JOURNAL OF COMPLEXITY 1, 65–106 (1985)

A 71/60 Theorem for Bin Packing*

DAVID S. JOHNSON AND MICHAEL R. GAREY

AT&T Bell Laboratories, Murray Hill, New Jersey 07974

The FIRST FIT DECREASING algorithm for bin packing has long been famous for its guarantee that no packing it generates will use more than 11/9 = 1.222... times the optimal number of bins. We present a simple modified version that has essentially the same running time, should perform at least as well on average, and yet provides a guarantee of 71/60 = 1.18333... . © 1985 Academic Press, Inc.

1. INTRODUCTION

In the classical one-dimensional bin packing problem, we are given a list $L = (x_1,x_2,...,x_n)$ of items, each with a *size* $s(x_i)$ in the interval (0,1], together with a sequence of empty bins $X_1,X_2,...$, each having a capacity of 1. Our goal is to pack the items into a minimum number of bins in such a way that no bin receives items whose total size exceeds 1. This problem has a variety of applications, from traditional stock-cutting problems (Brown, 1971; Gilmore and Gomory, 1961, 1963) to the packing of television commercials into station breaks (Brown, 1971), and it and its variants have been the subject of well over 100 technical papers (see Coffman *et al.*, 1984, for a survey). Since the problem of finding optimal packings is NP-hard (Garey and Johnson, 1979), most researchers on this problem have concentrated on studying *approximation algorithms* for it, i.e., fast heuristics that generate good but not necessarily optimal packings.

The most famous of these is the FIRST FIT DECREASING algorithm (FFD for short), defined as follows. First, we order the items so that $s(x_1) \geqslant s(x_2) \geqslant \cdots \geqslant s(x_n)$. We then proceed to pack the items in order, starting with x_1, which we place in the first bin X_1. In general, item x_i is placed into the first bin that has room for it, i.e., we find the minimum j such that the total size of items currently in X_j is no more than $1-s(x_i)$ and place x_i in X_j.

This algorithm is appealing for both its simplicity and the efficiency with which it can be implemented; with just a little effort one can improve on the obvious $\Theta(n^2)$ implementation and get FFD to run in time $\Theta(n \log n)$

*Presented at the Symposium on Complexity of Approximately Solved Problems, April 17, 1985

0885-064X/85 $3.00

(Johnson, 1973; Johnson *et al.*, 1974). Even more appealing is the performance guarantee satisfied by FFD.

Let FFD(L) denote the number of non-empty bins in the FFD packing of L and let OPT(L) denote the number of bins in an optimal packing. Johnson (1973) showed that, for all lists L,

$$\mathrm{FFD}(L) \leqslant \frac{11}{9}\mathrm{OPT}(L) + 4.$$

This means that FFD is guaranteed never to use a number of bins that exceeds the optimal number by more than approximately 22%. In more technical terms, if we let $R_{\mathrm{FFD}}(n)$ denote the worst possible value of the ratio FFD(L)/OPT(L) when OPT(L) $= n$ and $R_{\mathrm{FFD}}^{\infty}$ denote the "lim sup" of $R_{\mathrm{FFD}}(n)$ as $n \to \infty$ (the "asymptotic worst-case ratio" for FFD), we have $R_{\mathrm{FFD}}^{\infty} \leqslant 11/9$.

This 11/9 bound is "tight," in that for any $n > 0$ there exists a list L such that OPT(L) $> n$ and FFD(L) $=$ (11/9)OPT(L). Hence $R_{\mathrm{FFD}}^{\infty} = 11/9$. Thus to obtain better guarantees we need new algorithms. Over the years a variety of attempts to beat 11/9 have been made, with varying amounts of success. Much of the progress has been theoretical rather than practical. The first such theoretical breakthrough came in 1978 (although it did not reach print until 1980): A. C. Yao (1980) devised an algorithm he called "REFINED FFD" (or RFFD for short) which runs in time $O(n^{10}\log n)$ and provides a guarantee that

$$\mathrm{RFFD}(L) \leqslant \left(\frac{11}{9} - \frac{1}{10{,}000{,}000}\right)\mathrm{OPT}(L) + 8.$$

Shortly thereafter, Fernandez de la Vega and Lueker (1981) showed that, from a theoretical point of view, one could do *much* better than this. For any $\epsilon > 0$, there is a *linear time* algorithm A_ϵ that has an asymptotic worst-case ratio no greater than $1 + \epsilon$. More precisely, the guarantee provided is that $A_\epsilon(L) \leqslant (1+\epsilon)\mathrm{OPT}(L) + O(\epsilon^{-2})$. Unfortunately the running times for these A_ϵ are exponential in $1/\epsilon$. This drawback was eliminated, at a price, by Karmarkar and Karp (1982), who found modified versions of the A_ϵ with running times growing only polynomially with $1/\epsilon$. The "price" was that programming complexity was increased by several orders of magnitude. (The modified algorithms use the ellipsoid method of Khachiyan (1979) and Grötschel *et al.* (1981) as a subroutine, as well as subroutines for finding near-optimal solutions to the NP-hard "knapsack problem.") Carrying their techniques to the limit, Karmarkar and Karp (1982) also devised an algorithm A that guarantees a packing using no more than $\mathrm{OPT}(L) + O(\log^2 \mathrm{OPT}(L))$ bins and hence has $R_A^{\infty} = 1$. However, this algorithm is both complicated to program *and* expensive to use (the best time bound they are able to provide is $O(n^8 \log^3 n)$).

Although the above results are interesting, they are unlikely to have any effect on the practical applications of bin packing. This paper presents an algorithm, which we shall call MODIFIED FIRST FIT DECREASING (MFFD), that improves substantially on the worst-case behavior of FFD without significantly increasing either its running time or its programming complexity. (The asymptotic worst-case ratio for MFFD is $71/60 = 1.18333...$ rather than $11/9 = 1.222...$.) Moreover, MFFD does not fall into the common trap of sacrificing average-case performance in order to obtain improved worst-case bounds. FFD is a very impressive performer "on average" (e.g., see Bentley *et al.*, 1983, 1984), but MFFD should perform equally well. (For instance, the results of Frederickson (1980), Lueker (1982), and Bentley *et al.* (1984) for FFD also apply to MFFD.)

Although MFFD has been the subject of various claims since 1979, when it was first developed in response to Yao's result mentioned above, it has not previously been analyzed in print. The initial claim, reported by Garey and Johnson (1981), was that $71/60 \leqslant R_{\mathrm{MFFD}}^{\infty} \leqslant 6/5 = 1.20$. A year later, in 1980, the upper bound claim was revised downward to match the lower bound, thus yielding the tight result that $R_{\mathrm{MFFD}}^{\infty} = 71/60$. Since that time there have been many citations of the result, but no published proof. This paper presents such a proof.

Before concluding this historical summary, we should briefly mention one other recent bin packing result, first claimed around 1980 by Friesen and Langston (and done independently of our work; see Friesen and Langston, 1984). This result concerns a hybrid algorithm for bin packing that is claimed to provide an asymptotic worst-case ratio no greater than 6/5. Since the upper bound of 6/5 is not known to be tight in this case (the true upper bound might indeed be better than 71/60), this algorithm too may be worth practical consideration, at least in those situations of moderate size where its slower running time of $\Theta(n^2)$ is not a crucial drawback. We shall have a bit more to say about this algorithm in the Conclusion.

The remainder of this paper is organized as follows. In Section 2 we define MFFD and show how it was designed to handle the situations in which FFD behaves most poorly, although the two algorithms share a 71/60 worst-case example when no item size exceeds 1/2. Section 3 provides an overview of the upper bound proof, which occupies most of the remainder of the paper. Section 4 then begins the upper bound proof by defining a weighting function and using it to show that $\mathrm{MFFD}(L) \leqslant (71/60)\mathrm{OPT}(L) + C$ when no item size exceeds 1/2. The two main lemmas from which this follows are improved versions of similar results presented by Johnson (1973) for FFD, and their proofs are included in appendixes. Section 5 then presents the induction argument for our general result, which involves a step-by-step transformation of an optimal packing into the one provided by MFFD and includes some of the intricate case analysis for

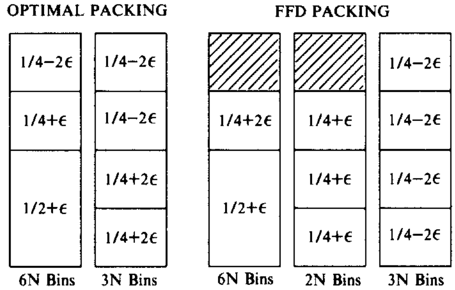

FIG. 1. Lists L for which $\mathrm{FFD}(L) = (11/9)\mathrm{OPT}(L)$.

which the bin packing field is famous. We conclude in Section 6 with some comments about our proof techniques and a further discussion of the practical implications of our results.

2. MODIFIED FIRST FIT DECREASING

If one is trying to modify an algorithm to improve its worst-case behavior, a standard first step is to look at the types of instances that make the original algorithm perform poorly. Fig. 1 shows a schematic of the type of bin packing instance for which FFD uses 11/9 times the optimal number of bins. Note that FFD gets into trouble because it puts one item of size $1/4+2\epsilon$ in a space that could be much better filled by two slightly smaller items. MFFD is designed to avoid this "error" with a minimum of effort. It proceeds as follows.

As with FFD, we first reorder the input list $L = (x_1, x_2, ..., x_n)$ so that $s(x_1) \geqslant s(x_2) \geqslant \cdots \geqslant s(x_n)$. Let us classify the items in L according to their size and also assign to certain items x a numerical type $type(x)$, as specified in Table I. Notice that, when $type(x)$ is defined, it is the denominator of the smallest unit fraction that is at least as large as $s(x)$.

TABLE I
THE ITEM TYPES

$A = \{x : s(x) \in (1/2, 1]\}$	A-items	$type(x) = 1$
$B = \{x : s(x) \in (1/3, 1/2]\}$	B-items	$type(x) = 2$
$C = \{x : s(x) \in (1/4, 1/3]\}$	C-items	$type(x) = 3$
$D = \{x : s(x) \in (1/5, 1/4]\}$	D-items	$type(x) = 4$
$E = \{x : s(x) \in (1/6, 1/5]\}$	E-items	$type(x) = 5$
$F = \{x : s(x) \in (11/71, 1/6]\}$	F-items	$type(x) = 6$
$G = \{x : s(x) \in (0, 11/71]\}$	G-items	

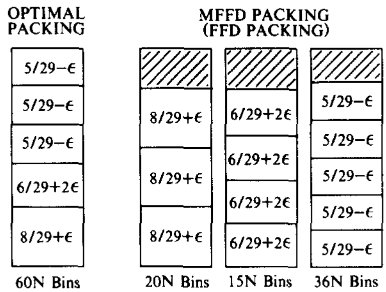

FIG. 2. Lists L with all item sizes in $(0,1/2]$ for which $\mathrm{MFFD}(L) = \mathrm{FFD}(L) = (71/60)\mathrm{OPT}(L)$

We divide the action of MFFD into five distinct phases:

1. Assign the A-items to the first $|A|$ bins in order, so that the levels of the bins form a nonincreasing sequence. (The *level* of a bin is the total size of the items it contains.) Call these bins the *A-bins*.

2. Proceed through the A-bins from left to right (i.e., from bin X_1 through bin $X_{|A|}$), treating the current bin X_i as follows: If any unpacked B-item will fit in X_i, put in the *largest* such B-item that will fit. (Note that there can be room for at most one.)

3. Proceed through the A-bins from right to left (i.e., from bin $X_{|A|}$ through bin X_1), treating the current bin X_i as follows:

   If X_i contains a B-item, do nothing.

   If the two smallest unpacked items from $C \cup D \cup E$ will not fit together in X_i (or if there is only one such item left), do nothing.

   Otherwise, place the smallest unpacked item from $C \cup D \cup E$ in X_i, together with the largest remaining unpacked item from $C \cup D \cup E$ that will fit.

4. Proceed through the A-bins one last time from left to right, treating the current bin X_i as follows: If *any* unpacked item will fit in X_i, put in the largest such item that will fit, repeating until no unpacked item will fit.

5. Use FFD to pack the remaining items in the bins starting with $X_{|A|+1}$.

Observe that MFFD will pack the example of Fig. 1 optimally. Observe also, however, that MFFD is identical to FFD when there are no A-items. Hence it too falls victim to the instances from Johnson (1973) and Johnson *et al.* (1974), depicted in Fig. 2, that show that FFD can still use as many as 71/60 times the optimal number of bins in such situations. Thus we have the following:

THEOREM 1. $R^{\infty}_{\mathrm{MFFD}} \geqslant 71/60$.

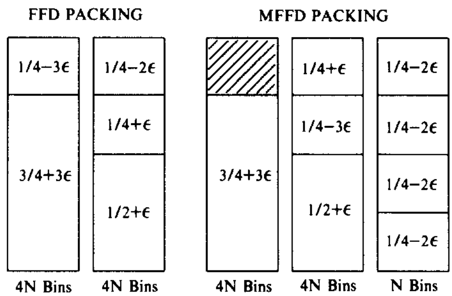

FIG. 3. Lists L for which MFFD(L) = (9/8)FFD(L).

The remainder of this paper will be devoted to proving that this lower bound is tight. Before we conclude this section, however, let us briefly address the question of how FFD and MFFD compare to *each other*, rather than to the optimal packings. Since MFFD packs the examples of Fig. 1 optimally, we conclude that FFD(L)/MFFD(L) can be as large as 11/9 for lists L with arbitrarily large values OPT(L). This bound is tight, given that $R^{\infty}_{\mathrm{FFD}} = 11/9$. In the other direction we do not have a tight bound, but conjecture that the examples pictured in Fig. 3 provide the worst possible values of MFFD(L)/FFD(L), that value being 9/8 = 1.125.

3. OVERVIEW OF THE PROOF

In this section we provide an overview of the proof of the following theorem, which together with the lower bound examples of Fig. 2 will imply that $R^{\infty}_{\mathrm{MFFD}} = 71/60$.

THEOREM 2. *For all lists L,* MFFD(L) $\leqslant$ (71/60)OPT(L) + (31/6).

