

**CSL** *COORDINATED SCIENCE LABORATORY*

*APPLIED COMPUTATION THEORY GROUP*

## **ON-LINE BIN PACKING IN LINEAR TIME**

PRAKASH RAMANAN  
DONNA J. BROWN

APPROVED FOR PUBLIC RELEASE. DISTRIBUTION UNLIMITED.

REPORT R-1003

UILU-ENG 83-2224

UNIVERSITY OF ILLINOIS – URBANA, ILLINOIS

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle)  ON-LINE BIN PACKING IN LINEAR TIME		5. TYPE OF REPORT & PERIOD COVERED  Technical Report
7. AUTHOR(s)  Prakash Ramanan and Donna J. Brown		6. PERFORMING ORG. REPORT NUMBER ACT-43 UILU-ENG 83-2224, R-1003
9. PERFORMING ORGANIZATION NAME AND ADDRESS Coordinated Science Laboratory University of Illinois at Urbana-Champaign Urbana, IL 61801		8. CONTRACT OR GRANT NUMBER(s) N00014-79-C-0424 IST 80-12240
11. CONTROLLING OFFICE NAME AND ADDRESS Joint Services Electronics Program; National Science Foundation		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE October 1983
		13. NUMBER OF PAGES 25
		15. SECURITY CLASS. (of this report)  UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)  Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)  bin packing, on-line, heuristic algorithm		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) In this paper we study the one-dimensional on-line bin packing problem. A list of pieces, each of size between zero and unity are to be packed, in order of their arrival, into a minimum number of unit-capacity bins. We present a new linear-time algorithm, the Modified Harmonic Algorithm, and show that it has an asymptotic worst-case performance ratio less than $\frac{3}{2} + \frac{1}{9} + \frac{1}{222} = 1.61(561)^*$ . The analysis of the algorithm's performance involves a novel use of weighting functions. In addition, we also show that for a large class of linear-time on-line algorithms,		

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

the performance ratio is at least  $\frac{3}{2} + \frac{1}{9} = 1.61^*$ .

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

## On-line Bin Packing in Linear Time

*Prakash Ramanan*<sup>†</sup>

Department of Computer Science

University of Illinois

Urbana, IL 61801

and

*Donna J. Brown*<sup>††</sup>

Coordinated Science Laboratory

University of Illinois

Urbana, IL 61801

**Abstract.** In this paper we study the one-dimensional on-line bin packing problem. A list of pieces, each of size between zero and unity are to be packed, in order of their arrival, into a minimum number of unit-capacity bins. We present a new linear-time algorithm, the Modified Harmonic Algorithm, and show that it has an asymptotic worst-case performance ratio less than  $\frac{3}{2} + \frac{1}{9} + \frac{1}{222} = 1.61(561)^*$ . The analysis of the algorithm's performance involves a novel use of weighting functions. We also show that for a large class of linear-time on-line algorithms, the performance ratio is at least  $\frac{3}{2} + \frac{1}{9} = 1.61^*$ .

---

<sup>†</sup> This author's work was supported by the Office of Naval Research under grant N00014-79C-0775, the National Science Foundation under grant MCS 81-07647, and an IBM Graduate Fellowship.

<sup>††</sup> This author's work was supported by the National Science Foundation under Grant NSF IST 80-12240 and the Joint Services Electronics Program (U.S. Army, U.S. Navy, and U.S. Air Force) under contract N00014-79-C-0424.

## 1. Introduction

Let  $L = (p_1, p_2, \dots, p_n)$  be a list of pieces with sizes in the interval  $(0, 1]$ . The one-dimensional *bin packing problem* is to pack the pieces into a minimum number of bins in such a way that the sum of the piece sizes in each bin is at most one. As this problem is known to be NP-complete [GJ79, K72], much work has been done in the study of approximation algorithms; a survey of these results is given in [CGJ83].

For any (heuristic) bin packing algorithm  $A$ , let  $A(L)$  denote the number of bins used by algorithm  $A$  in packing list  $L$ , and let  $OPT(L)$  denote the minimum (optimum) number of bins required to pack list  $L$ . We are concerned with the asymptotic worst-case *performance ratio*

$$R_A = \lim_{n \rightarrow \infty} \max_{OPT(L)=n} \frac{A(L)}{OPT(L)}.$$

Thus, we would like to construct an algorithm  $A$  which has a performance ratio close to one. Intuitively, we want an algorithm that minimizes, for large lists, the worst-case percentage of excess bins used compared to an optimal packing.

In this paper we concern ourselves with algorithms for which the pieces in list  $L$  are available one at a time, and each piece must be packed in some bin before the next piece is available; such an algorithm is referred to as *on-line*. Previously known on-line algorithms include the  $O(n)$  Next-Fit ( $NF$ ), and the  $O(n \log n)$  First-Fit ( $FF$ ) [J73, J74, JDUGG74], the  $O(n)$  Harmonic ( $H$ ) [LL83], the  $O(n \log n)$  Refined First-Fit ( $RFF$ ) [Y80], and the  $O(n \log n)$  Doubly-Refined First-Fit ( $DRFF$ ) [B79b]. These algorithms have the following performance ratios:  $R_{NF} = 2$ ,  $R_{FF} = 1.7$ ,  $R_H = 1.692\dots$ ,  $R_{RFF} = 1.6^*$ , and  $R_{DRFF} < 1.64$ . In this paper we present a new linear-time algorithm which we call Modified Harmonic ( $MH$ ) and show that  $R_{MH} < 1.61(561)^*$ .

On the lower bound side, Yao [Y80] showed that for any on-line algorithm, the performance ratio is at least 1.5. This lower bound was further improved to 1.536... independently by Brown [B79a] and Liang [L80].

It should be observed that considerably better performance ratios exist for algorithms which are not on-line. For instance, running the First-Fit Algorithm on pieces that have been ordered by decreasing size gives the First-Fit Decreasing ( $FFD$ ) Algorithm [J73, J74, JDUGG74], for which  $R_{FFD} = \frac{11}{9} = 1.2^*$ .

Friesen and Langston [FrL81] devised a hybrid algorithm with a performance ratio of 1.2, and Garey and Johnson [GJ81] modified First-Fit Decreasing to obtain an algorithm (MFFD) with  $R_{MFFD} = \frac{71}{60} = 1.183^*$ . Fernandez de la Vega and Lueker [FeL81] showed that for every  $\epsilon > 0$ , there is a linear-time algorithm  $A[\epsilon]$  with  $R_{A[\epsilon]} \leq 1 + \epsilon$ . More recently, Karmarkar and Karp [KK82] presented an algorithm that is asymptotically optimal; i.e., has performance ratio one.

In Section 2, we present our Modified Harmonic Algorithm, and describe the packings produced. In Section 3, we use a novel weighting function scheme to analyze the algorithm. In Section 4, we characterize a large class of linear-time on-line algorithms for which the performance ratio is at least  $\frac{3}{2} + \frac{1}{9} = 1.61^*$ , suggesting that it may be difficult to improve on our Modified Harmonic Algorithm if we restrict ourselves to linear time. In Section 5, we summarize our results and make some further observations.

