

# Asymptotic Worst-Case Analyses for the Open Bin Packing Problem

Pornthipa Ongkunaruk

Dissertation submitted to the Faculty of the  
Virginia Polytechnic Institute and State University  
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy  
in  
Industrial and Systems Engineering

Dr. Lap Mui Ann Chan, Co-Chair  
Dr. Kyle Y. Lin , Co-Chair  
Dr. Christine Anderson-Cook  
Dr. Ebru K. Bish  
Dr. Subhash C. Sarin

October 7, 2005  
Blacksburg, Virginia

**Keywords:** Bin Packing, Off-Line Algorithms, Asymptotic Worst-Case Performance Ratio

Copyright 2005, Pornthipa Ongkunaruk

# Asymptotic Worst-Case Analyses for the Open Bin Packing Problem

Pornthipa Ongkunaruk

(ABSTRACT)

The open bin packing problem (OBPP) is a new variant of the well-known bin packing problem. In the OBPP, items are packed into bins so that the total content before the last item in each bin is strictly less than the bin capacity. The objective is to minimize the number of bins used. The applications of the OBPP can be found in the subway station systems in Hong Kong and Taipei and the scheduling in manufacturing industries. We show that the OBPP is NP-hard and propose two heuristic algorithms instead of solving the problem to optimality. We propose two offline algorithms in which the information of the items is known in advance. First, we consider the First Fit Decreasing (FFD) which is a good approximation algorithm for the bin packing problem. We prove that its asymptotic worst-case performance ratio is no more than  $3/2$ . We observe that its performance for the OBPP is worse than that of the BPP. Consequently, we modify it by adding the algorithm that the set of largest items is the set of last items in each bin. Then, we propose the Modified First Fit Decreasing (MFFD) as an alternative and prove that its asymptotic worst-case performance ratio is no more than  $91/80$ . We conduct empirical tests to show their average-case performance. The results show that in general, the FFD and MFFD algorithms use no more than 33% and 1% of the number of bins than that of optimal packing, respectively. In addition, the MFFD is asymptotically optimal when the sizes of items are  $(0, 1)$  uniformly distributed.

Dedicated to my parents and family.

# Acknowledgments

First of all, I would like to express my profound gratitude to my advisors, Dr. Lap Mui Ann Chan and Dr. Kyle Lin who have encouraged me to finish this dissertation. Especially, Dr. Lap Mui Ann Chan gracefully agreed to co-advise me at the late stage of my PhD study, and her dedicated encouragement and insightful guidance has been a key to the completion of this dissertation. Without support and guidance, it might not be possible to accomplish my Ph.D. study.

Secondly, I wish to acknowledge and thank my committee members, Dr. Christine Anderson-Cook, Dr. Ebru K. Bish, and Dr. Subhash C. Sarin, for their support and comments on my research. I learned a lot from their classes and feel fortunate to have them as my committee members. Without their helpful comments, my dissertation would not be as completed as it is.

My heartfelt thanks go to the Royal Thai Government for providing the scholarship that enables me to fulfill the doctoral degree at Virginia Tech. Also thank the Grado Department of Industrial and Systems Engineering for its support and guidance to graduate students. I would love to specially thank Lovedia Cole and Kim Ooms who kindly assisted and supported

me throughout my education here.

Especially thank to Michael McCrea and Xiaomei Zhu who partially edit this dissertation. Indeed thank my roommates, Xiaomei Zhu and Jennifer Tsai. I really appreciate their assistance and encouragement throughout my study. I would like to thank my classmates and friends who empower me to achieve my goal, specifically, Bacer Maddah, Supanit Tangsangasaksri, and Ahmed Ghoniem for their support and assistance.

Finally, I would like to thank my family and friends in Thailand who give me the willpower to complete my dissertation. Especially thank my parents for their support and encourage my entire education.

