based on P_F and P^*,

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

For any subset $T \subseteq L$, we write f(T) for $\sum_{x_i \in T} f(x_i)$, and g(T) for $\sum_{x_i \in T} g(x_i)$. The definitions of f and g are complicated ([5]), and were shown to possess the following properties.

Property B1. $\mathcal{F} = \bigcup_{x \in L} f(x)$, $|A_L| \geq \sum_{x \in L} g(x)$.

Property B2. $W(f(X_i)) + g(X_i) \leq \frac{11}{9}(y(X_i) + g(X_i))$, $1 \leq i \leq L^*$

where

$$y(X_i) = \begin{cases} 0, & \text{if } X_i \text{ is an A-bin,} \\ 1, & \text{otherwise.} \end{cases}$$

Also, the following are true from properties of W (see Properties A1 and A2).

Property B3. $W\left(\bigcup_{x \in L} f(x)\right) \leq \sum_{x \in L} W(f(x))$.

Property B4. $W(\mathcal{F}) \geq FFD(L) - |A_L| - 4$.

Summing over X_i in the formula of Property B2, and using Properties B1, B3 and B4, one obtains $FFD(L) \leq \frac{11}{9} L^* + 4$, for any list under Assumptions 1 and 2.

The above is an outline of proof for the bound 11/9 . For our purpose, a strengthened analysis for FFD is needed.

A Strengthened FFD Analysis.

We shall work under a weaker form of Assumption 1.

Assumption 1'. Let L be a list of numbers in $(\lambda,1]$.

Let P_F , P^* , X_i , $\mathcal{F}$ and W have the same meaning as before, We shall say a bin X_i in P^* is regular, if X_i is not of one of the following configurations: an A-bin with 3 pieces, BBC , BCC , CCCD , or CCDD . Otherwise X_i is irregular.

For any list L satisfying Assumption 1' and Assumption 2, one can define f and g such that the following properties are true, in addition to Properties B1 - B4.

Property B5. $W(f(X_i)) + g(X_i) \leq \left(\frac{11}{9} - \delta\right)(y(X_i) + g(X_i))$, if X_i is regular.

Property B6. If X_i is a regular A-bin, then $g(X_i) \geq 1/3$.

The proofs of Properties B1-B6 under Assumptions 1' and 2 follow closely the original analysis [5]. A description of the necessary modifications is given in the Appendix.

We can now give a characterization of lists L for which FFD may have a bad performance.

Theorem 5. Let L be a list satisfying Assumption 1', and P* an optimal packing of L . If there are more than ηL^* regular bins in P^*, then $FFD(L) \leq \left(\frac{11}{9} - \epsilon\right)L^* + 4$.

Proof. If Assumption 2 is not true for L , it can be shown [5, p. 277, Reduction 3] that $FFD(L) \leq \frac{6}{5} L^* + 1$, and the theorem is true. We can therefore suppose that Assumption 2 holds.

Take the formulas in Properties B2, B5, and sum over all X_i , We have

$$\sum_{\text{all } X_i} W(f(X_i)) + g(L) \leq \frac{11}{9} \sum_{\text{all } X_i} (y(X_i) + g(X_i)) - \delta \sum_{\substack{\text{regular} \\ X_i}} (y(X_i) + g(X_i)) \quad . \tag{35}$$

Using Properties B1, B3, and B4, we see that the left hand side of (35) is at least,

$$\text{L.H.S.} \geq FFD(L) - |A_L| - 4 + g(L) \tag{36}$$

Now, to estimate the right hand side of (35), we note that

$$\sum_{\text{all } X_i} (y(X_i) + g(X_i)) = L^* - |A_L| + g(L) \quad . \tag{37}$$

Also, because of Property B6 and the fact that there are at least T/L* regular bins X_i , we have

$$\begin{aligned} \sum_{\substack{\text{regular} \\ X_i}} (y(X_i) + g(X_i)) &\geq (\text{\# of regular non-A bins}) + \frac{1}{3}(\text{\# of regular A-bins}) \\ &\geq \frac{1}{3}(\text{\# of regular } X_i) \\ &\geq \frac{1}{3}\eta L^* \quad . \end{aligned} \tag{38}$$

From (37) and (38), the right hand side of (35) is at most

$$\text{R.H.S.} \le \frac{11}{9}(L^* - |A_L| + g(L)) - \delta \cdot \frac{1}{3}\eta L^* . \qquad (39)$$

Formulas (35), (36), and (39) lead to

$$FFD(L) \le \left(\frac{11}{9} - \frac{1}{3}\delta\eta\right)L^* - \frac{2}{9}(|A_L| - g(L)) + 4 .$$

Noting that $\epsilon = \frac{1}{3}\delta\eta$ and that $|A_L| - g(L) \ge 0$ by Property B1, the theorem follows. □

The EPSI Algorithms.