As a first simplification, let us note the following:

LEMMA 1. *In proving Theorem 2, we can without loss of generality restrict attention to lists L that contain no G-items.*

Proof. We need only show that, if there exists a counterexample to Theorem 2, then there exists such a counterexample containing no G-items. So suppose there exists a counterexample to Theorem 2. If the list L for the counterexample is such that the MFFD packing of L includes no bin started with a G-item, then we are done, because we can delete all G-items from L without changing the number of bins in the MFFD packing (and the optimal number of bins cannot increase), thus obtaining a counterexample with no G-items. On the other hand, if the MFFD packing of L includes a bin that starts with a G-item, then the last bin must start with a G-item, and hence each previous bin must have total contents exceeding

$1-11/71=60/71$. If we denote $\Sigma_{x \in L} s(x)$ by $s(L)$, this means that $s(L) > (60/71)(\text{MFFD}(L)-1)$. Since $\text{OPT}(L) \geqslant s(L)$, this implies that $\text{MFFD}(L) < (71/60)\text{OPT}(L)+1 < (71/60)\text{OPT}(L)+(31/6)$. Thus in this case the list L cannot be a counterexample to Theorem 2, and the Lemma follows. ■

As with many proofs in the field of bin packing, our proof of Theorem 2 will be based on a "weighting function argument." In its simplest form, such an argument defines a weight $W(x)$ for each item x in L, extends this to sets X of items by the rule $W(X)=\Sigma_{x \in X} W(x)$, and then proves two things for all lists L (in our case, with no G-items): (*i*) except for a constant number C of bins, each bin in the MFFD packing contains total weight at least 1 and (*ii*) no optimal bin contains total weight more than 71/60. This would imply, for all lists L with no G-items:

$$\text{MFFD}(L) - C \leqslant W(L) \tag{1}$$

and

$$W(L) \leqslant (71/60)\text{OPT}(L). \tag{2}$$

If $C \leqslant (31/6)$, Theorem 2 would clearly follow.

Unfortunately, and for a variety of reasons, both the weighting function and the argument will need to be more complicated than this. Our weighting function w, which will be described in detail in the next section, will be based on weights for individual items, but will also include *discounts* for certain pairs and triples of items satisfying specified constraints on their relative sizes. The total weight for a list L of items will then be defined to be the sum of the individual item weights minus the maximum discount achievable over all legal partitions of L into 1-, 2-, and 3-item sets. In addition, w will be defined only for lists L consisting of B-, C-, D-, E-, and F-items (i.e., no A-items or G-items).

Given this weighting function, we will begin by proving that (1) and (2) hold for all lists L with no A- or G-items, with $C=5$ in (1). This will provide the basis for an induction argument for the general case. Fortunately, MFFD is identical to FFD on such restricted lists, so techniques developed by Johnson (1973) and Johnson *et al.* (1974) for dealing with FFD can be applied.

To extend this result to the general case, we then proceed as follows. Let S_{MFFD} denote the set of items from L that are placed in the non-A-bins by MFFD. From the restricted result, we know that any (legal) partition π of S_{MFFD} gives a total weight $w(\pi)$ satisfying

$$w(\pi) \geqslant \text{MFFD}(S_{\text{MFFD}}) - 5 = \text{MFFD}(L) - |A| - 5.$$

If we can also show that there exists such a partition π for S_{MFFD} satisfying

$$w(\pi) \leqslant \frac{71}{60}(\text{OPT}(L) - |A|) + \frac{11}{60}|A| + \frac{1}{6}, \qquad (*)$$

then we will have

$$\text{MFFD}(L) - |A| - 5 \leqslant \frac{71}{60}(\text{OPT}(L) - |A|) + \frac{11}{60}|A| + \frac{1}{6},$$

or, rewriting,

$$\text{MFFD}(L) \leqslant \frac{71}{60}\text{OPT}(L) + \frac{31}{6},$$

proving Theorem 2. Hence we concentrate on proving (*).

In order to prove (*), we focus on the A-bins. Each of these contains a single A-item in both the optimal and MFFD packings, so we can identify the A-bins in the two packings that contain the same A-item. The proof consists of a sequence of steps that transforms the optimal packing of the A-bins into the MFFD packing of the A-bins, maintaining certain properties as the transformation proceeds, with (*) following by an induction on the steps in this transformation.

Let P_0 denote the optimal packing of the A-bins, with those bins indexed in the same order as the corresponding A-items occur in L (these are also the indices of the bins in which those items are placed by MFFD). Let S_0 be the sublist of items from L that are not in A-bins in the optimal packing. Let π_0 be a legal partition of the items in S_0 satisfying $w(\pi_0) \leqslant (71/60)(\text{OPT}(L)-|A|)$, which we know exists from our result for the restricted case with no A-items. In addition, let CR_0 be an initial "credit function" that assigns an additional weight of 11/60 to each of the A-bins, corresponding to the extra weight allowed for those bins in (*).

We will think of the MFFD algorithm being applied in place to this packing, with each step of the transformation corresponding to the movement of certain items from where they are in the current packing to where MFFD places them, displacing certain items to new locations (either into other A-bins or into the collection of items not in A-bins) and giving us a new packing P_i of the A-bins and a new set S_i of items not in A-bins. We may also need to alter the credit function and the partition of S_i to new values CR_i and π_i.

Recall that MFFD makes a total of three passes over the A-bins, first from left to right (Phase 2), then from right to left (Phase 3), and finally from left to right again (Phase 4). Whenever some collection of items is placed in an A-bin during its turn in one of these passes, we will say that the recipient A-bin has been processed in that phase. Each such processing step will give rise to a corresponding step in our transformation. We will use P_i, S_i, π_i, and CR_i to denote the transformed packing, set of items not in A-bins, partition of S_i, and credit function after the ith processing step (or, equivalently, after the ith transformation step).

The properties we will maintain (induction hypotheses) are the following:

H1. For $1 \leqslant k \leqslant |A|$, if the A_k-bin was processed during the first i processing steps of MFFD, then in P_i it has precisely the same contents as in the MFFD packing after its first i processing steps, and it has $CR_i(A_k) = 0$.

H2. If no bin has yet been processed under Phase 3 of MFFD, then

$$w(\pi_i) + \sum_j CR_i(A_j) \leqslant \frac{71}{60}(\mathrm{OPT}(L) - |A|) + \frac{11}{60}|A|.$$

Otherwise,

$$w(\pi_i) + \sum_j CR_i(A_j) \leqslant \frac{71}{60}(\mathrm{OPT}(L) - |A|) + \frac{11}{60}|A| + \frac{1}{6}.$$

(As we shall see, the 1/6 in the second part of (H2) is needed to take care of a small technicality.)

Note that (H1) and (H2) both hold for $i = 0$: Initially no bins have been processed, so (H1) holds vacuously. As for (H2), the first case applies and is satisfied because of our choices for π_0 and CR_0.

The desired result (*), and hence Theorem 2, will follow if (H1) and (H2) continue to hold after Phase 4 has been completed, since at that point P_i will agree with the MFFD packing of L on *all* the items in A-bins, so that S_i will equal the sublist S_{MFFD} of (*). Thus all we need show is that if the two hypotheses hold after the ith processing step, then we can arrange that they continue to hold after the $(i+1)$st.

Here is an illustration of what is involved. Only local changes will be made in updating P_i, S_i, π_i, and CR_i. Suppose the A_j-bin is the A-bin processed in the $(i+1)$st processing step and that it receives two new items x and y. These new items came either from A-bins in P_i or from S_i. If either was contained in an A-bin other than the A_j-bin, we call that bin an *accessory* A-bin. The items displaced from the A_j-bin by x and y will be moved either to S_{i+1} or to one or more of the "accessory" A-bins (in which case they may displace other items into S_{i+1}), depending on what we know about their sizes and the gaps in those bins. In a few cases, slightly more complicated shiftings may go on, but in general this will be all that is involved in updating P_i to P_{i+1}. The credit function CR_{i+1} will be the same as CR_i for all but the A_j-bin and the accessory A-bins; typically these will have their CR-values reduced. The partition π_{i+1} will have to differ from π_i, since there will be changes in S_{i+1} induced by the changes in P_{i+1}. Any items from S_i that went into the A_j-bin must be deleted, and any items that were moved from P_i to S_{i+1} must be accommodated. If an item from a tuple in π_i disappears, that tuple must either be split up (and its discount lost) or else modified by having its departed member replaced by a new item that keeps the tuple discountable.

As a result of these alterations, the value of $w(\pi)$ will change by an amount

$$\Delta w = w(\pi_{i+1}) - w(\pi_i).$$

This will always be offset, however, by a corresponding change in

$$\Delta CR = \sum_{k=1}^{|A|} \left[CR_{i+1}(A_k) - CR_i(A_k) \right].$$

More precisely, we will always have $\Delta w + \Delta CR \leqslant 0$, which will be enough to ensure that (H2) is preserved. (We shall use "Δ" as a shorthand for $\Delta w + \Delta CR$.) To verify this claim will require a detailed case analysis, depending on the sizes of the items placed in the processing step, the contents of the A-bin in which they are placed, and the phase of MFFD in which the placement occurs.

4. WEIGHTING FUNCTIONS AND THE CASE OF NO A-ITEMS

In this section we define the weighting function w, as well as two simpler auxiliary weighting functions W and V, and address the case in which there are no A-items.

We will define our three weighting functions in order of increasing complexity. The first weighting function W is defined solely in terms of individual items. Of the two properties given at the beginning of the previous section, it satisfies (1) but violates (2), i.e., although the MFFD bins will contain an average weight that is large enough (roughly 1), the optimal bins can have an average weight that is too large (greater than 71/60). W will be said to define the *base weights* of the items. It is quite straightforward. The weight $W(x)$ is defined simply as $1/type(x)$ (recall that

TABLE II
VALUES OF BASE WEIGHTING FUNCTION W

Item Class	$W(x)$
B	30/60
C	20/60
D	15/60
E	12/60
F	10/60

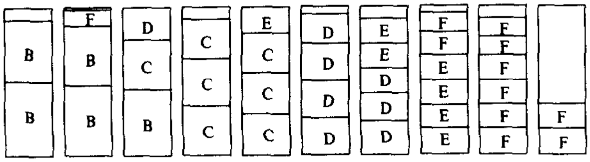

FIG. 4. A packing of a list L with all item sizes in $(0,1/2]$.

type(x) was defined in Section 2 to be the denominator of the smallest unit fraction that equals or exceeds $s(x)$). For future reference Table II shows the values of $W(x)$ as a function of the "alphabetic" class to which x belongs, expressed as fractions over a common denominator of 60.

To see that W satisfies property (1), we need to look in more detail at the structure of an MFFD (FFD) packing P of a list L with all items of size 1/2 or less. Let us call a bin in P a *k-bin* if the first (largest) item x it receives has *type*$(x) = k$. The *regular* items in P are those items whose type matches that of the bin containing them. A *regular* k-bin is a k-bin that receives its full complement of k (regular) k-items, and hence has total W-weight at least $k(1/k) = 1$. See Fig. 4 for a typical packing P.

Note that P consists of a sequence of 2-bins, followed by sequences of 3-, 4-, 5-, and 6-bins in turn, with all but the last bin in each sequence being regular. Thus, if we let *Regular(P)* be the set of all the regular items in P, we have the following.

LEMMA 2. *If P is the MFFD packing of a list L composed of only B-, C-, D-, E-, and F-items, then*

$$W(Regular(P)) \geqslant \mathrm{MFFD}(L) - \sum_{i=2}^{6} \frac{i-1}{i} = \mathrm{MFFD}(L) - 3.55$$

This implies that property (1) holds for W and our restricted class of lists L, even if we ignore all the non-regular or *surplus* items (the items of type $k' > k$ that may fill up the top of a k-bin). If we are to obtain a weighting function that satisfies property (2) in addition to property (1), however, we will need to take advantage of the "slack" these surplus items provide. (Note that W easily yields weights that are too heavy to satisfy property (2). A list L consisting of $2n$ items of size 7/20 and n items of size 3/10 would have $\mathrm{OPT}(L) = n$ but $W(L) = 4n/3 > (71/60)\mathrm{OPT}(L)$.)

Our second weighting function, V, takes advantage of the slack by means of "discounts." Weighting function V by itself will turn out to be enough to satisfy both properties (1) and (2) when all items have size 1/2 or less. However, to cope with the complexities of the A-bins, as we must in the next section, a more complicated, and slightly heavier, version of V will be needed, and this will be the role filled by w.

Discounts are defined for pairs of items that are "discountable." The pair (x,y) is ***discountable*** whenever $type(x) \cdot s(x) + s(y) \leqslant 1$. If (x,y) is discountable, then the discount $D(x,y) = W(y)/type(x)$. The weighting function V is defined both for single elements x, in which case it equals $W(x)$, and for discountable pairs (x,y), in which case it equals $W(x) + W(y) - D(x,y)$. For example, if (x,y) is discountable and x is a C-item, then $V(x,y) = W(x) + (2/3)W(y)$.

As a motivation for this definition, consider a list L that is the disjoint union of k discountable pairs (x,y), where all the x's have the same size and are of type k, and all the y's have the same size. MFFD would start by creating a regular k-bin, but then, because x and y are discountable, it would also put at least one y in the top of that k-bin before starting any new bins. The total weight (under W) of the items in the k-bin would thus be at least $1 + W(y)$. The discounts of the k pairs that contained the x's in the bin are precisely enough to balance out this excess $W(y)$, reducing the effective weight of the bin to 1, which is all we need to satisfy property (1).

The weighting function V is extended to a list L of items as follows: Define a V-legal partition of L to be any partition of L into 1- and 2-item subsets such that each 2-item subset is discountable. Then

$$V(L) = \min\{V(\pi) : \pi \text{ is a } V\text{-legal partition of } L\},$$

where $V(\pi)$ is defined to be

$$\Sigma_{(x)\in\pi} V(x) + \Sigma_{(x,y)\in\pi} V(x,y).$$

Notice that $V(\pi)$ can also be written as $V(\pi) = W(L) - D(\pi)$, where $D(\pi) = \Sigma_{(x,y)\in\pi} D(x,y)$.

We can conclude that V, like W, satisfies property (1) for lists in our restricted class as a corollary of the following general lemma.

LEMMA 3. *For any list L with all item sizes in the interval* $(1/N, 1/K]$, $N > K \geqslant 2$,

$$\mathrm{MFFD}(L) = \mathrm{FFD}(L) \leqslant V(L) + (N-K).$$

A restricted version of this lemma was first proved by Johnson (1973) using a very convoluted (and perhaps somewhat shaky) argument. Our much simpler proof of the more general result is postponed to Appendix 1, so as not to interrupt the flow of the presentation, which has brought us to the point where we can now define our final weighting function w.

Like V, w is a "discounted" version of W. For all singletons x from L, w is defined by $w(x) = W(x) = V(x)$. The function w is also defined for all discountable pairs (x,y) (with the same definition of "discountable"), although here the discount $d(x,y)$ taken by w may be smaller than the

TABLE III
DISCOUNTS USED IN WEIGHTING FUNCTIONS w AND W

Tuple	Comment	d	D	Tuple	d	D
(BBC)	-	9/60	10/60	(CD)	5/60	5/60
(BC)	-	6/60	10/60	(CE)	2/60	4/60
(BD)	-	6/60	7.5/60	(CF)	3/60	~3.3/60
(BE)	$B \leqslant 5/14$	6/60	6/60	(DE)	2/60	3/60
(BE)	$B > 5/14$	4/60	6/60	(DF)	2/60	2.5/60
(BF)	-	5/60	5/60	(EF)	0	2/60

discount $D(x,y)$ taken by V, so that $w(x,y) = W(x)+W(y)-d(x,y) \geqslant V(x,y)$. (Moreover, for one case, $d(x,y)$ depends on the *size* of x, not just its type.) Finally, w is also defined for a limited number of *triples* (x,y,z).

