Acta Informatica 26, 697–709 (1989)

An On-Line Algorithm for Variable-Sized Bin Packing

J. Csirik*

Department of Computer Science, József Attila University, Szeged, Somogyi u. 7, H-6720 Szeged, Hungary

Summary. An on-line algorithm for variable-sized bin packing with an asymptotical worst-case ratio of <1.7 is given. The method is derived from the Harmonic Fit proposed by Lee and Lee. It is proven that, if it is allowed to choose any bin sizes, then with an appropriately chosen second bin size we can have an asymptotical worst-case ratio of 1.4, even with the same algorithm and two bin sizes.

Introduction

In variable-sized bin packing we are given a list

$$L=(a_1, a_2, \ldots, a_n)$$

of items (elements), each with item size $s(a_i)$ $(0<s(a_i)\leqq 1)$ and a finite collection of bin sizes. Our aim is to pack the items into the bins so that the sum of the sizes of the bins used is minimum.

The variable-sized bin packing problem is NP-hard (Friesen and Langston [3]), and thus efficient heuristic algorithms which ensure near-optimal packings are required. Friesen and Langston [3] gave three approximation algorithms, with performance ratios (asymptotical worst-case ratios) of 2, 3/2 and 4/3. Unfortunately, only the first of these algorithms is on-line (which means that it packs the elements in the order they are given, without a knowledge of the later elements). Murgolo [7] presented an ε-approximation scheme, which for any $\varepsilon>0$ yields an approximation algorithm with a performance ratio of $1+\varepsilon$. However, these algorithms are not on-line either. Very recently, Kinnerley and Langston [4] gave fast on-line algorithms for the variable-sized bin packing. They devised a scheme based on a user specified factor $f\geqq 1/2$, and proved that their strategy guarantees a worst-case bound not exceeding $1.5+f/2$.

* Part of this work was carried out when the author was visiting the Institut für Informatik und angewandte Mathematik, Berne, Switzerland

This paper has been supported by Grant OTKA Nr. 1135 from the Hungarian Academy of Sciences

On the other hand, in "classical" bin packing (where we have only one bin size) it is known that no on-line algorithm can have an asymptotical worst-case ratio $<1.53...$ (Brown [1]; Liang [5]). The best on−line algorithm to date is the "modified Harmonic Fit" given by Lee and Lee [6], which has a worst-case ratio of 1.63... .

A third line of research was also done by Friesen and Langston [2]. Their question was: what is the best bin size for a given list in the sense that the wasted space is minimized?

In this paper we give an on-line algorithm for variable-sized bin packing with an asymptotical worst-case ratio of <1.7. The method is derived from the Harmonic Fit proposed by Lee and Lee [6]. We shall prove that, if we are allowed to choose any bin sizes, then with an appropriately chosen second bin size we can have an asymptotical worst-case ratio of 1.4, even with the same algorithm and two bin sizes!

Definitions

Let k denote the number of different bin sizes available. We assume that there is an inexhaustible supply of bins of each size. Let us denote the different bins by $B_1, B_2, \ldots, B_k$, and their sizes by $s(B_1), s(B_2), \ldots, s(B_k)$. Without loss of generality we may assume that $s(B_1)>s(B_2)>\ldots>s(B_k)$. We suppose further that the elements of

$$L=(a_1, a_2, \ldots, a_n)$$

and the bins are so normalized that the largest bin has a size of 1. We shall sometimes use the list in the form of

$$L=(s(a_1), s(a_2), \ldots, s(a_n)),$$

i.e. as a list of real numbers. Let

$$s(L)=\sum_{i=1}^{n} s(a_i),$$

and let

$$B(A, L)=(B_A^1, B_A^2, \ldots, B_A^l)$$

denote the list of bins used by a heuristic algorithm A. Let

$$B(*, L)=(B_*^1, B_*^2, \ldots, B_*^m)$$

denote the list of bins used in some optimal packing of the list L. Then

$$s(B(A, L))=\sum_{i=1}^{l} s(B_A^i)$$

is the total size required by algorithm A with respect to L, and

$$s(B(*, L)) = \mathrm{OPT}(L) = \sum_{i=1}^{m} s(B_*^i)$$

is the minimum (optimal) size to pack the list L.