For the rest of Section 6, all lists are assumed to satisfy Assumption 1'. We shall describe a family of algorithms $EPSI[\alpha_1,\alpha_2,\alpha_3,\alpha_4,\alpha_5,\beta_1,\beta_2,\gamma_1,\gamma_2]$ with non-negative integer parameters $\alpha_1,\alpha_2,\ldots,\gamma_2$. Given a list L with n items, we perform $EPSI[\alpha_1,\alpha_2,\ldots,\gamma_2]$ on L for each possible $0 \le \alpha_1,\alpha_2,\ldots,\gamma_2 \le n$, and pick the best packing. We call this procedure the EPSI algorithm. It will be seen that each $EPSI[\alpha_1,\alpha_2,\ldots,\gamma_2]$ works in O(n log n) time, thus EPSI works in time $O(n^{10} \log n)$.

We call a list L of type $(\alpha_1,\alpha_2,\ldots,\gamma_2)$ if there is an optimal packing of L with $\alpha_1,\alpha_2,\ldots,\gamma_2$ bins of type ACD , ADD , ADE , AEE, ACE, BBC , BCC , CCD , CCDD , respectively. Note that a list can be of several types. A list L is critical, if it is of some type $(\alpha_1,\alpha_2,\ldots,\gamma_2)$ with $\alpha_1+\alpha_2+ \cdots +\gamma_2 \ge (1-\eta)L^*$. The aim of $EPSI[\alpha_1,\alpha_2,\ldots,\gamma_2]$ is to produce a packing using less than $\frac{11}{9} - \epsilon$ times the minimum bins needed, for any critical list of type $(\alpha_1,\alpha_2,\ldots,\gamma_2)$. This ensures that EPSI has a bound better than $\frac{11}{9} - \epsilon$ for any critical list, Together with Theorem 5, which ensures a $\frac{11}{9} - \epsilon$ bound for non-critical lists, it completes the proof of Theorem 4 as stated at the beginning of this section.

Given a list L, and parameters $\alpha_1, \alpha_2, \ldots, \gamma_2$, we shall presently describe the action of EPSI$[\alpha_1, \alpha_2, \ldots, \gamma_2]$. If any of the described steps cannot be accomplished, it is understood that the packing of list L may then proceed arbitrarily.

Firstly, L is sorted in ascending order, Then we pack various pieces into four classes of bins according to the following rules.

Let $a_1 \leq a_2 \leq a_3 \leq \ldots$, $b_1 \leq b_2 \leq b_3 \leq \ldots$, $c_1 \leq c_2 \leq c_3 \leq \ldots$, ... be the lists of A-pieces, B-pieces, C-pieces, . . . , etc.

Step 1. Class 1-bins: First put $\{b_{2j-1}, b_{2j}\}$ into BIN_j, $1 \leq j \leq \beta_1 + \lfloor \beta_2/2 \rfloor$. Then, for $j = 1,2,\ldots,\lfloor \beta_1/2 \rfloor$, put the largest available fitting C-piece into BIN, ,

Step 2. Class 2-bins: Let $c'_1 \leq c'_2 \leq \ldots$ be the remaining C-pieces, Put $\{c'_{3j-2}, c'_{3j-1}, c'_{3j}\}$ into BIN_j, $1 \leq j \leq \gamma_1$. For $j = 1,2,\ldots,\lfloor \gamma_1/3 \rfloor$, put the largest fitting D-piece into BIN_j.

Step 3. Class 3-bins:

(a) Let $d'_1 \leq d'_2 \leq \ldots$ be the remaining D-pieces. Define $m = \lfloor \alpha_1/2 \rfloor + \lfloor \alpha_2/2 \rfloor$. For $1 \leq j \leq m$, put $\{d'_{2j-1}, d'_{2j}\}$ into BIN_j. Then, for $j = 1,2,\ldots,m$, put the largest fitting a_i into BIN_j.

(b) Define $m' = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 - m$. Let $a'_1 \leq a'_2 \leq \ldots \leq a'_m \leq \ldots$ be the list of A-pieces remaining. Put a total of $\lceil \alpha_1/2 \rceil + \alpha_5$ C-pieces, $\lceil \alpha_2/2 \rceil + \alpha_3$ D-pieces, and α_4 E-pieces into BIN_{m+1} to $BIN_{m+m'}$, one piece in each bin. Now, put a'_i into BIN_{m+i}, for $1 \leq i \leq m'$.

Step 4. Class 4-bins: For each $Y \in \{A, B, C, D, E\}$, pack all the Y-pieces first-fit by themselves.

We need some preliminary results before analyzing EPSI.

Definition. Let $Y = (y_1, y_2, \ldots, y_m)$ and $Z = (z_1, z_2, \ldots, z_p)$ be two lists of real numbers. The Cartesian product of Y and Z is $Y \times Z = \{(y_i, z_j) \mid 1 \le i \le m, 1 \le j \le p\}$. A partial match between Y and Z is a subset $\Phi \subseteq Y \times Z$ such that (i) $y_i + z_j \le 1$ for all $(y_i, z_j) \in \Phi$, and (ii) any two distinct (y_i, z_j) and $(y_{i'}, z_{j'})$ in Φ have $i \ne i'$ and $j \ne j'$. Let $\psi(Y,Z)$ denote the maximum possible size of $|\Phi|$. A partial match Φ is a maximum partial match, if $|\Phi| = \psi(Y,Z)$. For any partial match Φ between Y and Z, the range Z_Φ is the multiset $\{z_j \mid (y_i, z_j) \in \Phi$ for some $y_i \in Y\}$. (Thus, $|Z_\Phi| = |\Phi|$). Let $\bar{Z}_\Phi = Z - Z_\Phi$.