We say a triple (x,y,z) is *discountable* if both x and y are B-items (i.e., type 2), x precedes y on L, z is a C-item, and $s(x)+s(y)+s(z) \leqslant 1$. Note that this implies that both B-items are smaller than $1-1/3-1/4 = 5/12$, and that $2s(y)+s(z) \leqslant s(x)+s(y)+s(z) \leqslant 1$, so that (y,z) is a discountable pair. For such a triple, $d(x,y,z)$ will exceed $d(y,z)$ but will still be less than $D(y,z)$, yielding $w(x,y,z) = W(x)+W(y)+W(z) - d(x,y,z) \geqslant V(x)+V(y,z)$.

The exact definition of w can be derived from Table III, which gives the values of the reduced discount function d. The table also includes values for the original discount function D, so that the reader may readily verify the above claims about the relative weights of V and w. The table introduces some notational conventions that will be used extensively in what follows, but should be self-explanatory. For example, the letter B is used to denote a generic B-item and its size, "(BC)" is used to denote a generic discountable pair of a B- and a C-item, and (BBC) is used to denote the generic discountable triple satisfying (a) above. For later convenience, all fractions are again expressed with denominator 60.

Weighting function w is extended to lists L in a manner analogous to that used for V. Call a partition π of L into disjoint singletons, pairs, and triples *legal* if all pairs and triples are discountable. The value of w on such a partition is defined as

$$w(\pi) = \sum_{(x)\in\pi} w(x) + \sum_{(x,y)\in\pi} w(x,y) + \sum_{(x,y,z)\in\pi} w(x,y,z).$$

The function w is then extended to lists L as was V:

$$w(L) = \min\{w(\pi): \pi \text{ is a legal partition of } L\}.$$

Since a legal partition can be converted to a V-legal one by replacing each triple (x,y,z) by the singleton (x) and the pair (y,z), it is straightforward to derive the following from Table III:

LEMMA 4. *For any list L composed of only B-, C-, D-, E-, and F-items,*

$$V(L) \leqslant w(L) \leqslant W(L).$$

As a corollary, we immediately have that Lemma 3 holds with V replaced by w, and hence property (1) holds for w.

We are now in a position to prove that the weighting function w satisfies property (2). Given the definition of w, the appropriate way to state this claim is as follows:

LEMMA 5. *For any list L composed of only B-, C-, D-, E-, and F-items, there exists a legal partition π of L such that*

$$w(\pi) \leqslant \frac{71}{60}\mathrm{OPT}(L).$$

The proof is relegated to Appendix 2.

Note that as a corollary of Lemmas 1, 3, 4, and 5, we have an alternative proof of the following result by Johnson (1973), now extended to include MFFD.

COROLLARY 5.1. *For any list L with all item sizes in* $(0,1/2]$,

$$\mathrm{MFFD}(L) - \mathrm{FFD}(L) \leqslant \frac{71}{60}\mathrm{OPT}(L) + 5.$$

Proof. By Lemma 1 we can restrict attention to lists containing no A- or G-items, so Lemmas 3 and 4 apply, yielding $\mathrm{MFFD}(L) - \mathrm{FFD}(L) \leqslant V(L)+5 \leqslant w(L)+5$. But Lemma 5 implies that $w(L) \leqslant (71/60)\mathrm{OPT}(L)$, from which the claimed result follows. ∎

5. HANDLING THE ITEMS BIGGER THAN 1/2

In this section we show how to extend the results of Section 4 to the case where items x with $s(x) > 1/2$, i.e., *A-items*, are allowed. We will proceed from the introduction and definitions provided in Section 3, beginning by stating the key objective, inequality (*) from that section, as the lemma we seek to prove.

LEMMA 6. *For any list L composed of only A-, B-, C-, D-, E-, and F-items, if S_{MFFD} is the sublist of L consisting of all items in the non-A-bins of the MFFD packing of L, then there is a legal partition π of S_{MFFD} such that*

$$w(\pi) \leqslant \frac{71}{60}(\mathrm{OPT}(L) - |A|) + \frac{11}{60}|A| + \frac{1}{6}.$$

As indicated in Section 3, Theorem 2 follows directly from Lemma 6.

TABLE IV

POSSIBLE A-BIN CONFIGURATIONS AND CORRESPONDING ALLOWED VALUES FOR $CR(A)$ (ASSUMING THE A-BIN HAS NOT BEEN PROCESSED), EXPRESSED IN MULTIPLES OF 1/60[a]

Bin Type	Restrictions	Allowed Values For $60 \cdot CR(A)$
$[A,E_1,E_2,F]$		11
$[A,E,F_1,F_2]$	*headroom* $< E+F_2+1/6$	11
	headroom $\geqslant E+F_2+1/6$	10,11
$[A,F_1,F_2,F_3]$	*headroom* $< F_2+F_3+1/6$	11
	$F_2+F_3+1/6 \leqslant$ *headroom* $\leqslant F_3+1/3$	10,11
	headroom $> F_3+1/3$	9,10,11
$[A,X,Y]$	Y is not an F-item and hence $X \neq B$	11
$[A,X,F]$	X is not an F-item, *headroom* $< 1/3$	11
	X is not an F-item, *headroom* $\geqslant 1/3$	10,11
$[A,F_1,F_2]$	*headroom* $< F_2+1/6$	10,11
	headroom $\geqslant F_2+1/6$	9,10,11
$[A,B]$		11
$[A,X]$	$X \in C \cup D \cup E$, *headroom* $< 1/3$	11
	$X \in C \cup D \cup E$, *headroom* $\geqslant 1/3$, no $B \leqslant 5/12$ in S_i	1,7,10,11
	$X \in C \cup D$, $1/3 \leqslant$ *headroom* $< 11/30$, a $B \leqslant 5/12$ in S_i	10,11
	$X \in C \cup D$, *headroom* $\geqslant 11/30$, a $B \leqslant 5/12$ in S_i	7,10,11
	$X \in E$, *headroom* $\geqslant 1/3$, a $B \leqslant 5/12$ in S_i	7,10,11
$[A,F]$		0,11
$[A]$		0,11

[a] In each configuration, the items occur in nonincreasing order by size. The term "headroom" is a shorthand for $1-s(A)$, i.e., the space remaining after placement of the A-item. Note that no other A-bin configurations are possible, given that we must restrict ourselves to items of types A, B, C, D, E, and F.

5.1. *Organizing the Case Analysis*

In order to organize the analysis, we first need to know just what the cases are. By convention, the newly processed A-bin will always be referred to as the "A_j-bin." Our cases will be subdivided according to the phase of MFFD during which the processing occurs, the configuration of the A_j-bin (including the value of $CR(A_j)$) at the start of the processing step, and the makeup of the set of items that are placed by MFFD in the A_j-bin during the processing step.

Table IV shows how we will classify the possibilities for the contents of the A_j-bin. In each configuration, the items occur in order of nonincreasing size, and the configurations are listed roughly in the order that they will be considered in the case analysis. We use the term *headroom* to refer to the space left in an A-bin after placement of the corresponding A-item, i.e., $headroom_j = 1 - s(A_j)$. The reader may readily verify that all configurations omitted from the table are impossible due to size constraints.

Also included in the table are certain "allowed" values for $CR_i(A)$ (as multiples of 1/60) for any unprocessed A-bin of the given configuration. Recall that this value is initially 11/60 for all A-bins and goes to 0 once the bin has been processed. An implicit induction hypothesis is that $CR(A)$ never takes on any value other than those listed in the table, and this will help us limit the number of cases that need be considered. We [illegible] ing considered, as in $[A,X,P](11)$.

5.2. *Some Useful Sublemmas*

In this subsection, we introduce some general tools that will be useful when considering the updating of P, S, π, and CR.

Let us first introduce some additional notation. Let "*OLD*" denote the set of non-A-items in the A_j-bin in P_i, and let "*NEW*" denote the set of items placed in the bin by MFFD during the $(i+1)$st processing step. In updating P, one of the things we will be doing is to move all members of $NEW - OLD$ to the A_j-bin. When one of these items x comes from S, its deletion from the updated S can mean a savings in the value of $w(\pi)$, although the benefit will not always equal the full base weight $W(x)$ of x. (There may also be increases in $w(\pi)$ due to the addition of other items to S.)

Let $v(x)$ denote the reduction in $w(\pi)$ caused by deleting x from S and correspondingly altering π as follows:

(1) If x is a singleton in π, delete that singleton.

(2) If x is part of a pair in π, replace the pair by a singleton consisting of the other member of the pair.

(3) If x is part of a triple (B,x,C) or (B,B,x) in π, replace the triple by the two singletons consisting of the two other members of the triple.

(4) If x is part of a triple (x,B,C) in π, replace the triple by the pair (B,C), which is necessarily a discountable pair by the definition of discountable triple.

The following lemma is easily verified by referring to Tables II and III.

LEMMA 6.1. *The values given in Table* V *are lower bounds for* $v(x)$.

TABLE V

LOWER BOUNDS ON SAVINGS $v(x)$ INCURRED WHEN x IS DELETED FROM S

x	Location in π	$v(x) \geqslant$
B	Second Member of (BBC)	21/60
	First Member of (BBC)	27/60
	Member of (BC), (BD), (BE), or (BF)	24/60
	Member of (B)	30/60
C	Member of (BBC)	11/60
	Member of (BC)	14/60
	Member of (CD), (CD), or (CF)	15/60
	Member of (C)	20/60
D	Member of (BD)	9/60
	Member of (CD), (DE), or (DF)	10/60
	Member of (D)	15/60
E	Member of (BE) with $B \leqslant 5/14$	6/60
	Member of (BE) with $B > 5/14$	8/60
	Member of (CE) or (DE)	10/60
	Member of (E)	12/60
F	Member of (BF)	5/60
	Member of (CF)	7/60
	Member of (DF)	8/60
	Member of (EF) or (F)	10/60

In addition to moving all members of $NEW - OLD$ to the A_j-bin, we will also need to find new homes for certain displaced items. These include all members of $OLD - NEW$, as well as items not in NEW that are displaced from accessory bins (we will see later how this happens). Suppose y is such an item and that x is an item from $NEW - OLD$ satisfying $s(x) \geqslant s(y)$. Then the old location of x, which is either in some unprocessed A-bin or in a subset of the partition π, provides a potential home for y. The operation of "replacing x by y" consists of removing x from that old location and putting y in its place, plus some possible adjustments to CR if x was in an A-bin and to π if x was in S. Let $r(x,y)$ denote the reduction in the value of $\Delta = \Delta w + \Delta CR$ caused by such a replacement, i.e., the reduction in Δw caused by modifying the partition block that contained x or the reduction in ΔCR caused by modifying the CR-value for the A-bin that contained x, depending on which was the case. In order to place bounds on $r(x,y)$, as we did with v, it will be necessary first to define the detailed operation of such a replacement.

We divide the specification into eight cases based on the original location of x and the types of x and y. Along with the description, we also verify that $r(x,y) \geqslant 0$ in each case, i.e., that the replacement does not cause an increase in Δ:

1. If x was in the A_k-bin in P, $k \neq j$, move y to that bin. It will fit since x left and $s(y) \leqslant s(x)$. Its presence, however, may change the configuration of the A_k-bin, thus perhaps rendering the current value of $CR(A_k)$ illegal. However, it is easy to verify from Table IV that replacing a non-A-item in a configuration by a smaller item cannot cause the minimum allowed value of $CR(A)$ to increase. Thus we can set the new value of $CR(A_k)$ to the minimum allowed value and guarantee that $r(x,y) \geqslant 0$.

2. If x was a singleton in π, replace that singleton with the singleton (y). Since $s(y) \leqslant s(x)$ we thus have $w(y) = W(y) \leqslant W(x) = w(x)$, so $r(x,y) \geqslant 0$.

3. If x was a member of a pair (z,x) in π, replace this pair by (z,y). The new pair will still be discountable since $s(y) \leqslant s(x)$. It is easy to verify from Table II and the definition of w in terms of discounts that $w(z,y) \leqslant w(z,x)$. (If x and y are of the same type, the two values are the same, and if y is of smaller type, then $w(z,y)$ is *strictly* less than $w(z,x)$, except for the equality when (z,x) is type (C,D) and (z,y) is type (C,E).) It follows that $r(x,y) \geqslant 0$.

4. If x was a member of a pair (x,z) in π and y is of the same type as x, replace (x,z) by (y,z), which will still be discountable and either will have the same value under w or, if (x,z) has type (B,E) with $s(x) > 5/14$ and $s(y) \leqslant 5/14$, will decrease in value. Thus in this case $r(x,y) \geqslant 0$.

5. If x was a member of a pair (x,z) in π and y is of smaller type than x, replace the pair (x,z) by the two singletons (y) and (z). We then have $r(x,y) = w(x,z) - W(y) - W(z) = (W(x) - W(y)) - d(x,z) \geqslant (W(x) - W(y)) - D(x,z)$. Let $type(x) = k$, which implies that $W(x) = 1/k$ by the definition of W. Then we must have $type(z) \geqslant k+1$, and so $D(x,z) \leqslant (1/k)(1/(k+1))$ by the definition of D. On the other hand, $type(y) \geqslant k+1$, so that $W(y) \leqslant 1/(k+1)$, and $W(x) - W(y) \geqslant (1/k)(1/(k+1))$. Hence $r(x,y) \geqslant 0$.

6. If x was a member of a triple (x,u,z), (u,x,z), or (u,z,x) in π and y is of the same type, then we substitute y for x in that triple. The new triple will still be discountable and will have the same value under w, so $r(x,y) = 0$.

7. If x was a member of a triple (x,u,z) or (u,x,z) in π (in which case x is a B-item) and y is of smaller type (and hence no bigger than a C-item), replace the triple by the three singletons (y), (u), and (z). The discount lost is 9/60, but $W(x) - W(y) \geqslant 10/60$, so $r(x,y) > 0$.

8. If x was a member of a triple (u,z,x) in π (in which case x is C-item) and y is of smaller type (and hence no bigger than a D-item), replace (u,z,x) by (u) and (z,y). (The latter pair is discountable since (z,x) was, by the definition of discountable triple.) In this case $r(x,y) = (W(x)-W(y))+d(z,y)-9/60 \geqslant 5/60+4/60-9/60 = 0$.

This completes the specification of the replacement operation.

From the above, we have:

LEMMA 6.2. *If $x \in (NEW-OLD)$ and $s(y) \leqslant s(x)$, then if we replace x by y according to the above rules, it will always be the case that $r(x,y) \geqslant 0$.*

If y is an F-item and x is of larger type, stronger conclusions can be drawn.

LEMMA 6.3. *If $x \in (NEW-OLD)$ is a B-, C-, D-, or E-item, y is an F-item, and we replace x by y according to the above rules, then we will have the following.*

(1) $r(x,y) \geqslant 1/60$.