We shall give our results in the following way: we shall prove that for a heuristic algorithm A and every list L

$$s(B(A, L)) \leqq R \cdot \mathrm{OPT}(L) + C \tag{1}$$

holds. The smallest possible R will be called the asymptotic worst-case ratio of A, and will be denoted by R_A. Instead of the asymptotic worst-case ratio, we shall sometimes say that our bound is tight.

The Algorithm Variable Harmonic M (VH_M)

Let $M > 1$ be a positive integer and let $M_j = \lceil M \cdot s(B_j) \rceil$ $(j = 1, 2, \ldots, k)$, where $\lceil x \rceil$ denotes the smallest integer not less than x. We shall define our algorithm only for those M where $M_k \geqq 2$. Let us divide the intervals $(0, s(B_j)]$ $(j = 1, 2, \ldots, k)$ into M_j parts according to the following "harmonic partitioning":

$$I_{j,l} = \left(\frac{s(B_j)}{l+1}, \frac{s(B_j)}{l} \right], \quad j = 1, 2, \ldots, k;\ l = 1, 2, \ldots, M_j - 1,$$

and

$$I_{j,*} = \left(0, \frac{s(B_j)}{M_j} \right].$$

For each bin size $s(B_j)$ we define a weighting function as follows:

$$W_j(a_i) = \begin{cases} \dfrac{M_j}{M_j - 1} s(a_i) & \text{if } s(a_i) \in I_{j,*}, \\ \dfrac{s(B_j)}{l} & \text{if } s(a_i) \in I_{j,l},\ l = 1, 2, \ldots, M_j - 1, \\ \infty & \text{if } s(a_i) > s(B_j). \end{cases} \tag{2}$$

Now let

$$W(a_i) = \min_{j = 1, 2, \ldots, k} W_j(a_i) \tag{3}$$

the weight of a_i.

Example. Let $M = 4$, $L = (0.8, 0.1, 0.6, 0.4, 0.25)$, and $s(B_1) = 1$, $s(B_2) = 0.7$, $s(B_3) = 0.3$.

Then $M_1=4$, $M_2=3$, $M_3=2$, and

$I_{1,1}=(0.5, 1]$; $I_{1,2}=(1/3, 0.5]$; $I_{1,3}=(0.25, 1/3]$; $I_{1,\#}=(0,0.25]$;

$I_{2,1}=(0.35, 0.7]$; $I_{2,2}=(7/30, 0.35]$; $I_{2,\#}=(0, 7/30]$;

$I_{3,1}=(0.15, 0.3]$; $I_{3,\#}=(0, 0.15]$.

For the weights:

$W_1(a_1)=1$;	$W_2(a_1)=\infty$;	$W_3(a_1)=\infty$;	$W(a_1)=1$;
$W_1(a_2)=4/30$;	$W_2(a_2)=0.15$;	$W_3(a_2)=0.2$;	$W(a_2)=4/30$;
$W_1(a_3)=1$;	$W_2(a_3)=0.7$;	$W_3(a_3)=\infty$;	$W(a_1)=0.7$;
$W_1(a_4)=0.5$;	$W_2(a_4)=0.7$;	$W_3(a_4)=\infty$;	$W(a_4)=0.5$;
$W_1(a_5)=1/3$;	$W_2(a_5)=0.35$;	$W_3(a_5)=0.3$;	$W(a_5)=0.3$. □

The algorithm VH_M works as follows:
In the first step we assign to each element of the list a bin size:

1. B_1 is assigned to each element a_i with

$$W(a_i)=(M_1/(M_1-1))\, s(a_i).$$

These elements will be called small elements, and all others big elements.

2. A big a_i will be a B_j element if j is the smallest integer such that

$$W(a_i)=W_j(a_i).$$

A big element will be called an $I_{j,l}$ element if it is a B_j element and belongs to $I_{j,l}$.

In the second step of VH_M we shall perform a Harmonic fit-type packing:

a) All big B_j elements will be packed by Harmonic Fit into B_j bins as follows: we classify the B_j bins into M_j-1 categories. Each category is designated to pack the same type of elements. A B_j bin designated to pack $I_{j,l}$ items is called an $I_{j,l}$ bin. Clearly, each $I_{j,l}$ bin has room for exactly l pieces. We use a Next-Fit packing in all $I_{j,l}$ ($j=1, 2, \ldots, k$; $l=1, 2, \ldots, M_j-1$) bins, i.e. after packing l pieces into an $I_{j,l}$ bin, we close this bin and open a new $I_{j,l}$ bin.