# Contents

<b>Abstract</b>	<b>ii</b>
<b>Acknowledgments</b>	<b>iv</b>
<b>Table of Contents</b>	<b>v</b>
<b>List of Figures</b>	<b>viii</b>
<b>List of Tables</b>	<b>x</b>
<b>List of Abbreviations</b>	<b>xiii</b>
<b>List of Definitions</b>	<b>xiii</b>
<b>List of Notations</b>	<b>xiv</b>
<b>1 Introduction and Motivation</b>	<b>1</b>
1.1 Problem Description . . . . .	1
1.2 Applications of the Open Bin Packing Problem . . . . .	2
1.3 The Bin Packing Problem (BPP) . . . . .	3

1.4	Heuristic Algorithms for the Open Bin Packing Problem . . . . .	7
1.5	The Worst-Case Analysis . . . . .	8
<b>2</b>	<b>Literature Review</b>	<b>10</b>
2.1	Literature on the Bin Packing Problem (BPP) . . . . .	10
2.2	Literature on the Open Bin Packing Problem (OBPP) . . . . .	16
<b>3</b>	<b>Offline Algorithms and Analyses for the Open Bin Packing Problem</b>	<b>19</b>
3.1	The First Fit Decreasing (FFD) Algorithm . . . . .	19
3.2	The Modified First Fit Decreasing (MFFD) Algorithm . . . . .	24
3.2.1	Preliminary . . . . .	25
3.2.2	Weighting Function . . . . .	30
3.2.3	The Integer Program for the Maximum Weight of the Optimal Bins . .	33
3.2.4	The Proof of Theorem 3.2.1 . . . . .	37
3.2.5	If $\frac{1}{4} \leq s_n < \frac{1}{3}$ , then $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{10}{9}$ . . . . .	37
3.2.6	If $\frac{1}{5} \leq s_n < \frac{1}{4}$ , then $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{91}{80}$ . . . . .	39
3.2.7	If $\frac{1}{6} \leq s_n < \frac{1}{5}$ , then $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{91}{80}$ . . . . .	49
3.2.8	If $\frac{1}{7} \leq s_n < \frac{1}{6}$ , then $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{91}{80}$ . . . . .	58
3.2.9	If $\frac{1}{8} \leq s_n < \frac{1}{7}$ , then $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{91}{80}$ . . . . .	62
<b>4</b>	<b>Empirical Study of the FFD and MFFD Algorithms</b>	<b>67</b>
4.1	Empirical Tests of the FFD and MFFD Algorithms . . . . .	67
4.2	Conclusions and Future Research . . . . .	73

<b>Bibliography</b>	<b>75</b>
<b>Vita</b>	<b>82</b>



# List of Figures

1.1	Harmonic Fit algorithm for the BPP. . . . .	5
1.2	Next Fit and First Fit algorithms for the BPP. . . . .	5
1.3	First Fit and First Fit Decreasing algorithms for the BPP. . . . .	7
3.1	An example of the First Fit Decreasing solution for the OBPP. . . . .	21
3.2	An example of the optimal solution for the OBPP. . . . .	21
3.3	A worst-case example of the First Fit Decreasing algorithm for the OBPP. . .	23
3.4	A worst-case example of the Modified First Fit Decreasing algorithm for the OBPP. . . . .	26
3.5	An example of the MFFD packing that contains a $p_2$ , transition (t), $n_2$ , $p_3$ , and the last bins, respectively. . . . .	38
3.6	Summary of the weight adjustment when $\frac{1}{5} \leq s_n < \frac{1}{4}$ . . . . .	40
3.7	Summary of the matching when $\frac{1}{5} \leq s_n < \frac{1}{4}$ . . . . .	45
3.8	Summary of the weight adjustment when $\frac{1}{6} \leq s_n < \frac{1}{5}$ . . . . .	50
3.9	Summary of the weight adjustment when $\frac{1}{6} \leq s_n < \frac{1}{5}$ . . . . .	54
3.10	Summary of the matching when $\frac{1}{6} \leq s_n < \frac{1}{5}$ . . . . .	55

3.11	Summary of the weight adjustment when $\frac{1}{7} \leq s_n < \frac{1}{6}$ . . . . .	59
3.12	Summary of the weight adjustment when $\frac{1}{8} \leq s_n < \frac{1}{7}$ . . . . .	63
4.1	Summary of the average-case performance of the FFD and MFFD algorithms.	71