(2) $r(x,y) \geqslant 2/60$ if $x \in S$ and x is not the E in a (BE) pair of π with $B \leqslant 5/14$.

(3) $r(x,y) \geqslant 3/60$ if $x \in S$ and x is in a tuple of π, but is not the E in a (D,E) pair, in an (E,F) pair, or in a (BE) pair with $B \leqslant 5/14$.

Proof. Suppose x and y are as specified by the lemma's main hypothesis. We show that the conclusions hold no matter which replacement rule applies.

If Rule 1 applies, then x came from an A-bin, say the A_k-bin, and only Case 1 of the lemma applies. We first note that the A_k-bin cannot yet have been processed, as (H1) implies that no item in a processed bin ever gets moved again. Hence $CR(A_k)$ must have a value as specified in Table IV. We divide into six cases depending on the configuration of the A_k-bin. (We use primed symbols to distinguish items in the new configuration from those in the old that have the same name.)

$[A,E_1,E_2,F](c) \rightarrow [A,E,F_1,F_2](c')$. By Table IV, we must have $c = 11$. Moreover, *headroom* $\geqslant E_1+E_2+F > E+1/6+F_2$. The minimum allowed value for c' in this case is 10, so $r(x,y) = (11-10)/60 = 1/60$.

$[A,E,F_1,F_2](c) \rightarrow [A,F_1',F_2',F_3](c')$. By Table IV, c must equal 10 or 11. If $c = 10$, then *headroom* $\geqslant E+F_2+1/6 > F_3+1/3$, so $c' = 9$ and $r(x,y) = 1/60$. Otherwise, *headroom* $\geqslant E+F_1+F_2 > F_2'+F_3'+1/6$, so $c' \leqslant 10$ and $r(x,y) \geqslant 1/60$.

$[A,X,Y](c) \to [A,X',F](c')$, where Y is not an F-item. By Table IV, $c = 11$. Moreover, ***headroom*** $> 2/6 = 1/3$, and hence $c' = 10$. Thus $r(x,y) = 1/60$.

$[A,X,F](c) \to [A,F_1,F_2](c')$. By Table IV, c is either 10 or 11. If $c = 10$, ***headroom*** $\geqslant 1/3 \geqslant F_2 + 1/6$, so $c' = 9$ and $r(x,y) = 1/60$. Otherwise, $c' \leqslant 10$ and $r(x,y) \geqslant 1/60$.

$[A,B](c) \to [A,F](c')$. By Table IV, $c = 11$ and $c' = 0$, so $r(x,y) = 11/60 > 1/60$.

$[A,X](c) \to [A,F](c')$, where X is not a B-item. By Table IV, $c \geqslant 1$ and $c' = 0$, so $r(x,y) \geqslant 1/60$.

This exhausts the cases for Rule 1. For the remainder of the rules, we must consider which of the three cases of the lemma applies, and verify the appropriate upper bound, be it 1/60 (Case 1), 2/60 (Case 2), or 3/60 (Case 3).

If Rule 2 applies, then x is a singleton in π, $r(x,y) = W(x) - W(y)$, and the relevant case is Case 2. Since x must be at least as large as type E, we have $r(x,y) \geqslant (12/60 - 10/60) = 2/60$, as required.

If Rule 3 applies, then x was the second member of a pair (z,x) in π, which is replaced by (z,y). Once again we must subdivide the cases.

$(B,C) \to (B,F)$. Case 3 applies and $r(x,y) = W(C) - W(F) - (d(B,C) - d(B,F)) = 10/60 - (1/60) = 9/60 \geqslant 3/60$, as required.

$(B,D) \to (B,F)$. Case 3 applies and $r(x,y) = W(D) - W(F) - (d(B,D) - d(B,F)) = 5/60 - (1/60) = 4/60 \geqslant 3/60$, as required.

$(B,E) \to (B,F)$ with $B \leqslant 5/14$. Case 1 applies and $r(x,y) = W(E) - W(F) - (d(B,E) - d(B,F)) = 2/60 - (1/60) \geqslant 1/60$, as required.

$(B,E) \to (B,F)$ with $B > 5/14$. Case 3 applies and $r(x,y) = W(E) - W(F) - (d(B,E) - d(B,F)) = 2/60 - (-1/60) \geqslant 3/60$, as required.

$(C,D) \to (C,F)$. Case 3 applies and $r(x,y) = W(D) - W(F) - (d(C,D) - d(C,F)) = 5/60 - (2/60) \geqslant 3/60$, as required.

$(C,E) \to (C,F)$. Case 3 applies and $r(x,y) = W(E) - W(F) - (d(C,E) - d(C,F)) = 2/60 - (-1/60) \geqslant 3/60$, as required.

$(D,E) \to (D,F)$. Case 2 applies and $r(x,y) = W(E) - W(F) - (d(D,E) - d(D,F)) = 2/60 - (0) \geqslant 2/60$, as required.

This exhausts the cases for Rule 3.

Rule 4 cannot apply since x and y are not of the same type.

If Rule 5 applies, then x was the first member of a pair (x,z) in π, which is replaced by the singletons (y) and (z). Once again we must subdivide the cases.

$(B,X) \rightarrow (F),(X)$, for $X \in \{C,D,E,F\}$. Case 3 applies and $r(x,y) = W(B) - W(F) - d(B,X) \geqslant 20/60 - 6/60 = 14/60 \geqslant 3/60$ as required.

$(C,X) \rightarrow (F),(X)$, for $X \in \{D,E,F\}$. Case 3 applies and $r(x,y) = W(C) - W(F) - d(C,X) \geqslant 10/60 - 5/60 = 5/60 \geqslant 3/60$, as required.

$(D,X) \rightarrow (F),(X)$, for $X \in \{E,F\}$. Case 3 applies and $r(x,y) = W(D) - W(F) - d(D,X) \geqslant 5/60 - 2/60 \geqslant 3/60$, as required.

$(E,F) \rightarrow (F),(F)$. Case 2 applies and $r(x,y) = W(E) - W(F) - d(E,F) \geqslant 2/60 - 0 \geqslant 2/60$, as required.

This exhausts the cases for Rule 5.

Rule 6 does not apply because x and y are of different types.

If Rule 7 applies, then a (B,B,C) triple is replaced by singletons (B), (F), and (C), Case 3 applies, and $r(x,y) = W(B) - W(F) - d(B,B,C) = 20/60 - 9/60 \geqslant 3/60$, as required.

Finally, if Rule 8 applies, then a (B,B,C) triple is replaced by tuples (B) and (B,F), Case 3 applies, and $r(x,y) = W(C) - W(F) - (d(B,B,C) - d(B,F)) = 10/60 - 4/60 \geqslant 3/60$, as required.

Thus the lemma is satisfied in all cases. ■

5.3. *Case Analysis: Phase* 2 *Updates*

We now have the machinery in hand to begin proving that the induction hypotheses (H1) and (H2) can be preserved as MFFD processes the A-bins. The various tables presented earlier, as well as Lemmas 6.1, 6.2, and 6.3, will be used extensively and should be referred to often. We begin in this subsection with those processing steps that occur during Phase 2 of the algorithm.

Recall that in Phase 2 we proceed from left to right through the A-bins, testing each in turn to see if any as-yet-unpacked B-item will fit, and if so, putting in the largest B-item that does fit. A *processing step* here thus consists of dislodging the set OLD of non-A-items in the current bin (by convention the A_j-bin) by a set NEW consisting of a single B-item, which we shall denote by B'. We now show how to update P, S, π, and CR to take this into account. For this phase we shall be able to accomplish the updates without changing the initial $CR(A) = 11/60$ value for any unprocessed bin, either explicitly or via a replacement. (The only replacements we will be making here will be of B-items by B-items, which, by the replacement rules, do not affect CR.) However, we *will* make use of the fact that the bin being processed has its CR-value reduced from 11/60 to 0 in computing the effect of the updates on Δ.

We divide into cases according to the makeup of OLD. In each case we must show that we can find new homes for the elements of $OLD - NEW$ without making Δ increase. As a notational convenience, throughout the remainder of the proof we will identify the name of an item with its size, e.g., saying $A_j > 1/2$ rather than $s(A_j) > 1/2$, whenever that will cause no confusion.

2.1. $[A_j, B](11) \rightarrow [A_j, B']$. By (H1), B is available when MFFD chooses to put B' into the A_j-bin. Since B clearly fits, this means that $B' \geqslant B$. Thus either B and B' are the same item or we can replace B' by B. By Lemma 6.2 this cannot increase Δ. Hence, with the reduction of 11/70 in $CR(A_j)$, we have $\Delta \leqslant -11/60 \leqslant 0$, as required.

2.2. $[A_j, B, F](11) \rightarrow [A_j, B']$. By the same argument as used above, either B and B' are the same item or we can replace B' by B, incurring no increase in Δ. We then send F to S, yielding $\Delta w = 10/60$, balanced by the fact that, once again, $\Delta CR = -11/60$, so that $\Delta = -1/60 \leqslant 0$.

2.3. $[A_j, OLD](11) \rightarrow [A_j, B']$, *where* $OLD = [C,D]$, $[C,E]$, $[E,E,F]$, $[E,F,F]$, *or* $[F,F,F]$, *and hence headroom* $\geqslant 5/12$. We divide into subcases depending on the location of B'.

2.3.1. *B' is from an $[A_k, B'](11)$- or an $[A_k, B', F](11)$-bin.* The A_k-bin must be to the right of the A_j-bin. (If it had been to the left, it would already have been processed, given that B' fits in it.) Hence $1 - A_k \geqslant 1 - A_j$, and we can move all the items from OLD to the A_k bin, after first sending the F in that bin (if there is one) to S. Thus $\Delta \leqslant 10/60 - 11/60 < 0$.

2.3.2. *B' is in a singleton in π.* Send all of OLD to S. Then $\Delta w \leqslant W(OLD) - v(B') \leqslant 35/60 - 30/60 = 5/60$, and so $\Delta \leqslant 5/60 - 11/60 < 0$.

2.3.3. *B' is in a pair or triple in π.* B' cannot be the second member of a (B, B', C) triple, since in that case the first member B should have gone in the A_j-bin. (It fits because its size is less than $1 - 1/3 - 1/4 = 5/12$, and it is larger than B' (or precedes it on L) by our conventions for describing triples.) Thus by Lemma 6.1, $v(B') \geqslant 24/60$, so that $\Delta w \leqslant 35/60 - 24/60 = 11/60$ and $\Delta \leqslant 11/60 - 11/60 = 0$.

2.4. $[A_j, OLD](11) \rightarrow [A_j, B']$, *where OLD is not any of the above.* In this case one can check in Table IV that $W(OLD) \leqslant 30/60$. If B' is in an A-bin, or is in a singleton of π, proceed as in (2.3). If B' is in a pair or triple of π, we must still have $v(B') \geqslant 21/60$ by Lemma 6.1, so sending all of OLD to S, we have $\Delta w \leqslant 30/60 - 21/60 = 9/60$ and $\Delta \leqslant 9/60 - 11/60 < 0$.

Thus the possibilities for Phase 2 of MFFD have been completed.

5.4. *Case Analysis: Phase* 3 *Updates*

Recall that in Phase 3 we proceed from right to left through those A-bins that did not receive B-items during Phase 2, testing each in turn to see if the two smallest as-yet-unpacked items from the set $C \cup D \cup E$ will fit. If so, we put the smallest such item in, together with the largest remaining item that will fit. The set NEW thus consists of two items, which we shall denote by x and x', where x is the larger of the two (i.e., the first packed). Here are the cases:

3.1. $[A_j,E_1,E_2,F](c) \rightarrow [A_j,x,x']$. By Table IV, $c=11$. By the operation of MFFD, we must have $x' \leqslant E_2$ and $x \geqslant E_1$. If x and E_1 are not the same item, replace x by E_1 at no cost (by Lemma 6.2). Replace x' by F at a savings of at least 1/60 (by Lemma 6.3) and send E_2 to S. $\Delta \leqslant 12/60-1/60-11/60=0$.

3.2. $[A_j,X,F_1,F_2](c) \rightarrow [A_j,x,x']$, *where* $X=E$ *or* F. By Table IV, $c \geqslant 9$. Since $2x' \leqslant x'+x<1/2$, we have $x'<1/4<F_1+F_2$, and so $x \geqslant X$. Replace x by X (assuming they are distinct items), replace x' by F_1, and send F_2 to S. $\Delta \leqslant 10/60-1/60-9/60=0$.

3.3. $[A_j,X,Y](c) \rightarrow [A_j,x,x']$, *where* $Y=D$ *or* E, $X \neq B$. By Table IV, $c=11$. Moreover, we must have $x' \leqslant Y$ and $x \geqslant X$. If x and X are not the same item, replace x by X at no cost. For the treatment of Y, we divide into seven subcases, depending on the original location of x'. We assume that the replacement of x by X has already taken place. This allows us to avoid special treatment for the case where x and x' came from the same accessory A-bin, since X, like any other item in such a bin, is free to be moved again.

3.3.1. *Item* $x' \in NEW \cap OLD$. In this case, $x'=Y$ and no further changes are necessary. $\Delta \leqslant -11/60<0$.

3.3.2. *Item* x' *came from* S. Send Y to S. By Lemma 6.1, $\Delta w \leqslant 15/60-6/60$ and so $\Delta \leqslant 9/60-11/60<0$.

3.3.3. *Item* x' *was one of the two E-items in an* $[A_k,E_1,E_2,F](c')$-*bin*. By Table IV, $c'=11$. Since $E_1+E_2>1/3 \geqslant Y$, we can revise the A_k-bin to $[A_k,Y,F](10)$ and send the other E to S. $\Delta \leqslant 12/60-1/60-11/60=0$.

3.3.4. *Item* x' *was an E-item in an* $[A_k,E,F_1,F_2](c')$-*bin*. By Table IV, $c' \geqslant 10$. In addition, ***headroom***$_k>1/3$. Thus, since $x'+F_1>1/4 \geqslant Y$, we can revise the A_k-bin to $[A_k,Y,F_2](10)$ and send F_1 to S. $\Delta \leqslant 10/60-11/60<0$.

3.3.5. *Item* x' *was a non-A-item in an* $[A_k,X',Y'](c')$-*bin where* $Y'=D$ *or* E. By Table IV, $c'=11$. In addition, ***headroom***$_k>1/3$. Let $z \neq x'$ be the other non-A-item. If $z=E$, revise the A_k-bin to $[A_k,z](7)$ and send Y to S, yielding $\Delta \leqslant 15/60-4/60-11/60=0$. Otherwise, $z=D$ or C and ***headroom***$_k>1/5+1/6=11/30$, so we can once again revise the A_k-bin to $[A_k,z](7)$ and send Y to S, yielding $\Delta \leqslant 0$ in the same way.

3.3.6. *Item x' came from an $[A_k,x',F](c')$-bin.* By Table IV, $c' \geqslant 10$. Revise the A_k-bin to $[A_k,F](0)$ and send Y to S. $\Delta \leqslant 15/60-10/60-11/60 < 0$.