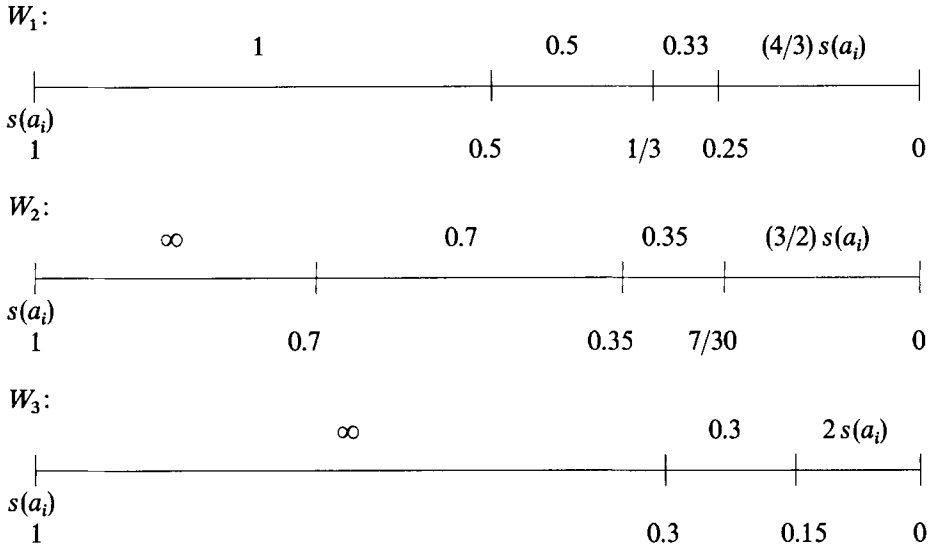
b) All small elements will be packed in B_1 bins by Next-Fit, i.e. if the next small element does not fit into the opened bin, then we close this bin and open a new one. These elements are called $I_\#$ items, and these bins are called $I_\#$ bins.

Thus, we have at most a number of

$$P=M_1+M_2+\ldots+M_k-k+1$$

series (categories) of bins, each designated to pack the same ($I_{j,l}$ or $I_\#$) type of items.

Example (continued). The weighting functions:



and therefore

Interval	Element	Weight
(0.7, 1]	$I_{1,1}$	1
(0.5, 0.7]	$I_{2,1}$	0.7
(0.35, 0.5]	$I_{1,2}$	0.5
(1/3, 0.35]	$I_{2,2}$	0.35
(0.3, 1/3]	$I_{1,3}$	1/3
(9/40, 0.3]	$I_{3,1}$	0.3
(0, 9/40]	$I_{\#}$	$(4/3)\,s(a_i)$

which means that in our list

a_1 is an $I_{1,1}$ piece,
a_2 is an $I_{\#}$ piece,
a_3 is an $I_{2,1}$ piece,
a_4 is an $I_{1,2}$ piece,
a_5 is an $I_{3,1}$ piece.

In packing of an arbitrary list into these bins we would have 7 series of bins ($I_{1,1}, I_{1,2}, I_{1,3}, I_{2,1}, I_{2,2}, I_{3,1}$ and $I_{\#}$ bins). □

The item classification can clearly be done in $O(\log P)$ time and we have at most P active bins, and hence the space requirement is $O(1)$ for fixed M. The running time is therefore $O(n)$ at a space requirement of $O(1)$. Our algorithm is obviously on-line.

The Asymptotic Worst-Case Ratio of VH_M

Let us first recall the main result of Lee and Lee [6]. Let

$$P(1)=\left\{(y_1, y_2, \ldots, y_t)\,\middle|\, t\in\mathbb{N}, y_i>0, 1\leqq i\leqq t, \text{ and } \sum_{i=1}^{t} y_i\leqq 1\right\}$$

be the set containing all possible partitions of any positive real numbers $\leqq 1$, and let

$$G_M(1)=\sup_{P(1)} \sum_{i=1}^{t} W_1(y_i),$$

and

$$G_M^{\$}(1)=\sup_{P(1)} \sum_{i=1}^{t} W(y_i),$$

where W_1 and W are defined in (2) and (3). It is obvious that

$$G_M^{\$}(1)\leqq G_M(1).$$

We define the sequence r_i as follows:

$$r_1=1, \quad r_{i+1}=r_i(r_i+1) \quad \text{for } i\geqq 1.$$

Then, Theorem 1 from Lee and Lee is the following:

Theorem 1 [6]. *Let $M\geqq 3$. For $i>1$, $r_i<M\leqq r_{i+1}$,*

$$G_M(1)=\sum_{l=1}^{i}\frac{1}{r_l}+\frac{M}{(M-1)r_{i+1}}. \quad \square \tag{4}$$

It is clear that $G_M(1)$ decreases monotonically as $M\to\infty$. A short computation gives $G_2(1)=2$, and therefore Theorem 1 is true for $M\geqq 2$, $i\geqq 1$, too. Lee and Lee proved that $G_M(1)$ is the asymptotic worst-case ratio of Harmonic Fit with M intervals in classical bin packing.

This result can easily be extended to all other bin sizes. In this case, let

$$P(s(B_j))=\left\{(y_1, y_2, \ldots, y_t)\,\middle|\, t\in\mathbb{N}, y_i>0, 1\leqq i\leqq t, \text{ and } \sum_{i=1}^{t} y_i\leqq s(B_j)\right\}, \quad j=2, 3, \ldots, k,$$

$$G_M(s(B_j))=\sup_{P(s(B_j))} \sum_{i=1}^{t} W_j(y_i),$$

$$G_M^{\$}(s(B_j))=\sup_{P(s(B_j))} \sum_{i=1}^{t} W(y_i).$$

Here also

$$G_M^{\$}(s(B_j)) \leqq G_M(s(B_j)),$$

and the following emerges immediately from Theorem 1.

Corollary 1. *If* $M_j \geqq 2$, $i(j) \geqq 1$, $r_{i(j)} < M_j \leqq r_{i(j)+1}$, *then*

$$G_M(s(B_j)) = \left(\sum_{l=1}^{i(j)} \frac{1}{r_l} + \frac{M_j}{(M_j - 1) r_{i(j)+1}} \right) s(B_j)$$

$$= G_{M_j}(1) \cdot s(B_j),$$

where G_{M_j} *is defined as in* (4). □

Consider now an optimal packing of L that uses $B(*, L)$. Then, L can be expressed as the concatenation of a number of m sublists, each of which corresponds to the content of $B_*^1, \ldots, B_*^m$. But then

$$W(L) \leqq \sum_{p=1}^{m} G_M^{\$}(s(B_*^p)) \leqq \sum_{p=1}^{m} G_M(s(B_*^p))$$

$$= \sum_{p=1}^{m} (G_{M_*^p}(1) \cdot s(B_*^p))$$

$$\leqq G_{M_k}(1) \sum_{p=1}^{m} s(B_*^p) = G_{M_k}(1) \cdot \mathrm{OPT}(L) \tag{5}$$

because of Corollary 1, and since $G_M(1)$ decreases monotonically as $M \to \infty$.

On the other hand, we know that in the packing by VH_M, in each full B_f bin which contains big items, the sum of the weights W of items is $s(B_f)$ and in all B_* bins this sum is not smaller than one, i.e. $\geqq s(B_*) = s(B_1) = 1$. Let

$$Q = 1 + \sum_{i=1}^{k} (M_i - 1) \cdot s(B_i),$$

which is the maximum of the sum of the sizes of the opened bins. We can classify the bins used by VH_M into two groups: in the first are the full bins (for these the sum of the sizes $\leqq$ the sum of weights W), and in the second the opened ones (the maximum of the sum of the sizes of these bins is Q). Hence

$$s(B(\mathrm{VH}_M, L)) - Q \leqq W(L), \tag{6}$$

and, rewriting (5) and (6) in the form (1), we get the following result:

Theorem 2. *Let* $M_k = \lceil M \cdot s(B_k) \rceil \geqq 2$. *If* $r_{i(k)} < M_k \leqq r_{i(k)+1}$, *then*

$$R_{\mathrm{VH}_M} \leqq \sum_{l=1}^{i(k)} \frac{1}{r_l} + \frac{M_k}{(M_k - 1) r_{i(k)+1}} = G_{M_k}. \quad \square$$