## 2. The Modified Harmonic Algorithm

The Modified Harmonic Algorithm (MH) is based on three previously known on-line algorithms: the Refined First-Fit Algorithm of Yao [Y80], the Next-Fit Algorithm of Johnson [J73], and the Harmonic Algorithm of Lee and Lee [LL83]. Because the latter two algorithms are needed to describe our algorithm, we first briefly describe them.

The Next-Fit Algorithm operates as follows. Initially, the empty bins are indexed as  $bin_1, bin_2, \dots$ . Piece  $p_1$  is packed in  $bin_1$ . Suppose that  $p_1, p_2, \dots, p_{i-1}$  have been packed, and  $p_i$  is the next piece to be packed. Let  $j$  be the largest index such that  $bin_j$  is nonempty. If  $p_i$  will fit in  $bin_j$ , then  $p_i$  is packed in  $bin_j$ ; otherwise  $p_i$  is packed in  $bin_{j+1}$ .

The Harmonic Algorithm is based on the harmonic partition of the interval  $(0, 1]$ :

$$(0, 1] = \bigcup_{j=1}^k I_j, \text{ where } I_j = (1/(j+1), 1/j], 1 \leq j < k, \text{ and } I_k = (0, 1/k],$$

for some  $k > 1$ . A piece  $p$  is called an  $I_j$ -piece if  $p \in I_j$ ,  $1 \leq j \leq k$ . Initially, the set of empty bins is divided into  $k$  infinite classes: bins of type  $B_j$ ,  $1 \leq j \leq k$ . A bin of type  $B_j$  is used to pack only  $I_j$ -pieces. Note that  $j$   $I_j$ -pieces can be packed in a bin of type  $B_j$  for  $1 \leq j < k$ . Suppose that

$p_1, p_2, \dots, p_{i-1}$  have been packed, and  $p_i$  is the next piece to be packed. If  $p_i$  is an  $I_j$ -piece for some  $j$ ,  $1 \leq j < k$ , then it is packed in a nonempty bin of type  $B_j$  that contains fewer than  $j$  pieces, if one exists; if no such bin exists, then  $p_i$  is packed in an empty bin of type  $B_j$ . If  $p_i$  is an  $I_k$ -piece, then it is packed in bins of type  $B_k$  by Next-Fit.

Now we are ready to describe the Modified Harmonic Algorithm. The algorithm is based on the following partition of the interval  $(0, 1]$ :

$$(0, 1] = I_1 \cup I_2 \cup \bigcup_{j=1}^k I_j,$$

where  $I_1 = (1 - y, 1]$ ,  $I_2 = (y, 1/2]$ ,  $I_1 = (1/2, 1 - y]$ ,  $I_2 = (1/3, y]$ ,

$I_j = (1/(j + 1), 1/j]$ ,  $3 \leq j < k$ , and  $I_k = (0, 1/k]$ ,

for some  $y$  and  $k$ ,  $1/3 < y < 1/2$  and  $k \geq 3$  (exact values for  $y$  and  $k$  will be given in Section 3). A piece  $p$  is called an

$I_1$ -piece if  $p \in I_1$ ,

$I_2$ -piece if  $p \in I_2$ ,

$I_j$ -piece if  $p \in I_j$  for some  $j$ ,  $1 \leq j \leq k$ .

Initially, the set of empty bins is divided into  $k + 2$  infinite classes: bins of type  $\bar{B}_1$ ,  $\bar{B}_2$ , and  $B_j$ ,  $1 \leq j \leq k$ . A bin of type  $\bar{B}_1$  is used to pack only  $I_1$ -pieces, a bin of type  $\bar{B}_2$  is used to pack only  $I_2$ -pieces, and a bin of type  $B_j$ ,  $2 \leq j \leq k$ , is used to pack only  $I_j$ -pieces. All  $I_1$ -pieces are packed in bins of type  $B_1$ . In addition, some of the  $I_\rho$ -pieces for  $2 \leq \rho \leq 3$  or  $6 \leq \rho \leq k - 2$ , are also packed in bins of type  $B_1$ . In particular, for each  $\rho$ , a fixed fraction  $\frac{1}{m_\rho}$  (values for the  $m_\rho$ 's will be given in Section 3) of the  $I_\rho$ -pieces are packed in bins of type  $B_1$ ; if there are a sufficient number of  $I_1$ -pieces, then each bin of type  $B_1$  will also contain an  $I_1$ -piece. Thus, each nonempty bin of type  $B_1$  will contain an  $I_1$ -piece and/or  $I_\rho$ -pieces (at most  $\max(1, \lfloor \rho y \rfloor)$  of them) for one  $\rho$ ,  $2 \leq \rho \leq 3$  or  $6 \leq \rho \leq k - 2$ . The algorithm packs a list in such a way that, at any stage in the packing:

- (1) each nonempty bin of type  $\bar{B}_1$  contains 1  $I_1$ -piece,
- (2) each nonempty bin of type  $\bar{B}_2$  (except possibly the last one) contains 2  $I_2$ -pieces,
- (3) each nonempty bin of type  $B_j$ ,  $2 \leq j < k$ , (except possibly the last one) contains  $j$   $I_j$ -pieces,
- (4) each nonempty bin of type  $B_k$  (except possibly the last one) is at least  $\frac{k-1}{k}$  full,

and

(5) each nonempty bin of type  $B_1$

- (i) contains only an  $I_1$ -piece,
- (ii) contains only an  $I_2$ -piece or an  $I_3$ -piece,
- (iii) contains an  $I_1$ -piece together with an  $I_2$ -piece or an  $I_3$ -piece,
- (iv) contains  $\lfloor \rho y \rfloor$   $I_\rho$ -pieces for some  $\rho$ ,  $6 \leq \rho \leq k - 2$ ,
- (v) contains an  $I_1$ -piece, and  $\lfloor \rho y \rfloor$   $I_\rho$ -pieces for some  $\rho$ ,  $6 \leq \rho \leq k - 2$ ,
- (vi) contains at least one, and at most  $\lfloor \rho y \rfloor - 1$   $I_\rho$ -pieces for some  $\rho$ ,  $6 \leq \rho \leq k - 2$ ,

or

- (vii) contains an  $I_1$ -piece together with at least one, and at most  $\lfloor \rho y \rfloor - 1$   $I_\rho$ -pieces for some  $\rho$ ,  $6 \leq \rho \leq k - 2$ .

Moreover, if there is a bin as in (i), then there can be no bins as in (ii), (iv), or (vi). Also, for each  $\rho$ ,  $6 \leq \rho \leq k - 2$ , the number of bins as in (vi) plus the number of bins as in (vii) is at most one.

When we say *harmonic pack*  $(p_i, I_j)$ , where  $p_i$  is an  $I_j$ -piece,  $2 \leq j < k$ , we mean:

- if there exists a nonempty bin of type  $B_j$  containing fewer than  $j$   $I_j$ -pieces
- then pack  $p_i$  in that bin
- else pack  $p_i$  in an empty bin of type  $B_j$ .

We now give a precise statement of our algorithm.