3.3.7. *Item x' came from an $[A_k,x'](c')$-bin.* If $c' < 11$, then we must have $c' \geqslant 1$ and *headroom*$_k \geqslant 1/3$, and so can revise the A_k-bin to $[A_k,Y](11)$, yielding $\Delta \leqslant 10/60-11/60 < 0$. If $c' = 11$, then we can revise the A_k-bin to $[A_k](0)$ and send Y to S, yielding $\Delta \leqslant 15/60-11/60-11/60 < 0$.

3.4. *$[A_j,X,F](c) \to [A_j,x,x']$, where $X = C$, D, E, or F.* By Table IV, $c \geqslant 9$. If X is an F-item, replace x by X and x' by F, yielding $\Delta \leqslant -1/60-1/60-9/60 < 0$. So we may assume that X is a C- D- or E-item, in which case $c \geqslant 10$. If X is either x or x', replace the other one of the two by F, yielding $\Delta \leqslant -1/60-10/60 < 0$. If X is neither x nor x', we consider six subcases based on the previous location of x'.

3.4.1. *Item x' came from S.* If $v(x') \geqslant 10/60$ replace x by F and send X to S, yielding $\Delta \leqslant 20/60-10/60-1/60-10/60 < 0$. Otherwise $6/60 \leqslant v(x') < 10/60$ and x' must be the second element in a (B,x') pair in π, for which we must have $B < (1-1/6)/2 = 5/12$, so that all gaps for unprocessed A-bins must be less than $5/12$ (else B would have been placed in an A-bin during Phase 2). In particular, there can be no unprocessed A-bins of the form $[A_k,E_1,E_2,F](c')$ or $[A_k,E,F_1,F_2](c')$. Furthermore $X < 5/12-1/7 = 23/84 < 2/7$. We subdivide our case into five further subcases, depending on the origin of x. In each of them we will replace x' by F for a savings of at least $1/60$.

3.4.1a. *Item x came from S.* If $v(x) \geqslant 10/60$, send X to S, yielding $\Delta \leqslant 20/60-1/60-10/60-10/60 < 0$. Otherwise $v(x) < 10/60$ and x must be in a pair (B',x) in π. If X is a D- or E-item, replace x' by F and send X to S, yielding $\Delta \leqslant 15/60-6/60-1/60-10/60 < 0$. Otherwise, X is a C-item, so that *headroom*$_j > 1/4+1/7 = 11/28 > 5/14$ and B, $B' > 5/14$. This implies that $v(x) \geqslant 8/60$ and, by Lemma 6.3, that $r(x',F) \geqslant 3/60$, so send X to S, yielding $\Delta \leqslant 20/60 - 8/60 - 3/60 - 10/60 < 0$.

3.4.1b. *Item x is a non-A-item in an $[A_k,X',Y'](c')$-bin where $Y' = D$ or E.* Let $z \neq x$ be the other non-A-item. By Table IV, $c' = 11$. Since *headroom*$_k < 5/12$, neither x nor z can be a C-item. If either is a D-item, then *headroom*$_k > 1/6+1/5 = 11/30 > 5/14$, so S contains no B-items with $B \leqslant 5/14$, and $r(x',F) \geqslant 3/60$. Revise the A_k-bin to $[A_k,X](7)$ and send z to S, yielding $\Delta \leqslant 15/60-4/60-3/60-10/60 < 0$. Otherwise, both x and z are E-items, and since *headroom*$_k > 1/6+1/6 = 1/3$, we can revise the A_k-bin to $[A_k,X](10)$ and send z to S, yielding $\Delta \leqslant 12/60-1/60-1/60-10/60 = 0$.

3.4.1c. *Item x came from an $[A_k,x,F'](c')$-bin.* By Table IV, $c' \geqslant 10$. Moreover, *headroom*$_k > 1/6+1/7 = 13/42 > 2/7$, so we can revise the A_k-bin to $[A_k,X](11)$ and send F' to S, yielding $\Delta \leqslant 10/60+1/60-1/60-10/60 = 0$.

3.4.1d. *Item x came from an $[A_k,x](c')$-bin.* If *headroom*$_k < 1/3$, then $c' = 11$ by Table IV. Revise the A_k-bin to $[A_k](0)$ and send X to S, yielding $\Delta \leqslant 20/60-1/60-11/60-10/60 < 0$. Otherwise, *headroom*$_k \geqslant 1/3$ and, since there is a $B \leqslant 5/12$ in S, $c' \geqslant 7/60$ by Table IV. Revise the A_k-bin to $[A_k,X]$ (11), yielding $\Delta \leqslant 4/60 - 1/60 - 10/60 < 0$.

3.4.2. *Item x' was an E-item in an $[A_k,x',E,F'](c')$-bin.* By Table IV, $c' = 11$. Note that the A_k-bin must be to the left of the A_j-bin, because if it were to the right it would have already been processed, given that it has room for the two non-F-items x' and E. Thus *headroom*$_j \geqslant$ *headroom*$_k > x'+E$, so that we must have $x \geqslant E$ by the operation of Phase 3. Replace x by E (unless x is E), revise the A_k-bin to $[A_k,X,F'](11)$, and send F to S, yielding $\Delta \leqslant 10/60-10/60 = 0$.

3.4.3. *Item x' was an E-item in an $[A_k,x',F_1,F_2](c')$-bin.* By Table IV, $c' \geqslant 10$. Since *headroom*$_k > 3/7 > 5/12$, there can be no $B \leqslant 5/12$ in S. Hence we can revise the A_k-bin to $[A_k,X](1)$, replace x by F, and send F_1 and F_2 to S, yielding $\Delta \leqslant 20/60-1/60-9/60-10/60 = 0$.

3.4.4. *Item x' came from an $[A_k,X',Y'](c')$-bin where $Y' = D$ or E.* By Table IV, $c' = 11$. Let $z \neq x'$ be the other non-A-item in the bin. Since *headroom*$_j \geqslant$ *headroom*$_k \geqslant x'+z$, we must have $x \geqslant z$ by the operation of Phase 3. Replace x by z (unless x is z), revise the A_k-bin to $[A_k,X](11)$, and send F to S, yielding $\Delta \leqslant 10/60-10/60 = 0$.

3.4.5. *Item x' came from an $[A_k,x',F'](c')$-bin.* By Table IV, $c' \geqslant 10$. Replace x by F, revise the A_k-bin to $[A_k,F'](0)$, and send X to S, yielding $\Delta \leqslant 20/60-1/60-10/60-10/60 < 0$.

3.4.6. *Item x' came from an $[A_k,x'](c')$-bin.* If $c' = 11$, replace x by F, revise the A_k-bin to $[A_k](0)$, and send X to S, yielding $\Delta \leqslant 20/60-1/60-11/60-10/60 < 0$. Otherwise, by Table IV we must have $c' \geqslant 1$ and *headroom*$_k \geqslant 1/3$. Replace x by F and revise the A_k-bin to $[A_k,X](10)$, yielding $\Delta \leqslant 9/60-1/60-10/60 < 0$.

3.5. $[A_j,X](c) \to [A_j,x,x']$. If X is either x or x', then *OLD* $-$ *NEW* is empty and there is nothing to do; hence $\Delta \leqslant -c/60 < 0$. If X is an F-item, replace x by X, yielding $\Delta \leqslant -c/60-1/60 < 0$. Thus we may assume that X is a C-, D-, or E-item and is neither x nor x'. We divide into two cases, depending on whether there is a $B \leqslant 5/12$ in S.

3.5A. *There is no $B \leqslant 5/12$ in S, and hence no (BC), (BD), or (BE) pairs in π.* The best we can say about c in this case is that $c \geqslant 1$. We divide into subcases depending on the original location of x'.

3.5A.1. *Item x' came from S.* Since there are no (B,x') pairs in π, we must have $v(x') \geqslant 10/60$ by Lemma 6.1. This, together with the savings of 1/60 in $\Sigma CR(A)$ gives us a total bankroll of 11/60 to help underwrite the cost of finding a new home for X. We subdivide our case further depending on the original location of x, showing that such a bankroll is sufficient. Note that the analysis will not depend on the fact that x' came from S, and so can be reused for other subcases of 3.5.1, so long as a bankroll of at least 11/60 is provided.

3.5A.1a. *Item x came from S.* We must also have $v(x) \geqslant 10/60$, so send X to S, yielding $\Delta \leqslant 20/60 - 10/60 - 11/60 < 0$.

3.5A.1b. *Item x was an E-item in an $[A_k,x,E,F](c')$-bin.* By Table IV, $c' = 11$, so revise the A_k-bin to $[A_k,X,F](10)$ and send E to S, yielding $\Delta \leqslant 12/60 - 1/60 - 11/60 = 0$.

3.5A.1c. *Item x was an E-item in an $[A_k,x,F_1,F_2](c')$-bin.* By Table IV, $c' \geqslant 10$. Since there is no $B \geqslant 5/12$ in S, we can revise the A_k-bin to $[A_k,X](1)$ and send F_1 and F_2 to S, yielding $\Delta \leqslant 20/60 - 9/60 - 11/60 = 0$.

3.5A.1d. *Item x came from an $[A_k,X',Y'](c')$-bin, where Y' is a D- or E-item.* By Table IV, $c' = 11$. Let z be the other non-A-item in the A_k-bin. Again we can revise the A_k-bin to $[A_k,X](1)$, now sending z to S and yielding $20/60 - 10/60 - 11/60 < 0$.

3.5A.1e. *Item x came from an $[A_k,x,F](c')$-bin.* By Table IV, $c' \geqslant 10$. Revise the A_k-bin to $[A_k,F](0)$ and send X to S, yielding $\Delta \leqslant 20/60 - 10/60 - 11/60 < 0$.

3.5A.1f. *Item x came from an $[A_k,x](c')$-bin.* By Table IV, $c' \geqslant 1$. If $headroom_k \geqslant 1/3$, revise the A_k-bin to $[A_k,X](11)$, yielding $\Delta \leqslant 10/60 - 11/60 < 0$. Otherwise $headroom_k < 1/3$ and $c' = 11$ by Table IV. Revise the A_k-bin to $[A_k](0)$ and send X to S, yielding $\Delta \leqslant 20/60 - 11/60 - 11/60 < 0$.

3.5A.2. *Item x' was an E-item in an $[A_k,x',E,F](c')$-bin.* By Table IV, $c' = 11$. As before we must have $x \geqslant E$ because $headroom_j \geqslant headroom_k$. Replace x by E (unless x is E), revise the A_k-bin to $[A_k,X,F](10)$, yielding $\Delta \leqslant -1/60 - 1/60 < 0$.

3.5A.3. *Item x' was an E-item in an $[A_k,x',F_1,F_2](c')$-bin.* By Table IV, $c' \geqslant 10$. Replace x by F_1, revise the A_k-bin to $[A_k,X](1)$, and send F_2 to S, yielding $\Delta \leqslant 10/60 - 9/60 - 1/60 - 1/60 < 0$.

3.5A.4. *Item x' came from an $[A_k,X',Y'](c')$-bin where Y' is a D- or E-item.* By Table IV, $c' = 11$. Let z be the other non-A-item in the A_k-bin. Again we must have $x \geqslant z$, so replace x by z (unless x is z) and revise the A_k-bin to $[A_k,X](11)$ yielding $\Delta \leqslant -1/60 < 0$.

3.5A.5. *Item x' came from an $[A_k,x',F](c')$-bin or an $[A_k,x'](c')$-bin.* By Table IV, $c' \geqslant 1/60$. If $headroom_k \geqslant 1/3$, revise the A_k-bin to $[A_k,X](1)$, replacing x by F in the first case, yielding $\Delta \leqslant -1/60 < 0$. Otherwise, $headroom_k < 1/3$ and $c' = 11$. Revise the A_k-bin to $[A_k,F](0)$ or $[A_k](0)$, respectively, for a local savings of 11/60 in ΔCR to be added to the savings of at least 1/60 for the A_j-bin. This gives a bankroll of 12/60 to underwrite the finding of a new home for X, and by the analysis in Case 3.5A.1 above, we know that this is more than sufficient.

3.5B. *There is a $B \leqslant 5/12$ in S.* In this case there can be no unprocessed $[A_k,E_1,E_2,F](c')$- or $[A_k,E,F_1,F_2](c')$-bins. In addition, by Table IV, $c \geqslant 7$. We divide into subcases depending on the value of c and the type of X.

3.5B.1. *$c = 7$ and X is an E-item.* We divide into subcases depending on the original location of x'.

3.5B.1a. *Item x' came from S.* By Lemma 6.1, $v(x') \geqslant 6/60$. Send X to S, yielding $\Delta \leqslant 12/60 - 6/60 - 7/60 < 0$.

3.5B.1b. *Item x' came from an $[A_k,X',Y'](c')$-bin where Y' is a D- or E-item.* By Table IV, $c' = 11$. Let z be the other non-A-item in the A_k-bin. Again we must have $x \geqslant z$, so replace x by z (unless x is z) and revise the A_k-bin to $[A_k,X](11)$ yielding $\Delta \leqslant -7/60 < 0$.

3.5B.1c. *Item x' came from an $[A_k,x',F](c')$- or an $[A_k,x'](c')$-bin.* By Table IV, $c' \geqslant 7$. Revise the A_k-bin to $[A_k,F](0)$ or $[A_k](0)$, respectively, and send X to S, yielding $\Delta \leqslant 12/60 - 7/60 - 7/60 < 0$.

3.5B.2. *$c = 7$ and X is a C- or D-item.* In this case $headroom_j \geqslant 11/30$ by Table IV, and since $11/30 > 5/14$, this means that there is no $B \leqslant 5/14$ in S and hence $v(z) \geqslant 8/60$ for all D- or E-items z. We divide into subcases depending on the original location of x'.

3.5B.2a. *Item x' came from S or from an $[A_k,x',F](c')$- or $[A_k](c')$-bin.* In the latter two cases we must have $c' \geqslant 7$ and so in all these three cases we can obtain a savings of 7/60 from x', either via $v(x')$, or via revising the A_k-bin to $[A_k,F](0)$ or $[A_k](0)$, respectively. This, together with the savings in ΔCR for the A_j-bin, gives us an aggregate bankroll of 14/60 to help find a place for X. We divide into two subcases depending on the original location of x, noting that this location must have been either S or some A_i-bin, $i \neq k$.

3.5B.2a.1. *Item x came from S or from an $[A_i,x,F](c')$- or $[A_i,x](c')$-bin.* In all three cases we obtain an additional savings of at least 7/60 for x by the same reasons argued above for x', so send X to S, yielding $\Delta \leqslant 20/60-7/60-14/60 < 0$.

3.5B.2a.2. *Item x came from an $[A_i,X',Y'](c'')$-bin where Y' is a D- or E-item.* By Table IV, $c''=11$. Let z be the other non-A-item in the a_i-bin. Since $headroom_i < 5/12$ and $x > 1/6$, z cannot be a C-item. Revise the A_i-bin to $[A_i,X](10)$ and send z to S, yielding $\Delta \leqslant 15/60-14/60-1/60 = 0$.