The following procedure clearly generates a partial match,

Algorithm PM(Y,Z):

Sort Y into $y_1 \le y_2 \le \ldots \le y_m$; sort Z into $z_1 \le z_2 \le \ldots \le z_p$,
    keep the elements of Z in an array T ($T[i] \leftarrow z_i$, $1 \le i \le p$) ;

$\Phi \leftarrow \emptyset$; $k \leftarrow p$;

for i := 1 until m do
    begin Search T[k] , T[k-1] , . . . to find the largest $j \le k$ satisfying
            $y_i + z_j \le 1$; if j does not exist, halt;
        $\Phi \leftarrow \Phi \cup \{(y_i, z_j)\}$;
        $k \leftarrow j-1$;
    end

END of Algorithm PM.

Lemma 5. Algorithm PM(Y,Z) works in time O(n log n) , where $n = |Y| + |Z|$. Furthermore, the partial match Φ generated is a maximum partial match between Y and Z .

Proof. The O(n log n) -time bound is obvious. To prove the other assertion, suppose PM(Y,Z) sorts Y and Z into $y_1 \leq y_2 \leq \ldots \leq y_m$ and $z_1 \leq z_2 \leq \cdots \leq z_p$, and produces $\Phi = \{(y_1,z_{i_1}),(y_2,z_{i_2}),\ldots,(y_s,z_{i_s})\}$. Clearly $i_1 > i_2 > \ldots > i_s$.

Now assume that there exists a partial match $\Phi' = \{(y_{j_1},z_{k_1}),(y_{j_2},z_{k_2}),\ldots,(y_{j_t},z_{k_t})\}$ with $t > s$. We will show that it leads to a contradiction, With no loss of generality, assume that $j_1 < j_2 < \ldots < j_t$. This implies that $y_1 \leq y_{j_1}$, $y_2 \leq y_{j_2}$, ... , etc., and therefore $\Phi'' = \{(y_1,z_{k_1}),(y_2,z_{k_2}),\ldots,(y_t,z_{k_t})\}$ is also a partial match, A moment's thought reveals that $\Phi''' = \{(y_1,z_{k'_1}),(y_2,z_{k'_2}),\ldots,(y_t,z_{k'_t})\}$ must also be a partial match, where $k'_1 > k'_2 > \ldots > k'_t$ is the sorted sequence of $(k_1,k_2,\ldots,k_t)$. Based on the description of PM , a simple induction argument gives $i_1 \geq k'_1$, $i_2 \geq k'_2$, . . . , $i_s \geq k'_s$. But this implies that PM should have found a $z_{i_{s+1}}$ with $z_{i_{s+1}} + y_{s+1} \leq 1$ ($z_{k'_{s+1}}$ is a candidate). This is a contradiction. □

Definition. Let X and Y be two multisets of real numbers. We say that X is dominated by Y if the i-th smallest element in X is no greater than the i-th smallest element in Y , for all $1 \leq i \leq |X| \leq |Y|$. A list X' is dominated by a list Y' if the corresponding multisets X and Y satisfy this relation.

Lemma 6. Let X , Y and Z be finite lists with X dominated by Y . Then

(a) $\psi(X,Z) \geq \min\{|X|, \psi(Y,Z)\}$,

(b) Let Φ be a partial match generated by PM(X,Z) , and Φ' any partial match between Y and Z with $|\Phi'| = |\Phi|$. Then $\bar{Z}_\Phi$ is dominated by $\bar{Z}_{\Phi'}$.

Proof. Let the sorted lists of X , Y , Z be $x_1 \leq x_2 \leq \ldots \leq x_m$, $y_1 \leq y_2 \leq \cdots \leq y_n$, $z_1 \leq z_2 \leq \ldots \leq z_p$, respectively.

(a) Let $\{(y_1, z_{i_1}), (y_2, z_{i_2}), \ldots, (y_s, z_{i_s})\}$ be the maximum partial match generated by PM(Y,Z) (Lemma 5). Let $\ell = \min\{|X|, s\}$. Then $\{(x_1,z_{i_1}),(x_2,z_{i_2}),\ldots(x_\ell,z_{i_\ell})\}$ is a partial match between X and Z, as $x_j \leq y_j$ by assumption. This proves $\psi(X, Z) \geq \min\{|X|, \psi(Y, Z)\}$.