# List of Tables

2.1	Summary of heuristics for the Bin Packing Problem. . . . .	12
2.2	Summary of heuristics for the Open Bin Packing Problem. . . . .	18
3.1	Summary of item sizes and weight for $\frac{1}{4} \leq s_n < \frac{1}{3}$ . . . . .	36
3.2	Summary of item sizes and weight for $\frac{1}{5} \leq s_n < \frac{1}{4}$ . . . . .	39
3.3	Summary of item sizes and weight for $\frac{1}{6} \leq s_n < \frac{1}{5}$ . . . . .	50
3.4	Summary of item sizes and weight for $\frac{1}{7} \leq s_n < \frac{1}{6}$ . . . . .	59
3.5	Summary of item sizes and weight for $\frac{1}{8} \leq s_n < \frac{1}{7}$ . . . . .	62
4.1	The average-case performance of the FFD and MFFD algorithms when the number of items is 100. . . . .	68
4.2	The average-case performance of the FFD and MFFD algorithms when the number of items is 500. . . . .	68
4.3	The average-case performance of the FFD and MFFD algorithms when the number of items is 1000. . . . .	69
4.4	The average-case performance of the FFD and MFFD algorithms when the number of items is 2000. . . . .	69

4.5	Summary of the average-case performance of the FFD and MFFD algorithms.	70
4.6	Summary of the average-case performance of the FFD and MFFD algorithms.	70
4.7	Summary of the worst-case performance of the FFD and MFFD algorithms. .	70
4.8	Summary of the worst-case and average-case performance of the FFD and MFFD algorithms of a uniformly distributed data set. . . . .	71
4.9	Summary of the worst-case and average-case performance of the FFD and MFFD algorithms of a mixed-uniformly distributed data set . . . . .	72
4.10	Summary of the worst-case of FFD and MFFD algorithms. . . . .	73

## List of Abbreviations

**BPP** The Bin Packing Problem

**OBPP** The Open Bin Packing Problem

**FFD** The First Fit Decreasing Algorithm

**BFD** The Best Fit Decreasing Algorithm

**FF** The First Fit Algorithm

**MFFD** The Modified First Fit Decreasing Algorithm

## List of Definitions

**$G$  item** an item of size  $\geq 1/2$ .

**$G$  bin** a bin with a  $G$  item in it.

**Non- $G$  item** an item of size less than  $1/2$ .

**Non- $G$  bin** a bin without a  $G$  item in it.

**A  $p$ -item** an item that is packed in a bin when all higher indexed bins are empty.

**A fallback item** an item that is packed in a bin when the higher indexed bin is not empty.

**A  $p_i$  bin** an MFFD bin that contains  $i$  items of  $p_i$ .

**An  $n_i$  bin** an MFFD bin that contains  $i$  items of  $n_i$  and one or more fallback items.

**A transition bin** a bin with the highest indexed item of any of item types except type  $n_1$ .

## List of Notations

$N$  a set of item indices,  $N = \{1, 2, \dots, n\}$ .

$L$  a set of item sizes  $s_j$ , where  $s_j \in (0, 1), \forall j \in N$ ,  $L = \{s_1, s_2, \dots, s_n\}$ .

$z^*$  the minimum number of bins required.

$z^{FFD}(L)$  the number of bins used in the FFD solution to pack all items in a list  $L$ .

$z^H$  the number of bins used in the MFFD solution to pack all items in a list  $L$ .

$k$  the smallest integer such that  $\sum_{j=k+2}^n s_j < k$ .

$L_k$  a set of item sizes starting from item indexed  $k$  to  $n$ ,  $L_k = \{s_{k+1}, \dots, s_n\}$ .

$N_k$  a set of item indices starting from  $k$  to  $n$ ,  $N_k = \{k+1, \dots, n\}$ .

$J_b$  a set of items assigned to the  $b^{th}$  bin opened before item  $s$  is assigned for  $b = 1, 2, \dots, z^H$ ,

$$J_b \subseteq \{k+1, \dots, n\}$$

$I_b = J_b \cup \{b\}$  for  $b = 1, 2, \dots, z^H$ .

$I_b^* \subseteq N = \{1, 2, \dots, n\}$  a set of items in an optimal bin  $b^*$ .