From (4) we obtain $G_6(1) = 1.7$ and $G_7(1) = 1.694\ldots$. Thus the asymptotic worst-case ratio is < 1.7 if for the bins $B_1, B_2, \ldots, B_k$ we choose M so that $M_k \geqq 7$. It is easy to see that

$$\lim_{M_k \to \infty} G_{M_k} = \sum_{l=1}^{\infty} \frac{1}{r_l} = 1 + \frac{1}{2} + \frac{1}{6} + \ldots = 1.691\ldots.$$

For general bin sizes it is not very easy to decide whether Theorem 2 gives a tight bound or not, but special cases may also be interesting. For example, for two bin sizes we can choose the second bin size so that the bound is tight, or so that it is much smaller than given in Theorem 2. To show the first, we consider the list defined by Lee and Lee: let $M = r_{i+1}$, $i \geqq 2$. Consider items of $i+2$ sizes:

$$a_j = 1/(r_j + 1) + 1/c, \qquad 1 \leqq j \leqq i,$$

$$a_{i+1} = 1/r_{i+1} - i/c,$$

$$a_{i+2} = 1/d,$$

where c and d are sufficiently large integers. With this list, it was proved that the asymptotic worst-case bound of Harmonic Fit for classical bin packing is tight. It is now clear that we get the same result if we choose the second bin size so that $s(B_2) = 1/2 + 1/(2c)$, because we use only B_1 bins in both optimal and Variable Harmonic algorithms.

On the other hand, it is an interesting question as to how we can choose the bin sizes so that the asymptotic worst-case ratio is smaller than the one given in Theorem 2.

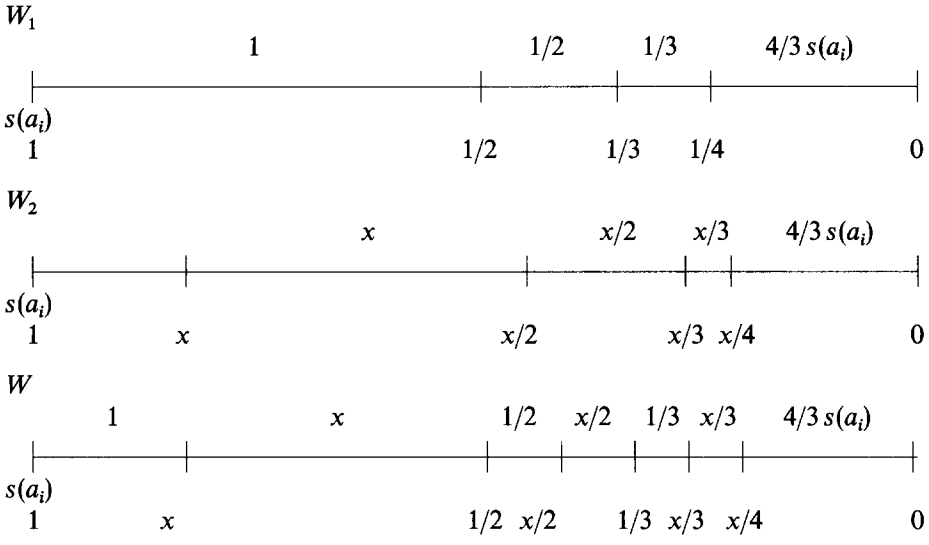
The Best Choice of Bin Sizes

We first assume that we have two bin sizes: $s(B_1) = 1$, and we can choose $s(B_2) = x$. We want to decide which x gives the smallest asymptotic worst-case ratio. For this we shall give a finer estimation of $G_M^{\$}(s(B_i))$ and hence of $W(L)$. Because of $G_M^{\$}(s(B_i)) = (G_M^{\$}(s(B_i))/s(B_i)) \cdot s(B_i)$ and of the procedure used in (6), we shall focus on an estimation of $G_M^{\$}(s(B_i))/s(B_i)$.

We shall solve this problem only for a special M. For example, let $M = 4$. Then, $I_{1,1} = (1/2, 1]$, $I_{1,2} = (1/3, 1/2]$, $I_{1,3} = (1/4, 1/3]$, $I_{\#} = (0, 1/4]$. We shall or-

ganize a case analysis to choose the best x. For this purpose we have to choose x so that the maximum of $G_4^{\$}(1)$ and that of $G_4^{\$}(x)/x$ have the lowest possible values.