(b) Let $\Phi = \{(x_1,z_{i_1}),(x_2,z_{i_2}),\ldots,(x_\ell,z_{i_\ell})\}$ with $i_1 > i_2 > \cdots > i_\ell$, and $\Phi' = \{(y_{j_1},z_{k_1}), (y_{j_2},z_{k_2}),\ldots,(y_{j_\ell},z_{k_\ell})\}$ with $j_1 < j_2 < \ldots < j_\ell$, As in the proof of Lemma 5, it can be shown that $\Phi'' = \{(x_1,z_{k'_1}),(x_2,z_{k'_2}),\ldots,(x_\ell,z_{k'_\ell})\}$ is a partial match between X and Z , when $k'_1 > k'_2 > \ldots > k'_\ell$ is the sorted sequence of $(k_1,k_2,\ldots,k_\ell)$. A simple induction argument then shows that $i_1 \geq k'_1$, $i_2 \geq k'_2$, ..., $i_\ell \geq k'_\ell$. This implies that, for each $1 \leq q \leq p$, $|\{i_t \mid i_t > q\}| \geq |\{k'_t \mid k'_t > q\}|$. Hence, we have

Fact 7. For each $1 \leq q \leq p$, $|\{i_t \mid i_t \leq q\}| \leq |\{k'_t \mid k'_t \leq q\}|$.

Now the multisets $\bar{Z}_\Phi$ and $\bar{Z}_{\Phi'}$ are obtained from Z by deleting $(z_{i_\ell}, z_{i_{\ell-1}}, \ldots, z_{i_1})$ and $(z_{k'_\ell}, z_{k'_{\ell-1}}, \ldots, z_{k'_1})$, respectively. Write $\bar{Z}_\Phi = \{z_{u_1}, z_{u_2}, \ldots, z_{u_c}\}$ and $\bar{Z}_{\Phi'} = \{z_{v_1}, z_{v_2}, \ldots, z_{v_c}\}$, where $u_1 < u_2 < \ldots < u_c$ and $v_1 < v_2 < \cdots < v_c$. Then, for each $1 \leq s \leq c$, $u_s = s + |\{i_t \mid i_t \leq u_s\}|$, and (# of $v_b \leq u_s$) $= u_s - |\{k'_t \mid k'_t \leq u_s\}|$.

Using Fact 7, we have for each $1 \leq s \leq c$,
(# of $v_b \leq u_s$) $\leq u_s - |\{i_t \mid i_t \leq u_s\}| = s$, and thus $v_s \geq u_s$. We have shown that $z_{u_s} < z_{v_s}$ for each $1 \leq s \leq c$, completing the proof that $\bar{Z}_\Phi$ is dominated by $\bar{Z}_{\Phi'}$. ❑

We now analyze the algorithm EPSI.

<u>Lemma 1</u>. For a list L of type $(\alpha_1, \alpha_2, \ldots, \gamma_2)$, every step of EPSI$[\alpha_1, \alpha_2, \ldots, \gamma_2]$ can be carried out.

<u>Proof</u>. Let P^* be an optimal packing of L with $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \beta_1, \beta_2, \gamma_1, \gamma_2$ bins of types ACD, ADD, ADE, AEE, BBC, BCC, CCCD, CCDD, respectively.

(i) <u>Step 1 can be done.</u>

As there are enough $(2\beta_1 + \beta_2)$ B-pieces in L, we need only show that the procedure can put $\lfloor \beta_1/2 \rfloor$ C-pieces into class-1 bins. We define the following multisets: $X = \{b_{2j-1} + b_{2j} \mid 1 \leq j \leq \lfloor \beta_1/2 \rfloor\}$,
$Y_1 = \{b' + b'' \mid \{b', b'', c\}$ is a BBC-bin in $P^*\}$,
$Y = \{y \mid y$ is the $(2j-1)$-st smallest of Y_1 for some $1 \leq j \leq \lfloor \beta_1/2 \rfloor\}$,
and $Z = \{$all C-pieces in L$\}$. As $b_{2j-1} + b_{2j}$ is no greater than the $(2j-1)$-st element in Y_1, it follows that X is dominated by Y. Also $\psi(Y, Z) = \lfloor \beta_1/2 \rfloor$. It follows from Lemma 6(a) that $\psi(X, Z) = \lfloor \beta_1/2 \rfloor = |X|$. As Step 1 is essentially the execution of PM(X,Z), that it can be accomplished

is guaranteed by Lemma 5. Finally we notice an important property following from Lemma 6(b).

Let Φ' be the partial match between Y and Z, defined by $\{(b'+b'', c) \mid \{b',b'',c\}$ has the $(2j-1)$-st smallest $b'+b''$ among BBC-bins in P^* for some $1 \le j \le \lfloor \beta_1/2 \rfloor \}$. According to Lemma 6(b), $\bar{Z}_\Phi$, the set of remaining C-pieces $c_1' \le c_2' \le \dots$, is dominated by $\bar{Z}_{\Phi'}$. It follows that the set of the first $3\gamma_1$ pieces in $c_1' \le c_2' \le \dots$ is dominated by the set of $3\gamma_1$ C-pieces in the CCCD-bins in P^*.

(ii) Step 2 can be carried out.