$J_b^* = I_b^* \setminus \{b^*\}$  for  $b^* = 1, 2, \dots, z^*$ .

$J^*$  a set of items in an optimal bin.

$J$  a set of items in an MFFD bin.

$W(L)$  the total weight of items in a list  $L$ .

$W(i)$  the total weight of an items  $i$ .

$\tau$  the total number of transition bins.

$\alpha$  be the number of  $G$  bins in the optimal solution.

$m$  any non- $G$  item with size  $\geq \frac{(1-s_n)}{3}$ .

$t$  any non- $G$  item with size  $< \frac{(1-s_n)}{3}$ .

$G_Y^X$  a set of items of type  $G$  packed with items of set  $X$  in the MFFD solution before Step 5,  
but packed with items of set  $Y$  in the optimal solution.

$u(i)$  a bin that an item  $i$  is assigned in an MFFD solution.

$u^*(i)$  a bin that an item  $i$  is assigned in an optimal solution.

# Chapter 1

## Introduction and Motivation

### 1.1 Problem Description

The Open Bin Packing Problem (OBPP) can be defined as follows: Let  $N = \{1, 2, \dots, n\}$  be a set of item indices and  $L = \{s_1, s_2, \dots, s_n\}$  be the set of item sizes, where  $0 < s_j < 1, \forall j \in N$ . An item can be packed in a bin so long as the total size of the items already in the bin is strictly less than one. After packing the last item, the total size of items is greater than or equal to one. The objective is to use the minimum number of bins to pack all items.

We review the definition of the complexity terminology as discussed in Garey and Johnson [27] and Coffman et al. [15] as follows:

**Decision Problem** The problem, which has only two possible solutions, whether the answer is “yes” or “no.” A decision problem consists of a set  $D_\Pi$  of instances and a subset  $Y_\Pi \subseteq D_\Pi$  of yes-instances.



**Yes instance** An instance belongs to  $Y_{\Pi}$  if and only if the answer for the stated question to that instance is yes.

**Time complexity function ( $O(n)$ )** Let  $f(n)$  be  $O(g(n))$  whenever there exists a constant  $c$  such that  $|f(n)| \leq c \cdot |g(n)|$ .

**Polynomial Time Algorithm** The algorithm whose time complexity function is  $O(p(n))$  for some polynomial function  $p$ , where  $n$  denotes the input length of the time complexity function.

**NP-hard problem** Any decision problem, whether a member of NP or not, to which we can transform an NP-complete problem. It has the property that it cannot be solved in polynomial time unless  $P=NP$ .

Leung et al. [37] prove that OBPP can be polynomially transformed to a 3-Partitioning problem which is an NP-complete problem as shown by Garey and Johnson [27]. Thus, by the definition of an NP-hard problem, the OBPP is an NP-hard problem.

## 1.2 Applications of the Open Bin Packing Problem

Applications of OBPP can be found in a wide variety of industries such as manufacturing, transportation, etc. For example, in the fare payment system in the Hong Kong and Taipei subway stations, the passengers can buy a ticket of a fixed value. A machine at the entrance gateway records the ticket's value, while another machine deducts the fare from the ticket's value at the exit gateway. The machine at the exit gateway returns the ticket if the remaining

balance of the ticket is positive; otherwise, it keeps the ticket. The objective from a passenger's point of view is to minimize the number of tickets they need to purchase or to minimize the number of bins generated by an algorithm.

Another application can be found in job scheduling in the manufacturing environment that operates one shift per day. A worker loads jobs to a machine that can operate automatically. The objective is to schedule jobs to minimize the total time (days) to complete all the jobs. Intuitively, the job with the longest processing time is scheduled to the last to fully utilize the automated machine when the worker leaves from work, and the worker can unload the finished job next day. Similarly, it also applies to the scheduling of running the computer programs.

For the complexity of the Open Bin Packing Problem, Leung et al. [37] prove that OBPP can be polynomially transformed to a 3-Partitioning problem. Thus, the OBPP is an NP-hard problem.

### 1.3 The Bin Packing Problem (BPP)

The Open Bin Packing Problem is a variant problem of a Bin Packing Problem. In this section, we introduce the problem description and the existing heuristics as the guideline of the Open Bin Packing Problem.