### Modified Harmonic Algorithm

for  $i := 1$  to  $n$  do

begin

case  $p_i$  in

$I_1$  : pack  $p_i$  in an empty bin of type  $\bar{B}_1$

$I_2$  : if there exists a bin of type  $\bar{B}_2$  containing only one  $I_2$ -piece

then pack  $p_i$  in that bin

else pack  $p_i$  in an empty bin of type  $\bar{B}_2$

$I_j$  ( $j = 4, 5$ , or  $k - 1$ ) : harmonic pack ( $p_i, I_j$ )

$I_k$  : pack  $p_i$  in bins of type  $B_k$  by Next-Fit

$I_1$  : if there exists a nonempty bin of type  $B_1$  that does not contain an  $I_1$ -piece

then pack  $p_i$  in that bin

else pack  $p_i$  in an empty bin of type  $B_1$

$I_\rho$  ( $2 \leq \rho \leq 3$ ) : if  $p_i$  is the  $(m_\rho r)^{\text{th}}$   $I_\rho$ -piece to arrive thus far for some integer  $r \geq 1$

then if there exists a bin of type  $B_1$  containing only an  $I_1$ -piece

then pack  $p_i$  in that bin

else pack  $p_i$  in an empty bin of type  $B_1$

else harmonic pack ( $p_i, I_\rho$ )

$I_\rho$  ( $6 \leq \rho \leq k - 2$ ) : if  $p_i$  is the  $(m_\rho r)^{\text{th}}$   $I_\rho$ -piece to arrive thus far for some integer  $r \geq 1$

then if there exists a bin of type  $B_1$  containing at least one, and at most

$\lfloor \rho y \rfloor - 1$   $I_\rho$ -pieces

then pack  $p_i$  in that bin

else if there exists a bin of type  $B_1$  containing only an  $I_1$ -piece

then pack  $p_i$  in that bin

else pack  $p_i$  in an empty bin of type  $B_1$

else harmonic pack ( $p_i, I_\rho$ )

end.

It is easy to see that the Modified Harmonic Algorithm runs in linear time and uses linear space. Note that in the above algorithm, we had implicitly assumed that the  $m_p$ 's are integers. The values we will be specifying in Section 3 will not be integers. In this case we require that at any stage if  $a_p$  is the number of  $I_p$ -pieces that have been packed, then  $\left\lfloor \frac{a_p}{m_p} \right\rfloor$  of them have been packed in bins of type  $B_1$ . It is easy to modify the algorithm accordingly.

### 3. Analysis of the Algorithm

In this section, we use weighting functions to analyze the performance of the Modified Harmonic Algorithm. Throughout this section we shall be considering only the nonempty, i.e. packed, bins. The weight of a bin is defined to be the sum of the weights of all the pieces in the bin. We shall assign weights to pieces in such a way that the average weight of all but a constant number of bins packed by our algorithm is at least one.

Let  $w_1$  and  $w_2$  be the weights of an  $I_1$ -piece and an  $I_2$ -piece, respectively. Let  $w_j$  be the weight of an  $I_j$ -piece,  $1 \leq j < k$ , and let  $w_k(p)$  be the weight of an  $I_k$ -piece  $p$ .

Since a bin can contain one  $I_1$ -piece, two  $I_2$ -pieces, four  $I_4$ -pieces, five  $I_5$ -pieces, or  $k-1$   $I_{k-1}$ -pieces, we have

$$w_1 = 1,$$

$$w_2 = \frac{1}{2},$$

$$w_4 = \frac{1}{4},$$

$$w_5 = \frac{1}{5},$$

$$w_{k-1} = \frac{1}{k-1}.$$

Since a bin of type  $B_k$  will be at least  $\frac{k-1}{k}$  full, we assign to an  $I_k$ -piece  $p$  the weight

$$w_k(p) = \frac{k}{k-1} p.$$

Interval containing piece $s$	weight of piece $s$ for	
	$\alpha > 0$	$\alpha = 0$
$I_1 : (1 - y, 1]$	1	1
$I_1 : (1/2, 1 - y]$	1	0
$I_2 : (y, 1/2]$	$\frac{1}{2}$	$\frac{1}{2}$
$I_2 : (1/3, y]$	$\frac{1}{2} - \frac{1}{2m_2}$	$\frac{1}{2} + \frac{1}{2m_2}$
$I_3 : (1/4, 1/3]$	$\frac{1}{3} - \frac{1}{3m_3}$	$\frac{1}{3} + \frac{2}{3m_3}$
$I_4 : (1/5, 1/4]$	$\frac{1}{4}$	$\frac{1}{4}$
$I_5 : (1/6, 1/5]$	$\frac{1}{5}$	$\frac{1}{5}$
$I_6 : (1/7, 1/6]$	$\frac{1}{6} - \frac{1}{6m_6}$	$\frac{1}{6} - \frac{1}{6m_6} + \frac{1}{m_6[6y]}$
$\vdots$	$\vdots$	$\vdots$
$I_\rho : (1/(\rho + 1), 1/\rho]$	$\frac{1}{\rho} - \frac{1}{\rho m_\rho}$	$\frac{1}{\rho} - \frac{1}{\rho m_\rho} + \frac{1}{m_\rho[\rho y]}$
$\vdots$	$\vdots$	$\vdots$
$I_{k-2} : (1/(k - 1), 1/(k - 2)]$	$\frac{1}{k - 2} - \frac{1}{(k - 2)m_{k-2}}$	$\frac{1}{k - 2} - \frac{1}{(k - 2)m_{k-2}} + \frac{1}{m_{k-2}[(k - 2)y]}$
$I_{k-1} : (1/k, 1/(k - 1)]$	$\frac{1}{k - 1}$	$\frac{1}{k - 1}$
$I_k : (0, 1/k]$	$\frac{k}{k - 1} s$	$\frac{k}{k - 1} s$

Table I. Weighting functions for  $\alpha > 0$  and  $\alpha = 0$ .

Recall that, for  $2 \leq \rho \leq 3$  or  $6 \leq \rho \leq k-2$ , some of the  $I_\rho$ -pieces are packed in bins of type  $B_1$ . The weights we assign to the  $I_1$ -pieces and  $I_\rho$ -pieces depend on the input list  $L$ . Let  $\alpha$  be the number of bins in the packing of  $L$  produced by the Modified Harmonic Algorithm that contain only an  $I_1$ -piece. The weights we assign to the  $I_1$ -pieces and  $I_\rho$ -pieces depend on whether  $\alpha > 0$  or  $\alpha = 0$ .

For  $\alpha > 0$ , there are bins containing only an  $I_1$ -piece; moreover, every bin of type  $B_1$  contains an  $I_1$ -piece. We assign weights to the  $I_1$ -pieces and  $I_\rho$ -pieces in such a way that the average weight of all bins of types  $B_1$  and  $B_\rho$  is one:

$$w_1 = 1,$$

$$w_\rho = \frac{1}{\rho} \left( 1 - \frac{1}{m_\rho} \right) \text{ for } 2 \leq \rho \leq 3 \text{ or } 6 \leq \rho \leq k-2.$$

For  $\alpha = 0$ , not all the bins of type  $B_1$  contain an  $I_1$ -piece. Again, we assign weights to the  $I_1$ -pieces and  $I_\rho$ -pieces in such a way that the average weight of all bins of types  $B_1$  and  $B_\rho$  is one:

$$w_1 = 0,$$

$$w_\rho = \frac{1}{\rho} \left( 1 - \frac{1}{m_\rho} \right) + \frac{1}{m_\rho} \text{ for } 2 \leq \rho \leq 3,$$

$$w_\rho = \frac{1}{\rho} \left( 1 - \frac{1}{m_\rho} \right) + \frac{1}{m_\rho \lfloor \rho y \rfloor} \text{ for } 6 \leq \rho \leq k-2.$$

Table I summarizes the above weighting functions.