By the preceding remark, we have for $1 \le j \le \lfloor \gamma_1/3 \rfloor$, $c_{3j-2}' + c_{3j-1}' + c_{3j}'$ is no greater than the $(3j-2)$-nd smallest element of the multiset $\{c + c' + c'' \mid \{c,c',c'',d\}$ is a CCCD-bin in $P^*\}$. An argument similar to that in (i) shows that Step 2 can be accomplished as specified, and that the first $\alpha_1 + 2\alpha_2 + \alpha_3$ in the remaining D-pieces $d_1' \le d_2' \le \dots$ are dominated by the set of D-pieces in the ACD, ADD and ADE-bins in P^*.

(iii) Step 3 can be carried out.

Step 3(a): The preceding statement implies that, for $1 \le j \le m$, $d_{2j-1}' + d_{2j}' \le$ the $(2j-1)$-st smallest in the multiset $\{c + d \mid \{c,d,a\}$ is an ACD-bin in $P^*\} \cup \{d + d' \mid \{a,d,d'\}$ is an ADD-bin in $P^*\}$. As in (i) and (ii), this fact together with Lemmas 5 and 6 can be used to prove that Step 3(a) can be done.

Step 3(b): As each A-piece in an ACD, ADD, ADE, AEE, or ACE-bin is less than $1 - \frac{1}{6} - \frac{1}{6} = \frac{2}{3}$, there are at least $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5$ A-pieces in L that are less than $2/3$. At most m of these A-pieces are packed in Step 3(a). Therefore,

$a'_1 \le a'_2 \le \cdots \le a'_{m'} \le 2/3$. Since each a'_i can fit with any C-piece (or D-piece, or E-piece) in a bin, Step 3(b) can be done provided the specified number of C , D , E-pieces exist. This latter fact can be easily verified.

(iv) Step 4 can always be done.

This proves Lemma 7. □

Lemma 8. Let L be a critical list satisfying Assumption 1' and of type $(\alpha_1, \alpha_2, \ldots, \gamma_2)$, and N_L the number of bins used by EPSI$[\alpha_1, \alpha_2, \ldots, \gamma_2]$ on L . Then

$$N_L \le \left(\frac{6}{5} + 5\eta\right)L^* + 8 .$$

Proof. To begin with, we note that

$$N_L = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 + \beta_1 + \lfloor \beta_2/2 \rfloor + \gamma_1 + (\text{\# of class 4-bins}). \quad (40)$$

We now bound the number of class 4-bins. The total number of C-pieces in L is at most $\alpha_1 + \alpha_5 + \beta_1 + 2\beta_2 + 3\gamma_1 + 2\gamma_2 + 3\eta L^*$. As there are $\lceil \alpha_1/2 \rceil + \alpha_5 + \lfloor \beta_1/2 \rfloor + 3\gamma_1$ C-pieces in class 1-3 bins, the number of C-pieces packed in class 4-bins is at most $\lfloor \alpha_1/2 \rfloor + \lceil \beta_1/2 \rceil + 2\beta_2 + 2\gamma_2 + 3\eta L^*$. A similar counting gives the following upper bounds on the numbers of A-pieces, B-pieces, . . . in class 4-bins.

$$\begin{cases} \#A \le \eta L^* , \\ \#B \le 2\eta L^* + 1 , \\ \#C \le 3\eta L^* + \lfloor \alpha_1/2 \rfloor + \lceil \beta_1/2 \rceil + 2\beta_2 + 2\gamma_2 , \\ \#D \le 4\eta L^* + 1 + \lceil \alpha_2/2 \rceil + \lceil \frac{2}{3}\gamma_1 \rceil + 2\gamma_2 , \\ \#E < 5\eta L^* + \alpha_3 + \alpha_4 + \alpha_5 . \end{cases} \quad (41)$$

Clearly,

$$\# \text{ of class } 4\text{-bins} \le \#A + \frac{1}{2}(\#B) + \frac{1}{3}(\#C) + \frac{1}{4}(\#D) + \frac{1}{5}(\#E) + 5 . \qquad (42)$$

From (40), (41), and (42), we obtain

$$N_L \le \frac{7}{6}\alpha_1 + \frac{9}{8}\alpha_2 + \frac{6}{5}(\alpha_3+\alpha_4+\alpha_5) + \frac{7}{6}(\beta_1+\beta_2+\gamma_1+\gamma_2) + 5\eta L^* + 8 . \qquad (43)$$

As $L^* \ge \alpha_1 + \alpha_2 + \cdots + \gamma_2$, we obtain from (43),

$$N_L \le \frac{6}{5} L^* + 5\eta L^* + 8 \quad . \qquad \square$$

Lemma 9. The algorithm $EPSI[\alpha_1,\alpha_2,\ldots,\gamma_2]$ can be implemented to run in time $O(n \log n)$ for list L with n numbers and parameters $\alpha_1,\alpha_2,\ldots,\gamma_2 \le n$.