3.5B.2b. *Item x' came from an $[A_k,X',Y'](c')$-bin where Y' is a D- or E-item.* By Table IV, $c'=11$. Let z be the other non-A-item in the A_k-bin. Once again, $x \geqslant z$, so replace x by z (unless x is z) and revise the A_k-bin to $[A_k,X](11)$, yielding $\Delta \leqslant -7/60 < 0$.

3.5B.3. $c \geqslant 10$. We divide into subcases depending on the original location of x'.

3.5B.3a. *Item x' came from S or from an $[A_k,x',F](c')$- or $[A_k](c')$-bin.* In the first case, $v(x') \geqslant 6/60$. In the latter two cases we must have $c' \geqslant 7$ and so in all these three cases we can obtain a savings of 6/60 from x', either via $v(x')$, or via revising the A_k-bin to $[A_k,F](0)$ or $[A_k](0)$, respectively. This, together with the savings in ΔCR for the A_j-bin, gives us an aggregate bankroll of 16/60 to help us in finding a place for X. The analysis used in Case 3.5B.2a now applies. Although the lower bound on $v(x)$ is here 6/60 instead of 7/60, we have a bankroll of 16/60 instead of 14/60, which more than cancels the potential reduction in $v(x)$.

3.5B.3b. *Item x' came from an $[A_k,X',Y'](c')$-bin where Y' is a D- or E-item.* By Table IV, $c'=11$. Let z be the other non-A-item in the A_k-bin. Again we must have $x \geqslant z$, so replace x by z (unless x is z) and revise the A_k-bin to $[A_k,X](11)$, yielding $\Delta \leqslant -10/60 < 0$.

3.6. $[A_j](c) \rightarrow [A_j,x,x']$. No further adjustments are necessary, so $\Delta \leqslant -c/60 \leqslant 0$.

This concludes the discussion of Phase 3, and proves that we can arrange that both (H1) and (H2) continue to hold when the phase is complete.

5.5. *Case Analysis: Phase 4 Updates*

At the beginning of the third phase there can be no unprocessed A-bins left of the form $[A,E_1,E_2,F](c)$ or $[A,X,Y](c)$, where Y is a D- or E-item, and there can be at most one A-bin of the form $[A,Z,F_1,F_2](c)$, where $Z > 1/6$. Otherwise, Phase 3 could have processed at least one more bin. If there *is* one $[A,Z,F_1,F_2](c)$-bin with $Z > 1/6$, it must be the case that $c \geqslant 10$. As a preliminary to the first processing of Phase 4, we revise any such bin to $[A,Z,F_1](10)$, sending F_2 to S. This increases $w(\pi)$ by at most 10/60 while not increasing $\Sigma CR(A)$, and so the second part of (H2), which now applies, is still satisfied.

In what follows, we shall denote the elements of *NEW* as $x_1 \geqslant x_2 \geqslant x_3 \geqslant \cdots$, in the order they occur in L. MFFD proceeds through the A-bins from left to right. Let the current bin again be the A_j-bin. If any unpacked item (including F-items this time) will fit in the space currently remaining in the A_j-bin, MFFD puts the largest such item into the A_j-bin, repeating until no unpacked item will fit.

Note that, unlike the third phase, Phase 4 can process previously processed bins. However, in such cases it is particularly easy to update P, S, π, and CR, since by (H1) no processed bin ever contains any items that it does not contain in the MFFD packing. Hence we do not need to find homes for any displaced items, so that $w(\pi)$ and $\Sigma CR(A)$ cannot increase. Thus we may restrict our attention to previously unprocessed bins. Let the current bin be the A_j-bin.

If *NEW* is empty, then so must be *OLD* (by (H1), the members of *OLD* cannot yet have been packed by MFFD, and so if *OLD* contained an item, that item would have fit in the A_j-bin, and so *NEW* could not be empty). Thus we simply relabel the A_j-bin as processed, leaving $w(\pi)$ unchanged and possibly reducing $\Sigma CR(A)$. So we may assume that *NEW* is nonempty, and divide into cases depending on the configuration of the A_j-bin.

4.1. $[A_j,F_1,F_2,F_3](c) \rightarrow [A_j,NEW]$. By Table IV, $c \geqslant 9$. We divide into cases depending on c's value.

4.1A. $c = 9$. By Table IV, we have *headroom*$_j \geqslant 1/3 + F_3$. We divide into cases depending on the type of x_1.

4.1A.1. *Item x_1 is a C-, D-, or E-item.* Then $|OLD - NEW| \geqslant 1$. If $|OLD - NEW| \leqslant 2$, replace x_1 by a member of $OLD - NEW$, for a savings of at least 1/60 by Lemma 6.2, and send the other member (if any) to S, yielding $\Delta \leqslant 10/60 - 1/60 - 9/60 = 0$. If $|OLD - NEW| = 3$, then not only is there an $x_1 \geqslant F_1$, there must also be an $x_2 \geqslant F_3$, so perform both replacements and send the remaining F_i to S, yielding $\Delta \leqslant 10/60 - 1/60 - 9/60 = 0$.

4.1A.2. *Item x_1 is an F-item.* In this case we must have $|NEW| \geqslant 3$, given that *headroom*$_j \geqslant 1/3 + F_3$. Thus we must have $x_1 \geqslant F_1$, $x_2 \geqslant F_2$, and $x_3 \geqslant F_3$. We can thus replace any member of $NEW - OLD$ by a corresponding member of $OLD - NEW$, yielding $\Delta \leqslant -9/60 < 0$.

4.1B. $c \geqslant 10$. We divide into subcases depending on the size of $OLD - NEW$.

4.1B.1. $|OLD-NEW| \leqslant 1$. Send the member (if any) of $OLD-NEW$ to S, yielding $\Delta \leqslant 10/60-10/60 = 0$.

4.1B.2. $|OLD-NEW| = 2$. If x_1 is not F_1 then x_1 did not come from the A_j-bin, so replace x_1 by one member of $OLD-NEW$ and send the other to S, yielding $\Delta \leqslant 10/60-10/60 = 0$. If x_1 is F_1, then F_2 and F_3 are both in $OLD-NEW$ and there must be an x_2 in $NEW-OLD$ such that $x_2 \geqslant F_2$. Replace x_2 by F_2 and send F_3 to S, yielding $\Delta \leqslant 10/60-10/60$.

4.1B.3. $|OLD-NEW| = 3$. If $x_1 \leqslant F_1+F_2$, then there must be an $x_2 \geqslant F_3$. Replace x_1 by F_1, replace x_2 by F_3, and send F_2 to S, yielding $\Delta \leqslant 10/60-10/60 = 0$. If $x_1 > F_1+F_2$, then x_1 is a C-item and we divide into subcases depending on the original location of x_1.

4.1B.3a. *Item x_1 came from S.* Since $headroom_j > 3/7 > 5/12$, there can be no $B \leqslant 5/12$ in S. Thus x_1 cannot have been in a (B,C) pair or a (B,B,C) triple of π (both would imply a $B < (1-2/7)/2 = 5/14 < 5/12$ in S). Moreover, x_1 cannot have been the first item in a (C,Z) pair (because $3x_1+1/7 > 1$). Thus $v(x_1) = 20/60$. Send all three F's in $OLD-NEW$ to S, yielding $\Delta \leqslant 30/60-20/6-10/60 = 0$.

4.1B.3b. *Item x_1 came from an $[A_k,X,Y](c')$-bin, where Y is a D- or E-item.* By Table IV, $c' = 11$. Let $z \neq x_1$ be the other non-A-item in the A_k-bin. Revise the A_k-bin to $[A_k,z,F_1,F_2](11)$ and send F_3 to S, yielding $\Delta \leqslant 10/60-10/60$.

4.1B.3c. *Item x_1 came from an $[A_k,x_1,F](c')$-bin.* By Table IV, $c' \geqslant 10$. Revise the A_k-bin to $[A_k,F_1,F_2,F](c)$; A credit of c is possible by Table IV since we must have $k > j$ and hence $headroom_k \geqslant headroom_j$. (We must have $k > j$ since in Phase 4, all bins to the left of the A_j-bin have already been processed, and so by (H1) cannot contain any as-yet-unpacked items.) If we then send F_3 to S we obtain $\Delta \leqslant 10/60 - 10/60 = 0$.

4.1B.3d. *Item x_1 came from an $[A_k,x_1](c')$-bin.* As before, we must have $headroom_k \geqslant headroom_j$, so revise the A_k-bin to $[A_k,F_1,F_2,F_3](c)$, yielding $\Delta \leqslant -c'/60 < 0$.

4.2. $[A_j,X,F](c) \rightarrow [A_j,NEW]$ *where X is not an F-item.* By Table IV, $c \geqslant 10$. Furthermore, we must by the operation of MFFD have $x_1 \geqslant X$. Replace x_1 by X (unless x_1 is X) and send F to S, yielding $\Delta \leqslant 10/60-10/60 = 0$.

4.3. $[A_j,F_1,F_2](c) \rightarrow [A_j,NEW]$. By Table IV, $c \geqslant 9$. If $c \geqslant 10$ we can proceed as in Case 4.2, so we may assume $c = 9$, in which case $headroom_j \geqslant 1/6+F_2$. If $OLD-NEW$ is empty, $\Delta \leqslant -9/60 < 0$. Otherwise, let z be a member of $OLD-NEW$, and note that $x_1 \geqslant z$. If x_1 is not itself an F-item, replace x_1 by z at a savings of at least 1/60 by Lemma 6.2, and send the at most one remaining item in $OLD-NEW$ to S, yielding $\Delta \leqslant 10/60-1/60-9/60 = 0$. Otherwise, x_1 is an F-item and we must have $|NEW| \geqslant 2$, with $x_2 \geqslant F_2$. We thus can replace every member of $NEW-OLD$ by a corresponding member of $OLD-NEW$, yielding $\Delta \leqslant -9/60 < 0$.

4.4. $[A_j,X](c) \rightarrow [A_j,NEW]$ *where X is any item.* We must have $x_1 \geqslant X$. Therefore, we can replace x_1 by X (unless x_1 is X), yielding $\Delta \leqslant -c/60 \leqslant 0$.

4.5. $[A_j](c) \rightarrow [A_j,NEW]$. By definition $OLD-NEW$ is empty, so $\Delta \leqslant -c/60 \leqslant 0$.

This exhausts the possibilities for Phase 4. Hence (H1) and (H2) continue to hold at the end of Phase 4, which, as we have seen, implies that Lemma 6 holds, which in turn implies Theorem 2. Thus we have at long last completed our proof. ■

6. CONCLUSION

In this paper we have presented an algorithm, MFFD, that provides a performance guarantee considerably stronger than that provided by the classical FFD algorithm. This improvement is obtained without a significant increase in running time and with only a slight increase in programming complexity. Moreover, one can expect that its performance "in practice" will be no worse than that of FFD.

For instance, the expected behavior results proved by Frederickson (1980), Lueker (1982), and Bentley *et al.* (1984) for FFD also hold for MFFD. To be specific, let L_n be a random n-item list with item sizes chosen independently from some fixed distribution on $(0,1]$ that is symmetric about 1/2 (for instance, the uniform distribution on $(0,1]$). Then the expected values of $\mathrm{MFFD}(L_n)$, $\mathrm{FFD}(L_n)$, and $\mathrm{OPT}(L_n)$ are all $n/2+\Theta(\sqrt{n})$, and hence both MFFD and FFD have the best possible asymptotic *expected* ratio to optimal, i.e., 1, and their expected *differences* from optimal have the same (sublinear) growth rate to within an (unknown) multiplicative constant. The result for MFFD can be derived from the observation that the A-bins to which Phase 3 of MFFD applies (the one part of MFFD that differs from FFD) all must have headroom exceeding 1/3 and yet no B-item. Almost surely the number of these is $O(\sqrt{n})$. As to the results of Frederickson (1980) and Lueker (1982) for random lists with all items of size 1/2 or less, here MFFD and FFD have the same expected behavior because they generate the same packings.

Such results do not tell the whole story, of course. One can invent distributions for which FFD outperforms MFFD on average (and vice versa). Moreover, there is some question as to whether *any* of the distributions for which expected performance results have been proved correspond to the ones that occur in practice. However, given the ease of implementing MFFD once one has done the work for FFD, it would seem a relatively inexpensive form of insurance; one can always run them both and take the better result.

The algorithm of Friesen and Langston (1984) mentioned in the Introduction is based on such a "run them both" approach. In the role of FFD-backup, they use an algorithm they call BEST-2 FIT, which we shall abbreviate as "B2F." Although $R^{\infty}_{\mathrm{B2F}} = 5/4 > 11/9$, BEST-2 FIT packs the 11/9 example of Fig. 1 optimally, and Friesen and Langston show that the "hybrid" heuristic that runs both FFD and B2F has $R^{\infty}_{\mathrm{H}} \leqslant 6/5$. This makes it strictly better than either of its components. Moreover, this bound is not known to be tight; B2F also packs the 71/60 example of Figure 2 optimally, so it is even possible that $R^{\infty}_{\mathrm{H}} < 71/60$. All that is currently known is that $R^{\infty}_{\mathrm{H}} \geqslant 9/8$. Unfortunately, even if this is true, the extra insurance that would be provided by using this hybrid algorithm comes at a price in running time, as there seems to be no way of implementing B2F that runs faster than $\Theta(n^2)$.

Running time considerations aside, it would still be nice to know whether the Friesen and Langston hybrid algorithm has a better asymptotic worst-case ratio than MFFD, although it is understandable why the question has not yet been resolved. As the proof presented here indicates, the precise analysis of bin packing heuristics can become complicated indeed. By analogy with Lemma 1, to prove a bound of the form $R^{\infty}_{\mathrm{A}} \leqslant r$ one must consider lists with items as small as $1-1/r$. Thus, as one attempts to prove bounds that are closer and closer to 1, one must pay attention to smaller and smaller items, giving rise to more and more cases, both because configurations with more items per bin are possible and because there are more item types available to go into those configurations. This can normally be expected to make things more complicated to prove. (The results of Fernandez de la Vega and Lueker (1981) and of Karmarkar and Karp (1982) for r arbitrarily close to 1 avoid this difficulty by having the algorithms themselves, rather than the proofs, [illegible] configurations. This allows for simpler proofs but impractical algorithms.)

It is hoped that the organization of the current proof has made it relatively easy to follow. Although many of the basic concepts are adapted from those used in the analysis of FFD by Johnson (1973), the idea of the progressive transformation of the optimal packing into the algorithm's packing is new, and has been a powerful tool in simplifying the case analysis. We believe that the 11/9 proof of Johnson (1973) for FFD can be recast in these terms and thereby considerably shortened and clarified. (A different weighting function w will be needed of course.) Fortunately, we have been spared this undertaking by Brenda Baker (1985), who has already presented a "short" alternative proof of the 11/9 result.