Case I. $3/4 < x < 1$.
Here



and hence we have the following intervals and elements:

Interval	Element	Weight W
$(x, 1]$	$I_{1,1}$	1
$(1/2, x]$	$I_{2,1}$	x
$(x/2, 1/2]$	$I_{1,2}$	$1/2$
$(1/3, x/2]$	$I_{2,2}$	$x/2$
$(x/3, 1/3]$	$I_{1,3}$	$1/3$
$(x/4), x/3]$	$I_{2,3}$	$x/3$
$(0, x/4]$	$I_{\#}$ (small)	$(4/3)\, s(a_i)$

It is now clear that we can increase the total weight of items in a bin if we choose an $I_{i,j}$ item as small as possible, and increase the size of small elements. On the other hand, for $I_{1,1}$, $I_{1,2}$, $I_{1,3}$ and $I_{2,3}$ items $W(a_i)/s(a_i) < 4/3$, and hence, if we replace them by small elements then the total weight will increase. Accordingly, we need not take $I_{1,1}$, $I_{1,2}$, $I_{1,3}$ and $I_{2,3}$ elements into account. The case analysis is given in Table 1. From this Table we have to choose x $(3/4 < x < 1)$ so that the maximum of weights W in 1.1.–1.5. and the maximum of weights W/x in 1.6.–1.10. are as small as possible. That is now in case 1.7. at $x = 3/4$, and has the weight W of 13/12, and hence the asymptotic worst-case ratio can be arbitrarily near to $(13/12)/(3/4) = 13/9 \approx 1.44\ldots$ if we choose $x = 3/4 + \delta$ where δ is small.

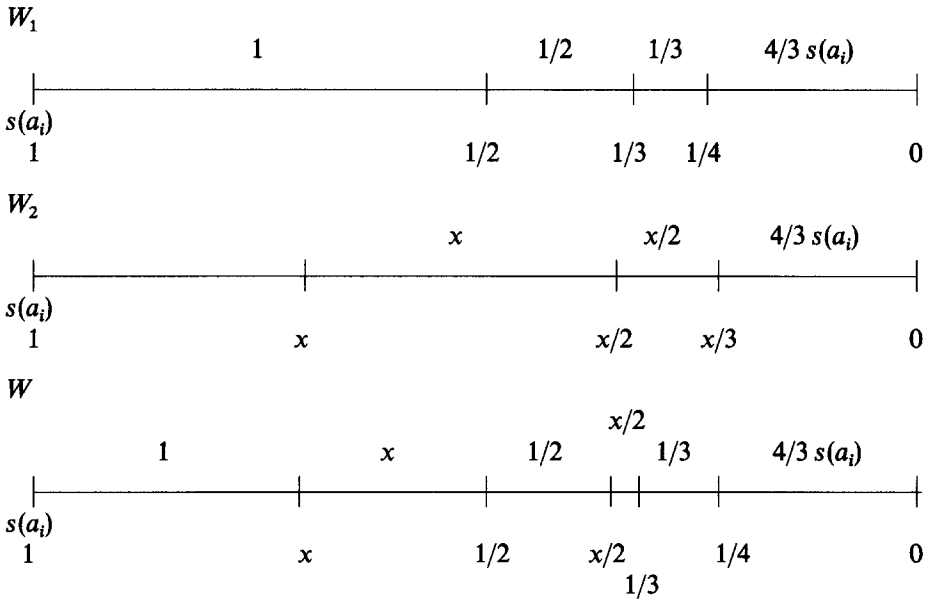
Table 1. The case analysis of possible partitions of $P(1)$ and $P(x)$ at $3/4 < x < 1$

Case	Elements (partitions)	Weight $W \leqq$
Analysis of $P(1)$		
1.1.	$I_{2,1}+I_{2,2}+$small	$3x/2+(4/3)(1-1/2-1/3)=2/9+3x/2$
1.2.	$I_{2,1}+$small	$x+(4/3)(1-1/2)=2/3+x$
1.3.	$I_{2,2}+I_{2,2}+$small	$x+(4/3)(1-1/3-1/3)=4/9+x$
1.4.	$I_{2,2}+$small	$x/2+(4/3)(1-1/3)=8/9+x/2$
1.5.	small	4/3
Analysis of $P(x)$		
1.6.	$I_{2,1}+I_{2,2}+$small	$x+x/2+(4/3)(x-1/2-1/3)=17x/6-10/9$
1.7.	$I_{2,1}+$small	$x+(4/3)(x-1/2)=7x/3-2/3$
1.8.	$I_{2,2}+I_{2,2}+$small	$x/2+x/2+(4/3)(x-1/3-1/3)=7x/3-8/9$
1.9.	$I_{2,2}+$small	$x/2+(4/3)(x-1/3)=11x/6-4/9$
1.10.	small	$(4/3)x$

Note: 1.6. is possible only for $x > 5/6$.