Proof. Steps 1, 2, and 3 (a) are executions of algorithm PM , which runs in time $O(n \log n)$. The other steps involve sorting and first-fit, and all can be done in $O(n \log n)$ time. $\square$

Theorem 6. The algorithm EPSI runs in polynomial time. For any critical list L satisfying Assumption 1', $EPSI(L) \le \left(\frac{11}{9} - \epsilon\right)L^* + 8$.

Proof. From Lemma 9 and the definition of EPSI, the algorithm runs in $O(n^{10} \log n)$ time. The rest of the theorem follows from the definition of EPSI, Lemma 8, and the fact $\frac{6}{5} + 5\eta < \frac{11}{9} - \epsilon$. $\square$

Theorem 5 and Theorem 6 imply Theorem 4, hence the existence of a heuristic better than FFD.

7. How Well Can An O(n) -time Algorithm Perform?

We have shown that 11/9 is not the limit on the performance ratio of polynomial-time bin packing algorithms. A most interesting open question is whether there exists such a limit to r(S) . Carey and Johnson [3] showed that, unless P = NP , no polynomial heuristic algorithm for graph coloring can guarantee to use less than twice the minimum number of colors needed. A similar result for bin packing would be especially interesting, since the known achievable bound on the performance ratio is already close to 1 . A more modest question along this line was raised in [7], namely, how well can an O(n) -time algorithm perform? A natural computation model is the decision tree model, counting only branching operations [6][9]. It would be interesting to prove the existence of an $\epsilon > 0$ such that, for any O(n) -time bin packing algorithm S , one must have $r(S) \geq 1+\epsilon$. We have not succeeded in proving such an assertion. However, a result of this spirit can be shown for a closely related problem, and it may throw some light on the present bin packing problem.

Consider the generalized bin packing problem discussed in [2]. Let $L = (\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n)$ bealist of d-dimensional vectors $(d \geq 1)$, with each component of the vectors in the intervals $(0,1]$. The problem is to pack these vectors into a minimum number of bins, such that the sum $\vec{v}$ of vectors in any bin has $v_i \leq 1$ for all $1 \leq i \leq d$. (When $d = 1$, this is just the bin-packing problem we have discussed,) The problem is clearly NP-complete for any fixed $d \geq 1$. For any heuristic algorithm, let r(S) denote the performance ratio as before. A simple extension S of the O(n) -time Next-Fit Algorithm [5][6] gives r(S) = 2d . We are interested in a universal lower bound to r(S) for any O(n) -time algorithm.

We consider the following decision tree model. Let S be an algorithm for the generalized d-dimensional bin packing. For each $n > 0$ the action of S on lists of n items $L = (\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n)$ can be represented by a ternary tree $T_n(S)$. Each internal node of $T_n(S)$ contains a test " $h(\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n)$: 0 ", where h is a rational function. For any input L the algorithm moves down the tree, testing and branching according to the result ($h < 0$, $h = 0$, or $h > 0$), until a leaf is reached. At the leaf, a packing valid for all lists that lead to this leaf is produced. The cost of S for input of size n $C_n(S)$, is defined to be the number of tests made in the worst case, i.e., the height of $T_n(S)$.

Theorem 7. Let S be an algorithm for the generalized d-dimensional bin packing. If there exists a constant $a > 0$ such that $C_n(S) \leq an$ for all n, then $r(S) \geq d$.

Proof. The case $d = 1$ is trivial. We therefore assume that $d > 1$. Let $n > 0$ be any integer. Define a sequence $\epsilon_0, \epsilon_1, \epsilon_2, \ldots, \epsilon_n$ such that

$$\epsilon_0 = 1/d^2 ,$$
$$\epsilon_i > (d-1)\epsilon_{i+1} , \quad 0 \leq i \leq n-1 . \tag{44}$$

Let $\vec{x}_{[i,j]}$ be the vector $(\underbrace{\epsilon_i, \epsilon_i, \ldots, \epsilon_i}_{j-1}, 1-(d-1)\epsilon_i, \underbrace{\epsilon_i, \ldots, \epsilon_i}_{d-j})$, for each $1 \leq i \leq n$, $1 \leq j \leq d$.

Consider the list $L_n = (\vec{x}_{[1,1]}, \vec{x}_{[1,2]}, \ldots, \vec{x}_{[1,d]}, \vec{x}_{[2,1]}, \ldots, \vec{x}_{[n,d]})$ with dn vectors. Clearly $L_n^* = n$, as $\sum_{1 \leq j \leq d} \vec{x}_{[i,j]} = (1,1,\ldots,1)$ for each- $1 \le i \le n$. Let Γ_n be the set of permutations of the dn elements in $E_n = \{[i,j] \mid 1 \le i \le n , 1 \le j \le d\}$. For each $\sigma \in \Gamma_n$, denote by $L_n(\sigma)$ the list $(\vec{x}_{\sigma(1)}, \vec{x}_{\sigma(2)}, \ldots, \vec{x}_{\sigma(dn)})$. Obviously, $L_n(\sigma)^* = L_n^* = n$, We shall prove that, for any fixed $\delta > 0$, if n is large enough, then there exists a $\sigma \in \Gamma_n$ such that $S(L_n(\sigma)) > (d-\delta)L_n(\sigma)^*$. This would imply the theorem.