There remains an interesting series of open questions that should be amenable to the type of analysis presented in Section 4 and Appendix 2. Let $R^{\infty}_{\mathrm{FFD}}[\alpha]$ be the asymptotic worst-case ratio for FFD when restricted to lists with item sizes in the interval $(0,\alpha]$. In Section 4 we showed that $R^{\infty}_{\mathrm{FFD}}[1/2] = 71/60$. Johnson (1973) shows that $R^{\infty}_{\mathrm{FFD}}[\alpha]$ equals 71/60 for $8/29 < \alpha \leqslant 1/2$ and that it equals 7/6 for $1/4 < \alpha \leqslant 8/29$. He claims that the value is 23/20 for $1/5 < \alpha \leqslant 1/4$, and conjectures that

$$R^{\infty}_{\mathrm{FFD}}[1/m] = 1 + \frac{1}{m+2} - \frac{2}{m(m+1)(m+2)}$$

for all $m \geqslant 4$ (examples exist that show that this is a lower bound). In principle this could be verified for all m by an appropriate case analysis. Moreover, for $m \geqslant 4$ we need only consider lists whose items are of types m, $m+1$, $m+2$, and $m+3$, and hence this case analysis should prove manageable for values of m considerably larger than those studied to date. Interested readers are invited to try their hands. The insights gained could conceivably lead to a generic proof (or a counterexample).

APPENDIX 1: PROOF OF LEMMA 3

LEMMA 3. *For any list L with all item sizes in the interval* $(1/N, 1/K]$, $N > K \geqslant 2$,

$$\mathrm{MFFD}(L) - \mathrm{FFD}(L) \leqslant V(L) + (N-K).$$

Proof. Let π be an arbitrary legal partition of L into singletons and discountable pairs. We show that $V(\pi) \geqslant \mathrm{FFD}(L) - (N-K)$, from which the lemma will follow by the definition of V.

Let P denote the FFD packing of L, *Regular(P)* the set of regular items in P, and *Surplus(P)* the set of surplus items in P, as defined in Section 4. We then have, by definition, $V(\pi) = W(P) - D(\pi) = W(\mathit{Regular}(P)) + W(\mathit{Surplus}(P)) - D(\pi)$. The obvious extension of Lemma 2 yields

$$W(\mathit{Regular}(P)) \geqslant \mathrm{FFD}(L) - \sum_{i=K}^{N-1} \frac{i-1}{i}. \tag{A1}$$

Thus, all we must show is that

$$W(\mathit{Surplus}(P)) \geqslant D(\pi) - \sum_{i=K}^{N-1} \frac{1}{i}. \tag{A2}$$

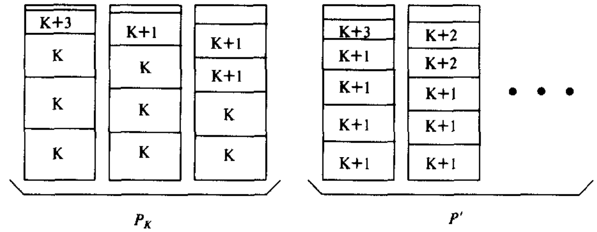

FIG. 5. Paritition of an FFD packing P into P_K and P'.

Our proof of (A2) will be by induction on the difference $N-K$. As a base case for our induction, assume $N-K=1$. In this case all items are of type K, so π contains only singletons, and hence $D(\pi)=0$. Consequently, $W(Surplus(P)) \geqslant D(\pi)$ even if $W(Surplus(P))$ is 0, so (A2) holds when $N-K=1$.

Suppose now that (A2) holds whenever $N-K \leqslant \Delta$, for some $\Delta \geqslant 1$. Let P be the FFD packing of the list L, with all item sizes in $(1/N, 1/K]$, where $N-K=\Delta+1$. If P contains no K-bins, then all item sizes are in $(1/N, 1/(K+1)]$, and since $N-(K+1)=\Delta$ and $\Sigma_{i=K}^{N-1} 1/i > \Sigma_{i=K+1}^{N-1} 1/i$, (A2) holds by the induction hypothesis. Hence we may assume that P contains at least one K-bin.

The basic format of our argument is as follows. Divide P into P_K and P', where P_K consists of the K-bins in P, and P' consists of the remaining bins. See Fig. 5. We will construct a new legal partition π' for the sublist L' of items in P' such that

$$W(Surplus(P_K)) \geqslant D(\pi) - D(\pi') - \frac{1}{K}. \tag{A3}$$

Since P' contains no K-items, we have by the inductive assumption that (A2) holds for P', π', and the interval $(1/N, 1/(K+1)]$, and hence $W(Surplus(P')) \geqslant D(\pi') - \Sigma_{i=K+1}^{N-1} 1/i$. Adding this to (A3) yields (A2) for P, π and the interval $(1/N, 1/K]$.

As a preliminary to the construction of π', let us first normalize π. Suppose there are m pairs in π whose first component is of type K. (In what follows we shall call these *K-pairs.*) Our first step is to modify these m K-pairs so that their first components are the last m (i.e., smallest) K-items in L. This can be done because if (x,y) is discountable and $x' \leqslant x$ is of the same type as x, then (x',y) is also discountable and $D(x',y)=D(x,y)$. Next, rearrange the second items in the K-pairs so that if (x_1,y_1), $(x_2,y_2), \ldots, (x_m,y_m)$ is a listing of these pairs in the order that their first components occur in L, then the y's come in the *reverse* order of their appearance in L (and hence $s(y_1) \leqslant s(y_2) \leqslant \cdots \leqslant s(y_m)$). It is easy to see that this, too, can be done without changing the set of discounts attributable to these pairs (the discount for the pair that contains y_i will be $W(y_i)/K$ both before and after the rearrangement).

In constructing π' from π, all we need worry about are those discountable pairs in π that contain elements from P_K. Most such pairs will simply be replaced by singleton sets for those of their constituent items that remain in P'. If the original pair was one of our m K-pairs (x_i,y_i), then the contribution to $D(\pi)-D(\pi')$ due to losing the discount for that pair is $W(y_i)/K$, for a total of $(1/K)\Sigma_{i=1}^m W(y_i)$ over all such pairs. On the other hand, if the original pair was not one of our K-pairs, then it contains an item z from *Surplus* (P_K), and the contribution to $D(\pi)-D(\pi')$ from the loss of the discount for that pair is more than offset by the contribution of $W(z)$ to $W(\mathit{Surplus}(P_K))$. However, a select group of surplus items in P_K, which we shall denote by $z_1, \ldots, z_J$, will be treated in a special way in order to balance the contribution to $D(\pi)-D(\pi')$ caused by the destruction of the K-pairs.

Here J is the number of K-bins in P_K whose initial (largest) item is one of the m items x_i involved in pairs in π. Note that, given our normalization of π, these are the rightmost J of the K-bins. Proceeding from left to right, let the initial K-items in these bins be denoted by $x_{j(1)}, \ldots, x_{j(J)}$. Consider the K-bin containing $x_{j(i)}$ in P_K. Since $x_{j(i)}$ is the largest K-item in its bin, we have that the sum of the regular items in this bin is at most $K \cdot s(x_{j(i)})$. By the fact that $(x_{j(i)},y_{j(i)})$ is a discountable pair, this implies that $y_{j(i)}$ would fit as the first surplus item above the regular items in the bin. Thus, by the operation of FFD, one of the following two possibilities must hold:

(a) Item $y_{j(i)}$ is in the $x_{j(i)}$-bin or in a K-bin to its left, or

(b) there is a surplus item z in the $x_{j(i)}$-bin that precedes $y_{j(i)}$ in L and hence has $s(z) \geqslant s(y_{j(i)})$.

Our special items z_i will be in one-to-one correspondence with the $y_{j(i)}$'s, with $s(z_i) \geqslant s(y_{j(i)})$. They are chosen inductively as follows. First consider $y_{j(J)}$. If $y_{j(J)}$ is in *Surplus* (P_K), let $z_J = y_{j(J)}$. Otherwise, z_J is the item z promised by (b) above. Now suppose that $z_{h+1},\ldots,z_J$ have been chosen, $h \geqslant 1$, and that each such z_i equals the corresponding $y_{j(i)}$ if that item is in *Surplus* (P_K), and otherwise is some item that precedes $y_{j(i)}$ in L. Now consider $y_{j(h)}$. By our normalization, $y_{j(h)}$ *follows* all the $y_{j(i)}$'s, $i > h$, in L, so $y_{j(h)}$ cannot be one of the already chosen z_i's. If $y_{j(h)}$ is in *Surplus* (P_K), let $z_h = y_{j(h)}$. Otherwise, note that $y_{j(h)}$ would have fit as the first surplus item in any of the $x_{j(i)}$ bins, $h \leqslant i \leqslant J$. Since $y_{j(h)}$ is not in any of these bins, all $J-h+1$ of those bins must contain items that precede $y_{j(h)}$ in L. Since only $J-h$ items have so far been chosen as z_i's, one of these must be as yet unchosen. Let z_j be this item.

Thus we can inductively choose the items z_i as claimed. "Special treatment" for a z_i is required only if z_i is in a pair of π with an item in P'. Note that this can happen only when $z_i \neq y_{j(i)}$ (and hence $y_{j(i)} \notin Surplus(P_K)$). Thus $y_{j(i)}$, which is no larger than z_i by definition, can replace z_i in its pair if z_i is either the second item in the pair or is the first item and of the same type as $y_{j(i)}$. In either case, the contribution to $D(\pi) - D(\pi')$ attributable to this change is no more than $(1/K)(W(z_i) - W(y_{j(i)}))$. If $type(z_i) < type(y_{j(i)})$ and z_i is the first component of its pair, then we simply replace the pair by a singleton, incurring a contribution to $D(\pi) - D(\pi')$ that is at most $(1/type(z_i))(1/(type(z_i)+1))$. However, in both cases the contribution to $D(\pi) - D(\pi')$ is no more than $W(z_i) - W(y_{j(i)})$, and the contribution to $Surplus(P_K)$ from z_i exceeds this by at least $W(z_i) - (W(z_i) - W(y_{j(i)})) = W(y_{j(i)})$. Combining this with the contributions to $D(\pi) - D(\pi')$ from breaking up the original K-pairs, we have that

$$\left[D(\pi) - D(\pi')\right] - W(Surplus(P_K)) \leqslant \frac{1}{K}\sum_{i=1}^{m} W(y_i) - \sum_{i=1}^{J} W(y_{j(i)}). \tag{A4}$$

All that remains to be shown is that the right-hand side of (A4) is at most $1/K$; (A3) will then follow, completing the induction step for (A2), which in turn implies Lemma 3.

The $1/K$ upper bound is argued as follows. Each $y_{j(i)}$ is an element of the list of items $y_1,\ldots,y_m$, with at most $K-1$ y_i's preceding $y_{j(1)}$ in this listing, at most $K-1$ following $y_{j(J)}$, and exactly $K-1$ occurring between $y_{j(h)}$ and $y_{j(h+1)}$, $1 \leqslant h < J$. Moreover, each $y_{j(h)}$ is at least as large as all those items that precede it in this listing, as by our normalization $s(y_1) \leqslant s(y_2) \leqslant \cdots \leqslant s(y_m)$. Thus the weight of each $y_{j(h)}$ in the second sum of the right hand side of (A4) can offset the contributions of itself and $K-1$ of its predecessors to the first sum of the right hand side of (A4), since each of these contributes only $1/K$ of its weight to that sum. The total weight that is not so offset is simply that for the at most $K-1$ y_i's that follow $y_{j(J)}$ in the listing, and, since each y_i has weight at most $1/(K+1)$, this remaining contribution can be at most $(1/K)(K-1)(1/(K+1)) < 1/K$, as desired. ∎

APPENDIX 2: PROOF OF LEMMA 5

LEMMA 5. *For any list L composed of only B-, C-, D-, E-, and F-items, there exists a legal partition π of L such that*

$$w(\pi) \leqslant \frac{71}{60}\mathrm{OPT}(L).$$

Proof. It is enough to consider only lists L such that $\mathrm{OPT}(L) = 1$, i.e., such that $s(L) \leqslant 1$. The general claim will follow, since any list L can be viewed as the union of lists $L_1, \ldots, L_{\mathrm{OPT}}(L)$, where L_i is the sublist of L consisting of the items in the ith bin of an optimal packing for L. Each such L_i satisfies $s(L_i) \leqslant 1$ and hence if the lemma holds for such L, each will have a partition π_{L_i} such that $w(\pi_{L_i}) \leqslant 71/60$. The union of the π_{L_i} will hence be a partition π of L satisfying the lemma.

So assume $s(L) \leqslant 1$. (As usual, we also assume that the items in L are in non-increasing order by size.) The proof that L satisfies the lemma is by case analysis, where the cases range over all possible configurations for such a list L.

We shall use the notation $[X_1, X_2, \ldots, X_k]$, where each "X" is a symbol from the set $\{B,C,D,E,F\}$, to represent any list $L = (x_1, x_2, \ldots, x_k)$, such that each x_i has the type of the corresponding X_i. For example, $[B_1, B_2, C_3]$ represents any configuration with two B-items and one C-item. We shall also use the following shorthands:

"X_i" is a shorthand for $s(x_i)$.

"W" is a shorthand for $W(L)$, the weight of the set before discounts.

"w" is a shorthand for the the value of $w(\pi)$ for the implied partition.

To further streamline our analysis, we shall use the concept of "domination." We say that configuration $[X_1, \ldots, X_k]$ *dominates* configuration $[Y_i, \ldots, Y_k]$ if, for $1 \leqslant i \leqslant k$, X_i has type at least as large (at least as early in the alphabet) as Y_i. Note that if $[X_1, \ldots, X_k]$ dominates $[Y_1, \ldots, Y_k]$ and $W \leqslant 71/60$ for $[X_1, \ldots, X_k]$, then we can get $w \leqslant 71/60$ for both configurations by simply taking the partitions into singletons, in which case $w = W$.

We now begin the case analysis proper.

Case 1. *One-Item Lists*

This class is dominated by $[B_1]$, for which $w \leqslant W = 1/2 < 71/60$.

Case 2. *Two-Item Lists*

This class is dominated by $[B_1, B_2]$, for which $W = 2(1/2) < 71/60$.

Case 3. *Three-Item Lists*

3.1. Two or more B's:

$[B_1, B_2, B_3]$. Impossible, as $s(L) > 1$.

$[B_1, B_2, C_3]$. By definition, (B_1, B_2, C_3) is discountable. Thus we have $w = (30+30+20-9)/60 = 71/60$.

$[B_1, B_2, D_3]$. $2B_2 + D_3 \leqslant B_1 + B_2 + D_3 \leqslant 1$, so (B_2, D_3) is discountable and $w = (30+30+15-6)/60 < 71/60$.

$[B_1,B_2,E_3]$. As above, (B_2,E_3) is discountable, so $w \leqslant (30+30+12-4)/60 < 71/60$.

$[B_1,B_2,F_3]$. $W = (30+30+10)/60 < 71/60$.