We make use of these weighting functions to analyze our algorithm's performance. Letting  $W(L)$  be the sum of the weights of all the pieces in list  $L$ , we show in Lemma 1 that

$$MH(L) < W(L) + 2k - 7,$$

and show in Lemma 2 that

$$W(L) < \left( \frac{3}{2} + \frac{1}{9} + \frac{1}{222} \right) OPT(L).$$

Combining these results gives us the following bound on our algorithm's performance:

$$R_{MH} < \frac{3}{2} + \frac{1}{9} + \frac{1}{222}.$$

In Lemma 3, we prove that this bound is essentially tight. The results of these three lemmas are

combined to give Theorem 1.

**Lemma 1.** For any list  $L$ ,  $MH(L) < W(L) + 2k - 7$ .

**Proof.** Let  $\bar{b}_1$  and  $\bar{b}_2$  be the number of bins of type  $\bar{B}_1$  and  $\bar{B}_2$ , respectively, and let  $b_j$  be the number of bins of type  $B_j$ ,  $1 \leq j \leq k$ . It is clear that

$$MH(L) = \bar{b}_1 + \bar{b}_2 + \sum_{j=1}^k b_j.$$

Let  $\bar{W}_1$  and  $\bar{W}_2$  be the sum of the weights of all the  $I_1$ -pieces and  $I_2$ -pieces, respectively, and let  $W_j$  be the sum of the weights of all the  $I_j$ -pieces,  $1 \leq j \leq k$ . It is clear that

$$W(L) = \bar{W}_1 + \bar{W}_2 + \sum_{j=1}^k W_j.$$

It is easy to see that we have constructed the weighting functions in such a way that each bin (except possibly the last one) of type  $\bar{B}_1$ ,  $\bar{B}_2$ ,  $B_4$ ,  $B_5$ , or  $B_{k-1}$ , has weight precisely one. So

$$\bar{b}_1 = \bar{W}_1,$$

$$\bar{b}_2 < \bar{W}_2 + 1,$$

$$b_4 < W_4 + 1,$$

$$b_5 < W_5 + 1,$$

$$b_{k-1} < W_{k-1} + 1.$$

Since each bin of type  $B_k$  (except possibly the last one) must be at least  $\frac{k-1}{k}$  full, its total weight is at least one, and so

$$b_k < W_k + 1.$$

Let  $a_j$  be the number of  $I_j$ -pieces,  $1 \leq j \leq k$ . To analyze the weights of bins of types  $B_1$  and  $B_\rho$ ,  $2 \leq \rho \leq 3$  or  $6 \leq \rho \leq k-2$ , we consider separately the two weighting functions, and show that the lemma holds both for  $\alpha > 0$  and  $\alpha = 0$ .

*Case 1.*  $\alpha > 0$ . Every bin of type  $B_1$  has an  $I_1$ -piece, and so

$$b_1 = W_1.$$

For  $2 \leq \rho \leq 3$  or  $6 \leq \rho \leq k-2$ , not all of the  $I_\rho$ -pieces are packed in bins of type  $B_\rho$ . In particular,

we have

$$\begin{aligned} b_\rho &= \left\lceil \frac{1}{\rho} \left( a_\rho - \left\lfloor \frac{a_\rho}{m_\rho} \right\rfloor \right) \right\rceil \\ &< a_\rho \left( \frac{1}{\rho} - \frac{1}{\rho m_\rho} \right) + 1 \\ &= W_\rho + 1. \end{aligned}$$

Summing all the inequalities for Case 1, we get

$$MH(L) < W(L) + k.$$

*Case 2.*  $\alpha = 0$ . A bin of type  $B_1$  containing an  $I_\rho$ -piece for some  $\rho$ ,  $2 \leq \rho \leq 3$  or  $6 \leq \rho \leq k-2$ , is called a bin of type  $B_{1,\rho}$ . Since  $\alpha = 0$ , every bin of type  $B_1$  is of one of the  $B_{1,\rho}$  types. Let  $b_{1,\rho}$  be the number of bins of type  $B_{1,\rho}$ . Clearly  $b_1 = \sum_\rho b_{1,\rho}$ . Noting that for  $\alpha = 0$  the weight of an  $I_1$ -piece is 0, the weights of the  $I_\rho$ -pieces must "compensate". For  $2 \leq \rho \leq 3$ , we have

$$\begin{aligned} b_\rho + b_{1,\rho} &= \left\lceil \frac{1}{\rho} \left( a_\rho - \left\lfloor \frac{a_\rho}{m_\rho} \right\rfloor \right) \right\rceil + \left\lfloor \frac{a_\rho}{m_\rho} \right\rfloor \\ &< \frac{a_\rho}{\rho} \left( 1 - \frac{1}{m_\rho} \right) + \frac{a_\rho}{m_\rho} + 1 \\ &= a_\rho \left( \frac{1}{\rho} + \frac{\rho-1}{\rho m_\rho} \right) + 1 \\ &= W_\rho + 1. \end{aligned}$$

For  $6 \leq \rho \leq k-2$ , each bin (except possibly the last one) of type  $B_{1,\rho}$  contains  $\lfloor \rho y \rfloor$   $I_\rho$ -pieces. As above, we have

$$\begin{aligned} b_\rho + b_{1,\rho} &= \left\lceil \frac{1}{\rho} \left( a_\rho - \left\lfloor \frac{a_\rho}{m_\rho} \right\rfloor \right) \right\rceil + \left\lfloor \frac{1}{\lfloor \rho y \rfloor} \left\lfloor \frac{a_\rho}{m_\rho} \right\rfloor \right\rfloor \\ &< \frac{a_\rho}{\rho} \left( 1 - \frac{1}{m_\rho} \right) + \frac{a_\rho}{m_\rho \lfloor \rho y \rfloor} + 2 \\ &= a_\rho \left( \frac{1}{\rho} - \frac{1}{\rho m_\rho} + \frac{1}{m_\rho \lfloor \rho y \rfloor} \right) + 2 \end{aligned}$$

$$= W_\rho + 2.$$

Summing all the inequalities for Case 2, we get

$$MH(L) < W(L) + 2k - 7. \quad \square$$

**Lemma 2.** Consider the weighting functions specified in Table I for  $\alpha > 0$  and  $\alpha = 0$ . Let  $y = \frac{265}{684}$ ,

$k = 38$ ,  $m_2 = 9$ ,  $m_3 = 12$ , and  $m_\rho = \frac{(k-1)(\rho+1)}{k-\rho-1} = \frac{37(\rho+1)}{37-\rho}$  for  $6 \leq \rho \leq k-2$  (see Table II).

Then

$$W(L) < \left( \frac{3}{2} + \frac{1}{9} + \frac{1}{222} \right) OPT(L).$$

**Proof.** Consider a bin  $B$  in the optimal packing, and let  $s_1 \geq s_2 \geq \dots \geq s_t$  be the pieces packed in it,  $s_1 + s_2 + \dots + s_t \leq 1$ . Let  $w(s_i)$  be the weight of a piece  $s_i$ , and let  $w(B)$  be the weight of  $B$ .