If the above assertion is false, then there exists a $\delta > 0$ such that $S(L_n(\sigma)) \le (d-\delta)n$ for all sufficiently large n . We will derive a contradiction.

Fact 8. In any packing, $\vec{x}_{[i,j]}$ and $\vec{x}_{[i',j']}$ cannot be in the same bin if $i \ne i'$.

Proof. It follows immediately from the definition of $\vec{x}_{[i,j]}$. □

Fact 9. Let ℓ be any leaf of T,(S) , and $\Sigma(\ell)$ be the set of lists L,(a) that will lead to ℓ . Then $|\Sigma(\ell)| \le (dn)! \cdot (cn)^{-\delta n/d}$ for some fixed constant c .

Proof. In the packing produced at ℓ , there must be at least $p = \delta n/d$ bins containing two items or more, because $S(L_n(y)) \le (d-\delta)n$, In other words, any in-put list $(\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_{dn})$ reaching ℓ must satisfy a set of inequalities of the following form.

$$
\begin{aligned}
\vec{y}_{k_1} + \vec{y}_{k_2} &\le (1,1,\ldots,1) , \\
\vec{y}_{k_3} + \vec{y}_{k_4} &\le (1,1,\ldots,1) , \\
&\vdots \\
\vec{y}_{k_{2p-1}} + \vec{y}_{k_{2p}} &< (1,1,\ldots,1) ,
\end{aligned}
\qquad (45)
$$

I

where " < " means componentwise inequalities, and all k_j are distinct.

An upper bound to $|\Sigma(\ell)|$ is given by the number of $L_n(\sigma)$ satisfying (45). Taking Fact 8 into consideration, we have

$$\begin{aligned} |\Sigma(\ell)| &\leq (nd(d-1))^p \times (dn-2p)! \\ &\leq n^p d^{2p} \times (dn-2p)! \quad . \end{aligned} \tag{46}$$

We now show that $n^p d^{2p} \times (dn-2p)! = (dn)! \times O((n/(4e^2))^{-p})$. There are two cases. If $2p \leq dn/2$, then

$$n^p d^{2p} \times (dn-2p)! \leq n^p d^{2p} \times \frac{(dn)!}{(dn-2p+1)^{2p}} \leq (dn)! \frac{n^p d^{2p}}{(dn/2)^{2p}} = (dn)! \times (n/4)^{-p} .$$

If $2p > dn/2$, then

$$n^p d^{2p} \times (dn-2p)! \leq (dn)! \times \frac{n^p d^{2p}}{(2p)!} \frac{1}{\binom{dn}{2p}} \leq (dn)! \times \frac{n^p d^{2p}}{(2p)!}$$

$$= (dn)! \times O\left(\left(\frac{e\sqrt{n}\,d}{2p}\right)^{2p}\right) = (dn)! \times O((n/(4e^2))^{-p}) .$$

We have used Stirling's approximation [9] in the last derivation. This proves Fact 9. □

As there are at most 3^{an} leaves, the total number of lists $L_n(\sigma)$ reaching any leaf of $T_n(S)$ is at most [illegible] $.(cn)^{-\delta n/d} \times 3^{an} <$ (dn)! for all sufficiently large n . This contradicts the fact that there are (dn)! possible lists $L_n(\sigma)$. This proves Theorem 7. □

8. Concluding Remarks.

We list some problems for further research.

(1) The e-improvement technique may be useful in other NP-complete problems, for example, in the scheduling of tasks on a multiprocessor system [4]. This technique seems to be particularly suitable for scheduling-type problems, when the set of possible worst-case input can be identified. For instance, it can be used to show that $r(S) < 2$ for the Next-2 fit bin-packing [5] [6]. It may be of interest to mention that, although the algorithm RFF was constructed and analyzed in a more conventional way as presented, it was first obtained in a fashion very similar to the process in Sections 5 and 6. Thus, the e-improvement viewpoint can provide a starting point for substantially improved algorithms.

(2) Let r(on-line) be $\inf\{r(S)\}$ over all on-line algorithms S , We have shown that $1.5 \leq$ r(on-line) $\leq 1.66\cdots$, It is of interest to determine it more precisely.

(3) Find and analyze off-line algorithms S with $r(S)$ "substantially" better than $11/9$.

(4) Is there an $\epsilon > 0$ such that finding a packing of L using less than $(1+\epsilon)L^*$ bins is NP-complete? Is there an $\epsilon > 0$ such that every $O(n)$ -time algorithm S (say, in the decision tree model described in Section 7) has $r(S) \geq 1+\epsilon$?

References

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, Mass., 1974.

[2] M. R. Garey, R. L. Graham, D. S. Johnson, and A. C. Yao, "Multiprocessor scheduling as generalized bin-packing," Journal of Combinatorial Theory 21 (1976), 257-298.

[3] M. R. Garey and D. S. Johnson, "The complexity of near-optimal graph coloring," Journal ACM 23 (1976), 43-49.