3.2. One or fewer B's: Class dominated by

$[B_1,C_2,C_3]$. $W = (30+20+20)/60 < 71/60$.

Case 4. *Four-Item Lists*

4.1. Three or more B's: Impossible, as $s(L) > 1$.

4.2. Two B's and a C or a D: Impossible, as $s(L) > 2/3+1/5+11/71 > 1$.

4.3. Two B's and two E's: Impossible, as $s(L) > 2/3+2/6 = 1$.

4.4. Legal configurations with two B's:

$[B_1,B_2,E_3,F_4]$. $B_1+F_4 < 1-1/3-1/6 = 1/2$, so $2B_1+F_4 \leqslant 1$, and (B_1,F_4) is discountable. Furthermore, $B_2+E_3 < 1-1/3-1/7 = 11/21$, so $2B_2+E_3 < 22/21-1/6 < 1$ and (B_2,E_3) is discountable. Moreover, $B_2 < 1-1/3-1/6-1/7 = 5/14$, so $d(B_2,E_3) = 6/60$. Thus we have $w = (30+30+12+10-5-6)/60 = 71/60$.

$[B_1,B_2,F_3,F_4]$. $2B_1+F_3 < 2(1-1/3-1/7)-1/7 = 18/21$, so (B_1,F_3) is discountable, as is (B_2,F_4) by the same argument. Thus $w = (30+30+10+10-5-5)/60 < 71/60$.

4.5. One B, two or more C's:

$[B_1,C_2,C_3,X_4]$ for $X_4 = C$, D, or E. Impossible, as $s(L) > 1/3+2/4+1/6 = 1$.

$[B_1,C_2,C_3,F_4]$. $2B_1+C_2 < 2(1-1/4-1/7)-1/4 = 17/14-1/4 < 1$, so (B_1,C_2) is discountable. Furthermore, $3C_3+F_4 < 3(1-1/3-1/4)-2(1/7) = 5/4-2/7 < 1$, so (C_3,F_4) is discountable. Thus $w = (30+20+20+10-6-3)/60 = 71/60$.

4.6. One B, one C:

$[B_1,C_2,D_3,D_4]$. $2B_1+D_3 \leqslant 2(1-1/4-1/5)-1/5 = 11/10-1/5 < 1$, so (B_1,D_3) is discountable. Furthermore, $3C_2+D_4 \leqslant 3(1-1/3-1/5)-2(1/5) = 1$, so (C_2,D_4) is discountable. Thus $w = (30+20+15+15-6-5)/60 < 71/60$.

$[B_1,C_2,D_3,E_4]$. $2B_1+D_3 \leqslant 2(1-1/4-1/6)-1/5 = 7/6-1/5 < 1$, so (B_1,D_3) is discountable. Thus $w = (30+20+15+12-6)/60 = 71/60$.

$[B_1,C_2,D_3,F_4]$. $2B_1+F_4 < 2(1-1/4-1/5)-1/7 = 11/10-1/7 < 1$, so (B_1,F_4) is discountable. Thus $w = (30+20+15+10-5)/60 < 71/60$.

$[B_1,C_2,E_3,E_4]$. $2B_1+E_3 \leqslant 2(1-1/4-1/6)-1/6 = 1$, so (B_1,E_3) is discountable. Thus $w \leqslant (30+20+12+12-4)/60 < 71/60$.

$[B_1,C_2,E_3,F_4]$. If $B_1+F_4 \leqslant 4/7$, then $2B_1+F_4 < 8/7-1/7 = 1$, so (B_1,F_4) is discountable and $w = (30+20+12+10-5)/60 < 71/60$. Otherwise, $C_2+E_3 < 3/7$, so $3C_2+E_3 < 9/7-2/6 < 1$, so (C_2,E_3) is discountable and $w \leqslant (30+20+12+10-2)/60 < 71/60$.

$[B_1,C_2,F_3,F_4]$. $W = (30+20+10+10)/60 < 71/60$.

4.7. One B, no C, two or more D's:

$[B_1,D_2,D_3,D_4]$. $2B_1+D_2 \leqslant 2(1-2/5)-1/5 = 1$, so (B_1,D_2) is discountable. $w = (30+15+15+15-6)/60 < 71/60$.

$[B_1,D_2,D_3,E_4]$. If $4D_3+E_4 \leqslant 1$, then (D_4,E_4) is discountable and $w = (30+15+15+12-2)/60 < 71/60$. Otherwise, $D_2 \geqslant D_3 \geqslant (1-E_4)/4$, so that $B_1+E_4 \leqslant 1-(1-E_4)/2$ and $2B_1+E_4 \leqslant (1+E_4)-E_4 = 1$. Thus (B_1,E_4) is discountable and $w <= (30+15+15+12-4)/60 < 71/60$.

$[B_1,D_2,D_3,F_4]$. $W = (30+15+15+10)/60 < 71/60$.

4.8. One B, no C, no more than one D: Class dominated by

$[B_1,D_2,E_3,E_4]$. $W = (30+15+12+12)/60 < 71/60$.

4.9. No B, three or more C's:

$[C_1,C_2,C_3,C_4]$. Impossible, as $s(L) > 1$.

$[C_1,C_2,C_3,D_4]$. $3C_3+D_4 \leqslant C_1+C_2+C_3+D_4 \leqslant 1$, so (C_3,D_4) is discountable and $w = (20+20+20+15-5)/60 < 71/60$.

$[C_1,C_2,C_3,E_4]$. For the analogous reason, (C_3,E_4) is discountable and so $w = (20+20+20+12-2)/60 < 71/60$.

$[C_1,C_2,C_3,F_4]$. $W = (20+20+20+10)/60 < 71/60$.

4.10. No B, no more than two C's: Class dominated by

$[C_1,C_2,D_3,D_4]$. $W = (20+20+15+15)/60 < 71/60$.

Case 5. *Five-Item Lists*

5.1. Two or more B's: Impossible, as $s(L) > 2/3+3/7 = 23/21$.

5.2. One B, one or more C's: Impossible, as $s(L) > 1/3+1/4+3/7 = 85/84$.

5.3. One B, no C, two or more D's: Impossible, as $s(L) > 1/3+2/5+2/7 = 107/105$.

5.4. One B, no C, one D, one or more E's: Impossible, as $s(L) > 1/3+1/5+1/6+22/71 > 1.009$.

5.5. Remaining configurations with one B:

$[B_1,D_2,F_3,F_4,F_5]$. $2B_1+D_2 < 2(1-3/7)-1/5 < 1$, so (B_1,D_2) is discountable and $w \leqslant (30+15+10+10+10-6)/60 < 71/60$.

$[B_1,E_2,E_3,E_4,E_5]$. Impossible, as $s(L) > 1/3+4/6 = 1$.

$[B_1,X_2,X_3,X_4,F_5]$, where each X is an E or an F. $2B_1+F_5 < 2(1-3/7)-1/7 = 1$, so (B_1,F_5) is discountable and $w \leqslant (30+12+12+12+10-5)/60 = 71/60$.

5.6. No B, three or more C's: Impossible, as $s(L) > 3/4+2/7 > 1$.

5.7. No B, two C's, one or more D's: Impossible, as $s(L) > 2/4+1/5+22/71 > 1.009$.

5.8. Remaining configurations with no B, two C's:

$[C_1,C_2,E_3,E_4,E_5]$. Impossible, as $s(L) > 2/4+3/5 > 1$.

$[C_1,C_2,E_3,E_4,F_5]$. $3C_1+F_5 < 3(1-1/4-2/6)-2/7 = 5/4-2/7 < 1$, so (C_1,F_5) is discountable, and $w = (20+20+12+12+10-3)/60 = 71/60$.

$[C_1,C_2,X_3,F_4,F_5]$, where X_3 is an E- or F-item. If $C_1+F_4 \leqslant 3/7$, then $3C_1+F_4 < 9/7-2/7 = 1$, so (C_1,F_4) is discountable. Otherwise, $C_2+F_5 < 1-3/7-1/7 < 3/7$, so (C_2,F_5) is discountable. In either case, $w \leqslant (20+20+12+10+10-3)/60 < 71/60$.

5.10. No B, one C, two or more D's:

$[C_1,D_2,D_3,D_4,X_5]$, where $X = D$, E, or F. Impossible, as $s(L) > 1/4+3/5+11/71 > 1.004$.

$[C_1,D_2,D_3,E_4,E_5]$. $3C_1+D_3 \leqslant 3(1-1/5-2/6)-2/5 = 1$, so (C_1,D_4) is discountable and $w = (20+15+15+12+12-5)/60 < 71/60$.

$[C_1,D_2,D_3,E_4,F_5]$. If $4D_3+E_4 \leqslant 1$, then (D_3,E_4) is discountable and $w = (20+15+15+12+10-2)/60 < 71/60$. Otherwise, $D_2 \geqslant D_3 \geqslant (1-E_4)/4$, so $C_1+E_4 < 1-(1-E_4)/2-1/7 = 5/14+E_4/2$. Thus $3C_1+E_4 \leqslant 15/14+(3/2)E_4-2E_4 < 15/14-1/12 < 1$, so (C_1,E_4) is discountable and $w = (20+15+15+12+10-2/60 < 71/60$.

$[C_1,D_2,D_3,F_4,F_5]$. $W = (20+15+15+10+10)/60 < 71/60$.

5.11. No B, one C, one D: Class dominated by

$[C_1,D_2,E_3,E_4,E_5]$. $W = (20+15+12+12+12)/60 = 71/60$.

5.12. No B, one C, no D: Class dominated by

$[C_1,E_2,E_3,E_4,E_5]$. $W = (20+12+12+12+12)/60 < 71/60$.

5.13. No B or C, four or more D's:

$[D_1,D_2,D_3,D_4,D_5]$. Impossible, as $s(L) > 1$.

$[D_1,D_2,D_3,D_4,X_5]$, where $X = E$ or F. $4D_4+X_5 \leqslant D_1+D_2+D_3+D_4+X_5 \leqslant 1$, so (D_4,X_5) is discountable, and $w \leqslant (4 \cdot 15+12-2)/60 < 71/60$.

5.14. No B or C, three or fewer D's: Class dominated by

$[D_1,D_2,D_3,E_4,E_5]$. $W = (3\cdot 15+2\cdot 12)/60 < 71/60$.

Case 6. *Six-Item Lists*

6.1. One or more B's or C's: Impossible, as $s(L) > 1/4+55/71 > 1.02$.

6.2. No B or C, two or more D's: Impossible, as $s(L) > 2/5+44/71 > 1.01$.

6.3. No B or C, one D, three or more E's: Impossible, as $s(L) > 1/5+3/6+22/71 > 1.009$.

6.4. Legal configurations with no B or C, one D: Class dominated by

$[D_1,E_2,E_3,F_4,F_5,F_6]$. $W = (15+2\cdot 12+3\cdot 10)/60 < 71/60$.

6.5. No B, C, or D:

$[E_1,E_2,E_3,E_4,E_5,E_6]$. Impossible, as $s(L) > 1$.

$[X_1,X_2,X_3,X_4,X_5,F_6]$, where each $X_i = E$ or F. $W \leqslant (5\cdot 12+10)/60 < 71/60$.

This exhausts all possibilities for lists with six or fewer items, and since no seven items of size exceeding 11/71 can fit in a single bin, completes the proof of Lemma 5. ■

References

Baker, B. S. (1985), A new proof for the first fit decreasing bin packing algorithm, *J. Algorithms* **6**, 49–70.

Bentley, J. L., Johnson, D. S., Leighton, T., and McGeoch C. C. (1983), An experimental study of bin packing, *in* "Proceedings, 21st Ann. Allerton Conf. on Communication, Control, and Computing," pp. 51–60, Department of Electrical Engineering and the Coordinated Science Laboratory, University of Illinois, Urbana, Ill.

Bentley, J. L., Johnson, D. S., Leighton, T., McGeoch, C. C., and McGeoch L. A. (1984), Some unexpected expected behavior results for bin packing, *in* "Proceedings, 16th Ann. ACM Symp. on Theory of Computing," pp. 279–288, Association for Computing Machinery, New York.

Brown, A. R. (1971), "Optimal Packing and Depletion," American Elsevier, New York.

Coffman, E. G., Jr, Garey, M. R., and Johnson D. S. (1985), Approximation algorithms for bin-packing – An updated survey, *in* "Algorithm Design for Computer System Design," pp. 49–106, G. Ausiello, M. Lucertini, and P. Serafini (Eds.), Springer-Verlag, New York.

Fernandez de la Vega, W., and Lueker G. S. (1981), Bin packing can be solved within $1+\epsilon$ in linear time, *Combinatorica* **1**, 312–320.

Frederickson, G. N. (1980), Probabilistic analysis for simple one- and two-dimensional bin packing algorithms, *Inform. Process. Lett.* **11**, 156–161.

Friesen, D. K., and Langston M. A. (1984), Analysis of a compound bin-packing algorithm, manuscript.

GAREY, M. R., AND JOHNSON D. S. (1979), "Computers and Intractability: A Guide to the Theory of NP-Completeness," Freeman, San Francisco.

GAREY, M. R., AND JOHNSON D. S. (1981), Approximation algorithms for bin packing problems – A survey, *in* "Analysis and Design of Algorithms in Combinatorial Optimization," pp. 147–172, G. Ausiello and M. Lucertini (Eds.), Springer-Verlag, New York.

GILMORE, P. C., AND GOMORY R. E. (1961), A linear programming approach to the cutting stock problem, *Oper. Res.* **9**, 849–859.

GILMORE, P. C., AND GOMORY R. E. (1963), A linear programming approach to the cutting [illegible]

[illegible] quences in combinatorial optimization, *Combinatorica* **1**, 169–197.

JOHNSON, D. S. (1973), Near-Optimal Bin Packing Algorithms, Doctoral Dissertation, Department of Mathematics, Massachussetts Institute of Technology, Cambridge, Mass.

JOHNSON, D. S., DEMERS, A., ULLMAN, J. D., GAREY, M. R., AND GRAHAM R. L. (1974), Worst-case performance bounds for simple one-dimensional packing algorithms, *SIAM J. Comput.* **3**, 299–325.

KARMARKAR, N., AND KARP R. M. (1982), An efficient approximation scheme for the one-dimensional bin packing problem, *in* "Proceedings, 23rd Ann. Symp. on Foundations of Computer Science," pp. 312–320, IEEE Computer Society, Los Angeles.

KHACHIYAN, L. G. (1979), A polynomial algorithm in linear programming, *Dokl. Akad. Nauk. SSSR* **244**, 1093–1096 [in Russian]; English transl. in *Soviet Math. Dokl.* **20**, 191–194.

LUEKER, G. S. (1982), "An Average-Case Analysis of Bin Packing with Uniformly Distributed Item Sizes," Report No. 181, Department of Information and Computer Science, University of California, Irvine, Calif.

YAO, A. C. (1980), New algorithms for bin packing, *J. Assoc. Comput. Mach.* **27**, 207–227.