It is clear that

$$w(B) = w(s_1) + w(s_2) + \dots + w(s_t).$$

We shall prove that

$$w(B) < \frac{3}{2} + \frac{1}{9} + \frac{1}{222},$$

both for  $\alpha > 0$  and  $\alpha = 0$ .

The proof is done by cases, depending on the sizes of some of the largest pieces in  $B$ . Tables III and IV summarize the cases for  $\alpha > 0$  and  $\alpha = 0$ , respectively. Columns  $s_1$ ,  $s_2$ , and  $s_3$  indicate the intervals containing the first, second, and third largest pieces in  $B$ . For convenience, we let  $I_j$  ( $j \geq r$ ) denote one of the intervals  $I_r, I_{r+1}, \dots, I_k$ , or the "interval"  $[0]$ ; i.e., no piece at all. Column  $\bar{s}$  gives an upper bound on the remaining space left in the bin after packing the pieces of sizes specified in columns  $s_1$ ,  $s_2$ , and  $s_3$ .

To determine an upper bound on  $w(B)$ , we find it useful to compute upper bounds on  $\frac{w(s)}{s}$  for each possible piece size  $s$  (see Table II). In particular, we shall make use of the fact that

$$\sum_{j=i}^t w(s_j) \leq \sum_{j=i}^t s_j \left( \max_{i \leq r \leq t} \frac{w(s_r)}{s_r} \right).$$

Interval containing piece $s$	$\alpha > 0$		$\alpha = 0$	
	$w(s)$	$\frac{w(s)}{s}$ (upper bound)	$w(s)$	$\frac{w(s)}{s}$ (upper bound)
$T_1: (1 - y, 1]$	1	$\frac{684}{419}$	1	$\frac{684}{419}$
$I_1: (1/2, 1 - y]$	1	2	0	0
$T_2: (y, 1/2]$	$\frac{1}{2}$	$\frac{342}{265}$	$\frac{1}{2}$	$\frac{342}{265}$
$I_2: (1/3, y]$	$\frac{4}{9}$	$\frac{4}{3}$	$\frac{5}{9}$	$\frac{5}{3}$
$I_3: (1/4, 1/3]$	$\frac{11}{36}$	$\frac{11}{9}$	$\frac{7}{18}$	$\frac{14}{9}$
$I_4: (1/5, 1/4]$	$\frac{1}{4}$	$\frac{5}{4}$	$\frac{1}{4}$	$\frac{5}{4}$
$I_5: (1/6, 1/5]$	$\frac{1}{5}$	$\frac{6}{5}$	$\frac{1}{5}$	$\frac{6}{5}$
$I_6: (1/7, 1/6]$	$\frac{38}{7 \cdot 37}$	$\frac{38}{37}$	$\frac{107}{14 \cdot 37}$	$\frac{107}{2 \cdot 37}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$I_{12}: (1/13, 1/12]$	$\frac{38}{13 \cdot 37}$	$\frac{38}{37}$	$\frac{177}{52 \cdot 37}$	$\frac{177}{4 \cdot 37}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$I_{18}: (1/19, 1/18]$	$\frac{38}{19 \cdot 37}$	$\frac{38}{37}$	$\frac{13}{6 \cdot 37}$	$\frac{247}{6 \cdot 37}$
$I_{19}: (1/20, 1/19]$	$\frac{38}{20 \cdot 37}$	$\frac{38}{37}$	$\frac{71}{35 \cdot 37}$	$\frac{284}{7 \cdot 37}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$I_{k-2}: (1/(k-1), 1/(k-2)]$	$\frac{38}{37 \cdot 37}$	$\frac{38}{37}$	$\frac{495}{481 \cdot 37}$	$\frac{495}{481}$
$I_{k-1}: (1/k, 1/(k-1)]$	$\frac{1}{37}$	$\frac{38}{37}$	$\frac{1}{37}$	$\frac{38}{37}$
$I_k: (0, 1/k]$	$\frac{38}{37} s$	$\frac{38}{37}$	$\frac{38}{37} s$	$\frac{38}{37}$

Note: For  $\alpha = 0$ ,  $s \in I_j$ , and  $s' \in I_{j'}$ ,  $6 \leq j \leq j' \leq k$ , upper bound on  $\frac{w(s)}{s} \geq$  upper bound on  $\frac{w(s')}{s'}$ .

Table II. Weight summary for all pieces  $s$ , when  $y = \frac{265}{684}$ ,  $k = 38$ ,  $m_2 = 9$ ,  $m_3 = 12$ , and

$$m_\rho = \frac{(k-1)(\rho+1)}{k-\rho-1} = \frac{37(\rho+1)}{37-\rho} \text{ for } 6 \leq \rho \leq k-2.$$

$s_1$	$s_2$	$s_3$	$\bar{s}$	$w(B)$ $w(s_1) + w(s_2) + \dots + w(s_t)$ (upper bound)
$I_1$	$\bar{I}_2$	-	$\frac{1}{2} - y = \frac{77}{684} < \frac{1}{6}$	$1 + \frac{1}{2} + \frac{77}{684} * \frac{38}{37}$
$I_1$ or $\bar{I}_1$	$I_2$	-	$\frac{1}{6}$	$1 + \frac{4}{9} + \frac{1}{6} * \frac{38}{37}$
$I_1$ or $\bar{I}_1$	$I_3$	$I_4$	$\frac{1}{20}$	$1 + \frac{11}{36} + \frac{1}{4} + \frac{1}{20} * \frac{38}{37}$
$I_1$ or $\bar{I}_1$	$I_3$	$I_j (j \geq 5)$	$\frac{1}{4}$	$1 + \frac{11}{36} + \frac{1}{4} * \frac{6}{5}$
$I_1$ or $\bar{I}_1$	$I_4$	$I_4$	$\frac{1}{10}$	$1 + \frac{1}{4} + \frac{1}{4} + \frac{1}{10} * \frac{38}{37}$
$I_1$ or $\bar{I}_1$	$I_4$	$I_j (j \geq 5)$	$\frac{3}{10}$	$1 + \frac{1}{4} + \frac{3}{10} * \frac{6}{5}$
$I_1$ or $\bar{I}_1$	$I_j (j \geq 5)$	-	$\frac{1}{2}$	$1 + \frac{1}{2} * \frac{6}{5}$
$\bar{I}_2$ or $I_j (j \geq 2)$	-	-	1	$1 * \frac{4}{3}$

Table III. List of cases proving  $w(B) < \frac{3}{2} + \frac{1}{9} + \frac{1}{222}$  for  $\alpha > 0$ .