[4] R. L. Graham, "Bounds on the performance of scheduling algorithms," in Computer and Job/Shop Scheduling Theory, edited by E. G. Coffman, Jr., Wiley, New York, 1976.

[5] D. S. Johnson, "Near optimal bin packing algorithm," Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, Mass., June 1973.

[6] D. S. Johnson, "Fast algorithms for bin packing," J. Comput, System Sci, 8 (1974), 272-314.

[7] D. S. Johnson, A. Demers, J. D. Ullman, M. R. Garey, and R. L. Graham, "Worst-case performance bounds for simple one-dimensional packing algorithms," SIAM J. on Computing 3 (1974), 299-325.

[8] R. M. Karp, "Reducibility among combinatorial problems," in Complexity of Computer Computations, edited by R. E. Miller and J. W. Thatcher, Plenum Press, New York, 1972.

[9] D. E. Knuth, The Art of Computer Programming, Vol. 3, Sorting and Searching, Addison-Wesley, Reading, Mass., 1973.

[10] D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis, "An analysis of several heuristics for the traveling salesman problem," SIAM J. on Computing 6 (1977), 563-581.

Appendix. The Strengthened FFD Analysis in Section 6.

At the beginning of Section 5, we listed some facts (Properties B1 - B6) which lead to the proof of Theorem 5. In this appendix, we will give more details on how these facts can be obtained from the original analysis of FFD in [5][7].

In [5], Properties B1 - B4 are proved under the following assumptions on the list L. Let P_F be an FFD packing and P' an optimal packing of L, Write the items in L as $x_1 \geq x_2 \geq \ldots \geq x_n$. Let $\mathcal{F} = \{x_i \mid x_i$ is not in an A-bin in $P_F\}$.

Assumption 1. All x_i are in $(2/11, 1]$.

Assumption 2. $\mathcal{F}$ contains at least a C-piece or a D-piece.

Assumption 3. The smallest piece x_n goes into a non-A bin in P_F, i.e., $x_n \in \mathcal{F}$.

We make the following observations. Let $\epsilon = 10^{-9}$, $\lambda = 1 - \frac{1}{\frac{11}{9} - \epsilon}$, $\delta = 3 \times 10^{-5}$, and $\eta = 10^{-4}$, as in Section 6.

Observation 1. One can replace Assumption 1 by a weaker constraint, Assumption 1', that $x_i \in (\lambda, 1]$.

Observation 2. One can replace P' by any packing of L.

Observation 3. Property B2 comes from the following facts.

$$W(f(X_i)) \leq \frac{2}{9} g(X_i), \text{ if } X_i \text{ is an A-bin in } P',$$

$$W(f(X_i)) - \frac{2}{9} g(X_i) \leq \frac{11}{9}, \text{ if } X_i \text{ is a non-A bin in } P'.$$

One can make stronger statements for regular bins X_i.

$$W(f(X_i)) \leq \frac{3}{14} g(X_i) \text{ , if } X_i \text{ is a regular A-bin in } P' ,$$

$$W(f(X_i)) - \frac{2}{9} g(X_i) \leq \frac{73}{60} , \text{ if } X_i \text{ is a regular non-A bin in } P' .$$

Observation 4. $g(X_i) \leq 50$, for any bin X_i in P' .

Observation 5. $g(X_i) \geq 1/3$, if X_i is a regular A-bin in P' .

Observations 3 and 4 lead to Property B5, and Observation 5 is Property B6. Therefore, if L satisfies Assumptions 1', 2 and 3, and P' is any packing of L , then one can define f and g such that Properties B1 - B6 are true.

It remains to show that Assumption 3 can be dropped. Let $L = (x_1 \geq x_2 \geq \ldots \geq x_n)$ be a list satisfying Assumptions 1' and 2, PF the FFD--packing of L , P* an optimal packing of L , and $\mathcal{F} = \{x_i \mid x_i \text{ is in a non-A bin in } P_F\}$. Suppose x_m is the smallest non-A -piece in $\mathcal{F}$. We consider the list $L' = (x_1, x_2, \ldots x_m)$, and let P' be the packing of L' , obtained from P^* by deleting pieces $x_{m+1}, x_{m+2}, \ldots, x_n$. Then L' satisfies Assumptions 1', 2 and 3. Applying the -previous results, we can define functions f' , g' satisfying B1 - B6 for the list L' . Now, we define functions f and g for the list L by

$$f(x_i) = \begin{cases} f'(x_i) & \text{if } x_i \in L' , \\ \emptyset & \text{otherwise,} \end{cases}$$

and

$$g(x_i) = \begin{cases} g'(x_i) & \text{if } x_i \in L' \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, $FFD(L) = FFD(L')$, $|A_L| = |A_{L'}|$, and the set $\mathcal{F}$ is the same for both L and L', Also notice that a regular bin in P^* must also be regular in P', and a bin in P^* is an A-bin if and only if it is an A-bin in P'. With these facts, it is straightforward to verify that Properties B1 - B6 are satisfied for L with this choice of f and g .