Case II. $2/3 < x \leqq 3/4$.

For the weighting functions:



and thus we have the following intervals and elements:

Interval	Element	Weight W
$(x, 1]$	$I_{1,1}$	1
$(1/2, x]$	$I_{2,1}$	x
$(x/2, 1/2]$	$I_{1,2}$	1/2

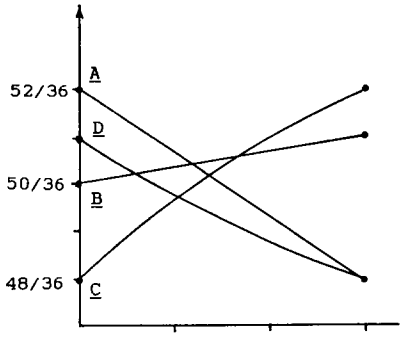


Fig. 1. (A): $7/3-4x/3$; (B) $7/6+x/3$; (C): $7/3-2/(3x)$; (D) $2/3+1/(2x)$

$(1/3, x/2]$	$I_{2,2}$	$x/2$
$(1/4, 1/3]$	$I_{1,3}$	$1/3$
$(0, 1/4]$	I_{*} (small)	$(4/3)\,s(a_i)$ □

In this case, $I_{1,3}$ and $I_{2,2}$ elements have a relative weight $<4/3$, and hence they can be replaced by small elements. We now have the cases given in Table 2.

Table 2. The case analysis of possible partitions of $P(1)$ and $P(x)$ at $2/3<x\leqq 3/4$

Case elements (partitions)		Weight $W\leqq$
Analysis of $P(1)$		
2.1.	$I_{1,1}$ + small	$1+(4/3)(1-x)=7/3-4x/3$
2.2.	$I_{2,1}+I_{1,2}$ + small	$1/2+x+(4/3)(1-1/2-x/2)=7/6+x/3$
2.3.	$I_{2,1}$ + small	$x+(4/3)(1-1/2)=2/3+x$
2.4.	$I_{1,2}+I_{1,2}$ + small	$1+(4/3)(1-x)=7/3-4x/3$
2.5.	$I_{1,2}$ + small	$1/2+(4/3)(1-x/2)=11/6-2x/3$
2.6.	small	$4/3$
Analysis of $P(x)$		
2.7.	$I_{2,1}$ + small	$x+(4/3)(x-1/2)=7x/3-2/3$
2.8.	$I_{1,2}$ + small	$1/2+(4/3)(x-x/2)=2x/3+1/2$
2.9.	small	$(4/3)x$

Here, 2.1. and 2.4. are the same. If now $2/3<x\leqq 3/4$, then 2.3, 2.5. and 2.6. are smaller than 2.2.; thus, from the first part we need only 2.1. and 2.2. For the same interval, in the second part, the relative weight in 2.9. is not greater than in 2.8.; therefore we have to compute here only 2.7. and 2.8.

We now have 4 functions in Case II: (*A*): $7/3-4x/3$; (*B*): $7/6+x/3$; (*C*): $7/3-2/(3x)$; (*D*): $2/3+1/(2x)$, and for $2/3<x\leqq 3/4$ we have to choose that x where the maxima of these functions are as small as possible. We can see these functions in Fig. 1.

The smallest values of the maxima are at the joint point of 2.1. (*A*) and 2.2. (*B*), that is at

$$7/3-4x/3=7/6+x/3,$$

and thus $x=7/10$. This means that the asymptotic worst-case ratio is not greater than $(7-2.8)/3=1.4$.

Case III. $1/2<x\leqq 2/3$.

We consider lists with optimal packing of B_1 bins with $x+\delta$ and small elements. These bins then have a weight W of

$$1+(4/3)(1-x)=7/3-4x/3.$$

This is not smaller in (1/2, 2/3] than $13/9=1.44\ldots$, and in this interval, therefore, the asymptotic worst-case ratio can not be smaller than 13/9.

Case IV. $x\leqq 1/2$.