$s_1$	$s_2$	$s_3$	$\bar{s}$	$w(B)$ $w(s_1) + w(s_2) + \dots + w(s_t)$ (upper bound)
$I_1$	$I_2$	$I_{18}$	$y - \frac{1}{3} - \frac{1}{19} = \frac{1}{684}$	$1 + \frac{5}{9} + \frac{13}{6 \cdot 37} + \frac{1}{684} \cdot \frac{38}{37}$
$I_1$	$I_2$	$I_j (j \geq 19)$	$y - \frac{1}{3} = \frac{37}{684}$	$1 + \frac{5}{9} + \frac{37}{684} \cdot \frac{284}{7 \cdot 37}$
$I_1$	$I_j (j \geq 3)$	-	$y = \frac{265}{684}$	$1 + \frac{265}{684} \cdot \frac{14}{9}$
$I_1$	-	-	$\frac{1}{2}$	$\frac{1}{2} \cdot \frac{5}{3}$
$I_2$	$I_2$	-	$1 - 2y = \frac{154}{684} < \frac{1}{4}$	$\frac{1}{2} + \frac{1}{2} + \frac{154}{684} \cdot \frac{107}{2 \cdot 37}$
$I_2$ or $I_2$	$I_2$	$I_3$	$\frac{1}{12}$	$\frac{5}{9} + \frac{5}{9} + \frac{7}{18} + \frac{1}{12} \cdot \frac{177}{4 \cdot 37}$
$I_2$ or $I_2$	$I_2$	$I_j (j \geq 4)$	$\frac{1}{3}$	$\frac{5}{9} + \frac{5}{9} + \frac{1}{3} \cdot \frac{107}{2 \cdot 37}$
$I_2$ or $I_2$	$I_j (j \geq 3)$	-	$\frac{2}{3}$	$\frac{5}{9} + \frac{2}{3} \cdot \frac{14}{9}$
$I_j (j \geq 3)$	-	-	1	$1 \cdot \frac{14}{9}$

Table IV. List of cases proving  $w(B) < \frac{3}{2} + \frac{1}{9} + \frac{1}{222}$  for  $\alpha = 0$ .

Column  $w(B)$  in tables III and IV specifies an upper bound on  $w(s_1) + w(s_2) + \dots + w(s_t)$ , calculated by taking  $w(s_1)$  if  $s_1$  is specified, plus  $w(s_2)$  if  $s_2$  is specified, plus  $w(s_3)$  if  $s_3$  is specified, plus

$$\bar{s} \max_{s \leq \bar{s}} \frac{w(s)}{s}.$$

For example, consider the case for  $\alpha > 0$ , with  $s_1 \in I_1$  and  $s_2 \in I_2$ . Then

$$\bar{s} = 1 - (s_1 + s_2) < 1 - (1/2 + y) = 1/2 - y = \frac{77}{684} < 1/6.$$

Thus, we know that each of  $s_3, \dots, s_t$  must be smaller than  $1/6$ , and so (from Table II)  $\frac{w(s_i)}{s_i} \leq \frac{38}{37}$  for

$3 \leq i \leq t$ . This gives

$$\begin{aligned} w(B) &= w(s_1) + w(s_2) + \sum_{i=3}^t w(s_i) \\ &< 1 + \frac{1}{2} + \frac{77}{684} * \frac{38}{37} \\ &= \frac{3}{2} + \frac{1}{9} + \frac{1}{222}. \end{aligned}$$

As another example, consider the case for  $\alpha > 0$ , with  $s_1 \in I_1$  and  $s_2 \in I_j$  ( $j \geq 5$ ). Then  $s_1 > 1/2$ , and  $\bar{s} = 1 - s_1 < 1/2$ . The  $\bar{s}$  portion of the bin has no piece larger than  $1/5$ , and so  $\frac{w(s)}{s} \leq \frac{6}{5}$  for any piece in  $B$  other than  $s_1$ . This gives

$$w(B) = w(s_1) + \sum_{i=2}^t w(s_i) < 1 + \frac{1}{2} * \frac{6}{5} < \frac{3}{2} + \frac{1}{9}.$$

As another example, consider the case for  $\alpha = 0$ , with  $s_1 \in I_2$ ,  $s_2 \in I_2$ , and  $s_3 \in I_3$ . Then

$$\bar{s} = 1 - (s_1 + s_2 + s_3) < 1 - (1/3 + 1/3 + 1/4) = 1/12.$$

Thus, we know that each of  $s_4, \dots, s_t$  must be smaller than  $1/12$ , and so  $\frac{w(s_i)}{s_i} \leq \frac{177}{4 * 37}$  for  $4 \leq i \leq t$ .

This gives

$$\begin{aligned} w(B) &= w(s_1) + w(s_2) + w(s_3) + \sum_{i=4}^t w(s_i) \\ &< \frac{5}{9} + \frac{5}{9} + \frac{7}{18} + \frac{1}{12} * \frac{177}{4 * 37} \\ &< \frac{3}{2} + \frac{1}{9}. \end{aligned}$$

Tables III and IV enumerate all possible cases, and it is easily checked that the values in the  $w(B)$  column are all at most  $\frac{3}{2} + \frac{1}{9} + \frac{1}{222}$ . Thus, the result of the lemma follows.  $\square$

Lemmas 1 and 2 guarantee that the performance ratio of  $MH$  is less than  $\frac{3}{2} + \frac{1}{9} + \frac{1}{222}$ . We now show that this bound is essentially tight.

**Lemma 3.** Let  $y = \frac{265}{684}$ ,  $k = 38$ ,  $m_2 = 9$ ,  $m_3 = 12$ , and  $m_\rho = \frac{(k-1)(\rho+1)}{k-\rho-1} = \frac{37(\rho+1)}{37-\rho}$  for  $6 \leq \rho \leq k-2$ . Then

$$R_{MH} \geq \frac{3}{2} + \frac{1}{9} + \frac{1}{222} - \frac{1}{987,012}.$$

**Proof.** We shall exhibit lists  $L$  with arbitrarily large  $OPT(L)$  such that

$$\frac{MH(L)}{OPT(L)} = \frac{3}{2} + \frac{1}{9} + \frac{1}{222} - \frac{1}{987,012}.$$

Let  $n$  be a multiple of 24,675,300, and let

$$L_1 = (p_1, p_2, \dots, p_n),$$

$$L_2 = (q_1, q_2, \dots, q_n),$$

$$L_3 = (t_1, t_2, \dots, t_n),$$

$$L_4 = (u_1, u_2, \dots, u_{2n}),$$

and

$$L_5 = (v_1, v_2, \dots, v_n),$$

where

$$p_i = \frac{1}{2} + \epsilon,$$

$$q_i = y + \epsilon,$$

$$t_i = \frac{1}{26} + \epsilon,$$

$$u_i = \frac{1}{27} + \epsilon,$$

and

$$v_i = \frac{1}{26,676} - 5\epsilon,$$

for some  $\epsilon$ ,  $0 < \epsilon < 10^{-10}$ . Let  $L$  be the list obtained by concatenating these lists; i.e.,

$$L = L_1 L_2 L_3 L_4 L_5.$$

Note that  $p_i + q_i + t_i + u_i + v_i = 1$ . Hence, the packing in which each bin contains a  $p_i$  piece, a  $q_i$  piece, a  $t_i$  piece, two  $u_i$  pieces, and a  $v_i$  piece is an optimal packing, and so  $OPT(L) = n$ .

Now consider the packing produced by our algorithm. Note that  $m_{25} = \frac{481}{6}$  and  $m_{26} = \frac{999}{11}$ . Since  $\lfloor 25y \rfloor = 9$  and  $\lfloor 26y \rfloor = 10$ , 9  $t_i$  pieces or 10  $u_i$  pieces can be packed in the same bin with a  $p_i$  piece. Thus, the packing produced by our algorithm consists of

$$\begin{aligned} n - \frac{n}{9m_{25}} - \frac{2n}{10m_{26}} &= \frac{64,702}{64,935} n \text{ bins, each containing 1 } p_i \text{ piece,} \\ \frac{n}{9m_{25}} &= \frac{2n}{1,443} \text{ bins, each containing 1 } p_i \text{ piece, and 9 } t_i \text{ pieces,} \\ \frac{2n}{10m_{26}} &= \frac{11n}{4,995} \text{ bins, each containing 1 } p_i \text{ piece, and 10 } u_i \text{ pieces,} \\ \frac{n}{25} \left( 1 - \frac{1}{m_{25}} \right) &= \frac{19n}{481} \text{ bins, each containing 25 } t_i \text{ pieces,} \\ \frac{2n}{26} \left( 1 - \frac{1}{m_{26}} \right) &= \frac{76n}{999} \text{ bins, each containing 26 } u_i \text{ pieces,} \\ \frac{n}{2} &\text{ bins, each containing 2 } q_i \text{ pieces,} \end{aligned}$$

and

$$\frac{n}{26,676} \text{ bins, each containing 26,676 } v_i \text{ pieces.}$$

Summing, we obtain

$$MH(L) = \left( \frac{3}{2} + \frac{1}{9} + \frac{1}{222} - \frac{1}{987,012} \right) n,$$

or

$$\frac{MH(L)}{OPT(L)} = \frac{3}{2} + \frac{1}{9} + \frac{1}{222} - \frac{1}{987,012}.$$

Since  $OPT(L) = n$  can be arbitrarily large, the result follows.  $\square$

Note that  $987,012 = 26,676 \cdot 37$ . The discrepancy of  $1/987,012$  in our analysis is due to the weights we assigned to the  $I_k$ -pieces. For the list considered in the proof of Lemma 3, bins of type  $B_k$  are completely packed, instead of being only  $\frac{k-1}{k}$  full.

Lemmas 1, 2, and 3 are summarized by the following theorem.

**Theorem 1.** Let  $y = \frac{265}{684}$ ,  $k = 38$ ,  $m_2 = 9$ ,  $m_3 = 12$ , and  $m_\rho = \frac{(k-1)(\rho+1)}{k-\rho-1} = \frac{37(\rho+1)}{37-\rho}$  for  $6 \leq \rho \leq k-2$ . Then

$$\frac{3}{2} + \frac{1}{9} + \frac{1}{222} - \frac{1}{987,012} \leq R_{MH} < \frac{3}{2} + \frac{1}{9} + \frac{1}{222};$$

i.e.,

$$1.6156146 < R_{MH} < 1.61(561)^*.$$

#### 4. A General Lower Bound

In this section, we present a lower bound for a class of algorithms, which includes the Modified Harmonic Algorithm presented in Section 2.

Let  $C$  be the class of algorithms which behave as follows. If  $A \in C$ , then  $A$  divides the interval  $(0, 1]$  into disjoint subintervals, including  $\bar{I}_1 = (1-y, 1]$ ,  $I_1 = (1/2, 1-y]$ ,  $\bar{I}_2 = (y, 1/2]$ ,  $I_2 = (1/3, y]$ , and  $I_\lambda = (0, \lambda]$ , for some  $y$  and  $\lambda$ ,  $1/3 < y < 1/2$  and  $0 < \lambda \leq 1/3$ . Pieces are classified according to the intervals to which they belong. The packing produced by  $A$  must obey the following rules:

(R1) The number of bins which contain only one  $I_2$ -piece (with or without pieces of other types) is a fixed fraction  $1/m$  ( $m$  need not be an integer, and if this fraction is zero, we take  $m$  to be  $\infty$ ) of the total number of  $I_2$ -pieces in the input list.

(R2) No bin may contain

(i) an  $\bar{I}_1$ -piece and an  $I_2$ -piece,

or

(ii) an  $I_1$ -piece and an  $\bar{I}_2$ -piece.

(R3) No bin may contain an  $I_\lambda$ -piece together with an  $\bar{I}_1$ -piece,  $I_1$ -piece,  $\bar{I}_2$ -piece, or an  $I_2$ -piece.

Note that the Modified Harmonic Algorithm is in  $C$ . Also note that the above rule R2 rules out packing  $I_1 \cup \bar{I}_1$ -pieces with  $I_2$ -pieces, or  $I_1$ -pieces with  $I_2 \cup \bar{I}_2$ -pieces, by First-Fit ( $O(n \lg n)$  time). We have the following lower bound on the performance of any algorithm in  $C$ .

**Theorem 2.** For any algorithm  $A \in C$ ,  $R_A \geq \frac{3}{2} + \frac{1}{9} = 1.61^*$ .

**Proof.** We shall exhibit three kinds of lists  $L$  with arbitrarily large  $OPT(L)$ , and show that the average value of  $\frac{A(L)}{OPT(L)}$  for these three kinds of lists is at least  $\frac{3}{2} + \frac{1}{9}$ . Each of these three kinds of lists will be of the form  $L = L_1 L_2 L_3$ , where  $L_1 = (u_1, u_2, \dots, u_n)$ ,  $L_2 = (v_1, v_2, \dots, v_n)$ , and  $L_3 = (t_1, t_2, \dots, t_{Mn})$ . Moreover, we will always have  $u_i + v_i + \sum_{j=1}^M t_{i,j} = 1$ . Hence, the packing in which each bin contains a  $u_i$  piece, a  $v_i$  piece, and  $M$   $t_i$  pieces is an optimal packing, and so  $OPT(L) = n$ .

Throughout the proof, we let  $n$  be a positive integer that is a multiple of  $6m$  (for simplicity, we assume that  $m$  is rational),  $M = \left\lceil \frac{1}{6\lambda} \right\rceil$ , and  $\epsilon$  be such that  $0 < \epsilon < \min \left\{ \frac{(1/2 - y)^2}{(M+1)^2}, \frac{(y - 1/3)^2}{(M+1)^2} \right\}$ .

**Instance 1.** Let

$$u_i = \frac{1}{2} + \frac{M}{2} \epsilon,$$

$$v_i = y + \frac{M}{2} \epsilon,$$

and

$$t_i = \frac{1/2 - y}{M} - \epsilon.$$

Clearly,  $u_i \in I_1$ ,  $v_i \in \bar{I}_2$ , and  $t_i \in I_\lambda$ .

Now consider the packing produced by algorithm  $A$ . By rules  $R1$  to  $R3$  imposed on the algorithms in class  $C$ , this packing consists of

$n$  bins, each containing 1  $u_i$  piece,

at least  $\frac{n}{2}$  bins containing  $v_i$  pieces,

and

at least  $\left\lceil \frac{\frac{nM}{M}}{\left\lfloor \frac{1}{2} - y \right\rfloor} \right\rceil$  bins containing  $t_i$  pieces.

Hence

$$A(L) \geq n + \frac{n}{2} + \left\lceil \frac{\frac{nM}{M}}{\frac{1}{2} - y} \right\rceil \geq (2 - y)n.$$

Since  $OPT(L) = n$  can be arbitrarily large, we have

$$R_A \geq 2 - y. \quad (1)$$

**Instance 2.** Let

$$u_i = \frac{1}{2} + \frac{M}{2}\epsilon,$$

$$v_i = \frac{1}{3} + \frac{M}{2}\epsilon,$$

and

$$t_i = \frac{1}{6M} - \epsilon.$$

Clearly,  $u_i \in I_1$ ,  $v_i \in I_2$ , and  $t_i \in I_3$ .