We consider now lists with optimal packing of B_1 bins with $1/2+\delta$ and small elements. The total weight in these bins is then $\geqq 1+(4/3)(1/2)=5/3$.

We have the following proposition:

Theorem 3. *Suppose there are two bin sizes* ($k=2$). *If the smaller bin size is optimally selected, then the asymptotic worst-case ratio of Variable Harmonic Algorithm is not greater than* 1.4.

We note that this algorithm is very good in space, because we use only at most six (four B_1, and two B_2) bins at the same time.

We can prove that this bound is tight. For this purpose we shall give an infinite series of lists $L_1, L_2, \ldots$ such that

$$\mathrm{OPT}(L_i)=10i+1,$$

and

$$s(B(\mathrm{VH}_M(L_i))=14i.$$

We choose the lists so that both optimal and VH_M use only B_1 bins. Let

$$L_i=(a_{i1}, a_{i2}, \ldots, a_{i(34i)})$$

where

$$s(a_{ij})=\begin{cases} 0.7+\delta_i & \text{if } 1\leqq j\leqq 10i, \\ 0.25-\delta_i & \text{if } j=10i+4k+1, \quad 0\leqq k<3i, \\ & \quad\ j=10i+4k+2, \quad 0\leqq k<3i, \\ & \quad\ j=10i+4k+3, \quad 0\leqq k<3i, \\ 5\delta_i & \text{if } j=10i+4k+4, \quad 0\leqq k\leqq 3i, \\ 0.25-\delta_i & \text{if } j=22i+12k+1, \quad 0\leqq k<i, \\ 0.05 & \text{if } j=22i+12k+l, \quad 0\leqq k<i,\ 2\leqq l\leqq 11, \\ 5\delta_i & \text{if } j=22i+12(k+1), \quad 0\leqq k<i, \end{cases}$$

and $5\delta_i=1/(4i)$.

It is clear that in the optimal packing of L_i we shall have a number $10i$ of B_1 bins containing elements

$$0.7+\delta_i, 0.25-\delta_i, 0.05,$$

and one B_1 bin with a number $4i$ of $5\delta_i$ elements. On the other hand, in VH_M packing we have only B_1 bins, namely $10i$ bins containing a $0.7+\delta_i$ element, $3i$ bins containing $0.25-\delta_i$, $0.25-\delta_i$, $0.25-\delta_i$, $5\delta_i$ elements, i bins containing one $0.25-\delta_i$, ten 0.05, and one $5\delta_i$ elements. Accordingly, our result is the following:

Corollary 2. *Consider variable-sized bin packing with 2 different bin sizes. Let* $s(B_1)=1$, $s(B_2)=0.7$. *Then* $R_{\mathrm{VH}_M}=1.4$.

Open Problems

Variable-sized bin packing is a relatively new field in bin packing. Thus a number of open questions may be asked. We concentrate here only on the "best choice" of bin sizes and list only some of these problems.

a) What is the lower bound for on-line algorithms at the best choice of bin sizes? From the choice of bins in Corollary 2, it is clear that this bound is smaller than those given by Liang and Brown for classical bin packing.

b) Give a best choice of bin sizes for two different bins from a probabilistic point of view, for example if the elements are independently and uniformly distributed random variables on (0, 1).

c) How can these questions (and the results of this paper) be extended to more bin sizes?

References

1. Brown, D.J.: A lower bound for on-line one-dimensional bin packing algorithms. Tech. Rep. No. R-864. Coordinated Sci. Lab., Univ. of Illinois, Urbana, Ill., 1979
2. Friesen, D.K., Langston, M.A.: A storage-size selection problem. Inf. Process. Lett. **18**, 295–296 (1984)
3. Friesen, D.K., Langston, M.A.: Variable sized bin packing. SIAM J. Comput. **15**, 222–230 (1986)
4. Kinnersley, N.G., Langston, M.A.: Online variable-sized bin packing. Discr. Appl. Math. **22**, 143–148 (1988/89)
5. Liang, F.M.: A lower bound for on-line bin packing. Inf. Process. Lett. **10**, 76–79 (1980)
6. Lee, C.C., Lee, D.T.: A simple on-line bin packing algorithm. J. ACM **32**, 562–572 (1985)
7. Murgolo, F.D.: An efficient approximation scheme for variable-sized bin packing. SIAM J. Comput. **16**, 149–161 (1987)

Received July 6, 1988/May 30, 1989