Now consider the packing produced by algorithm A. This packing consists of

$\frac{n}{2} \left(1 - \frac{1}{m}\right)$  bins, each containing 2  $v_i$  pieces,

$\frac{n}{m}$  bins, each containing 1  $v_i$  piece (and may be 1  $u_i$  piece),

at least  $n \left(1 - \frac{1}{m}\right)$  bins, each containing 1  $u_i$  piece alone,

and

at least  $\frac{n}{6}$  bins containing  $t_i$  pieces.

Hence

$$A(L) \geq \frac{n}{2} \left(1 - \frac{1}{m}\right) + \frac{n}{m} + n \left(1 - \frac{1}{m}\right) + \frac{n}{6} = \left(\frac{5}{3} - \frac{1}{2m}\right)n.$$

Since  $OPT(L) = n$  can be arbitrarily large, we have

$$R_A \geq \frac{5}{3} - \frac{1}{2m}. \quad (2)$$

**Instance 3.** Let

$$u_i = 1 - y + \frac{M}{2}\epsilon,$$

$$v_i = \frac{1}{3} + \frac{M}{2}\epsilon,$$

and

$$t_i = \frac{y - 1/3}{M} - \epsilon.$$

Clearly,  $u_i \in I_1$ ,  $v_i \in I_2$ , and  $t_i \in I_\lambda$ .

Now consider the packing produced by algorithm  $A$ . This packing consists of

$n$  bins, each containing 1  $u_i$  piece,

$\frac{n}{2} \left(1 - \frac{1}{m}\right)$  bins, each containing 2  $v_i$  pieces,

$\frac{n}{m}$  bins, each containing 1  $v_i$  piece,

and

at least  $\left\lceil \frac{\frac{nM}{M}}{y - 1/3} \right\rceil$  bins containing  $t_i$  pieces.

Hence

$$A(L) \geq n + \frac{n}{2} \left(1 - \frac{1}{m}\right) + \frac{n}{m} + \left\lceil \frac{\frac{nM}{M}}{y - 1/3} \right\rceil \geq \left( \frac{7}{6} + y + \frac{1}{2m} \right) n.$$

Since  $OPT(L) = n$  can be arbitrarily large, we have

$$R_A \geq \frac{7}{6} + y + \frac{1}{2m}. \quad (3)$$

Adding (1), (2), and (3), we have

$$3R_A \geq \frac{29}{6},$$

or

$$R_A \geq \frac{3}{2} + \frac{1}{9}. \quad \square$$

If the algorithm  $A$  in Theorem 2 uses Next-Fit to pack  $I_\lambda$ -pieces, then we can get the improved result  $R_A > \frac{3}{2} + \frac{1}{9}$ , by modifying the instances 1, 2, and 3 in the proof as follows. In  $L_3$  introduce

pieces whose sizes are of the form  $k\epsilon$  for some appropriate positive integer  $k$ , at regular frequency. For example, in Instance 2, every  $6M - 1$   $t_i$  pieces will be followed by a piece of size  $(6M + 1)\epsilon$ . Since  $\epsilon$  can be taken to be arbitrarily small, we have  $OPT(L) = n + 1$ . But  $A(L) \geq \left( \frac{3}{2} + \frac{M}{6(M-1)} - \frac{1}{2m} \right) n$ .

## 5. Conclusions

We have presented a new on-line algorithm, the Modified Harmonic Algorithm, which has a better asymptotic performance ratio than any previously known on-line algorithm. Moreover, this algorithm is a linear-time algorithm. It seems likely that a better algorithm could be constructed if the linear time constraint were relaxed.

It is argued in [LL83] that any linear-time, constant-space on-line algorithm has a performance ratio of at least 1.692... . We relaxed the constant space constraint and achieved 1.61(561)\*. In the previous section we described a class of linear-time on-line algorithms which have a performance ratio of at least 1.61\*. It seems quite likely that no linear-time on-line algorithm can do better.

Finally, we observe that our algorithm leads to improved on-line algorithms for packing in two-dimensions. For packing rectangles in a strip, we can devise a shelf algorithm similar to those in [BS83]. By choosing appropriate shelf heights, we can obtain a performance ratio arbitrarily close to  $R_{MH}$ . We can also devise an on-line algorithm for the problem of packing rectangles in finite two-dimensional bins discussed in [CGJ82]. We believe that our on-line algorithm for this problem will have a performance ratio  $R_{MH}^2$ . These will be discussed in more detail in [R84].

## References

- [BS83] Baker, B. S. and Schwarz, J. S., Shelf algorithms for two-dimensional packing problems, *SIAM J. Comput.* **12** (1983), 508-525.
- [B79a] Brown, D. J., A lower bound for on-line one-dimensional bin packing algorithms, Technical Report R-864 (1979), Coordinated Science Laboratory, University of Illinois, Urbana, IL.
- [B79b] Brown, D. J., draft (1980).
- [CGJ82] Chung, F. R. K., Garey, M. R., and Johnson, D. S., On packing two-dimensional bins, *SIAM J. Alg. Disc. Meth.* **3** (1982), 66-76.
- [CGJ83] Coffman, E. G., Jr., Garey, M. R., and Johnson, D. S., Approximation algorithms for bin packing - An updated survey, draft (1983).
- [FeL81] Fernandez de la Vega, W. and Lueker, G. S., Bin packing can be solved within  $1 + \epsilon$  in linear time, *Combinatorica*, **1** (1981), 349-355.
- [FrL81] Friesen, D. K. and Langston, M. A., Analysis of a compound bin packing algorithm, (to appear).
- [GJ79] Garey, M. R. and Johnson, D. S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Co., San Francisco (1979).
- [GJ81] Garey, M. R. and Johnson, D. S., paper in preparation.
- [J73] Johnson, D. S., Near-optimal bin packing algorithms, Technical Report MAC TR-109 (1973), Project MAC, Massachusetts Institute of Technology, Cambridge, Mass.
- [J74] Johnson, D. S., Fast algorithms for bin packing, *J. Comput. Syst. Sci.* **8** (1974), 272-314.
- [JDUGG74] Johnson, D. S., Demers, A., Ullman, J. D., Garey, M. R., and Graham, R. L., Worst-case performance bounds for simple one-dimensional packing algorithms, *SIAM J. Comput.* **3** (1974), 299-325.
- [K72] Karp, R. M., Reducibility among combinatorial problems, in R. E. Miller and J. W. Thatcher (eds.), *Complexity of Computer Computations*, Plenum Press, New York (1972), 85-103.
- [KK82] Karmarkar, N. and Karp, R. M., An efficient approximation scheme for the one-dimensional

bin packing problem, *Proc. 23rd Ann. Symp. on Foundations of Computer Science*, IEEE Computer Soc., Nov. 1982.

- [LL83] Lee, C. C. and Lee, D. T., A simple on-line packing algorithm, Dept. of Electrical Engineering and Computer Science (1983), Northwestern Univ., Evanston, Ill. 60201 (to appear).
- [L80] Liang, F. M., A lower bound for on-line bin packing, *Information Processing Lett.* **10** (1980), 76-79.
- [R84] Ramanan, P., Ph. D. Thesis, University of Illinois at Urbana-Champaign, to appear.
- [Y80] Yao, A. C., New algorithms for bin packing, *J. Assoc. Comput. Mach.* **27** (1980), 207-227.