In a Bin Packing Problem, let  $N = \{1, 2, \dots, n\}$  be a set of item indices and  $L = \{s_1, s_2, \dots, s_n\}$  be a set of item sizes  $s_j$ , where  $0 < s_j \leq 1, \forall j \in N$ . The objective is to minimize the number of bins used for packing items in  $N$  into a bin such that the total size of items in a bin does not exceed the bin capacity. Assume that the bins have capacity equal to one. According to

Garey and Johnson [27], a BPP is an NP-Complete Problem.

The existing heuristics for the BPP, which can be applied to the OBPP are of interest to us. We review some traditional heuristics for the BPP. Each algorithm indexes bins  $1, 2, \dots, n$  in the order they are opened. The online algorithms are applied as soon as the item arrives and that item cannot be repacked. The existing online algorithms are as follows:

1. Next Fit (NF): NF tests whether to pack item  $j$  in bin  $k$  by checking whether  $s_j$  is no more than the remaining space in bin  $k$ . If so, it packs item  $j$  in bin  $k$  and leaves the bin open. Otherwise, it closes bin  $k$ , and opens a new bin having  $j$  as its first item. NF will open only one bin at a time.
2. Harmonic Fit (HF): HF partitions items into sets according to their sizes. The number of sets can be any positive integer greater than one. Then, NF is applied to each set separately. Figure 1.1 shows the packing by Harmonic Fit algorithm of a list  $L$ .
3. First Fit (FF): FF tests whether to pack item  $j$  to bin  $k$  by checking whether  $s_j$  is no more than the remaining space in bin  $k$ . If so, it packs item  $j$  in the lowest indexed bin having this property. Otherwise, it opens a new bin and packs  $j$  as its first item. FF allows several bins to be opened simultaneously. Figure 1.2 shows the packing by Next Fit and First Fit algorithms of a list  $L$ .
4. Best Fit (BF): BF tests whether to pack item  $j$  to a bin by checking whether  $s_j$  is no more than the remaining space in bin  $k$ . If so, it packs item  $j$  in the highest content bin having this property, ties are broken in favor of the lowest index. Otherwise, it opens a new bin and packs  $j$  as its first item.

$L = \{0.9, 0.2, 0.2, 0.9, 0.4, 0.6, 0.5\}$

Two sets:  $s_i > 0.5$  and  $s_i \leq 0.5$

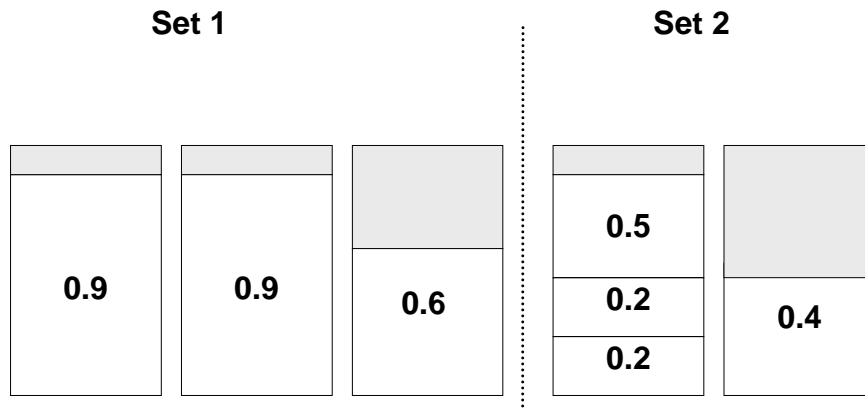
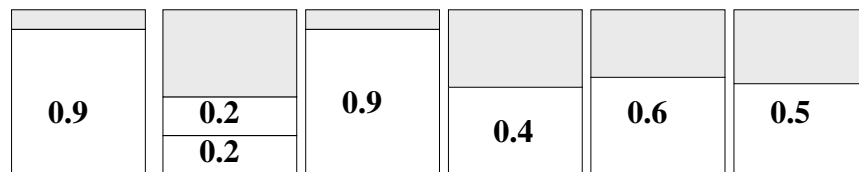
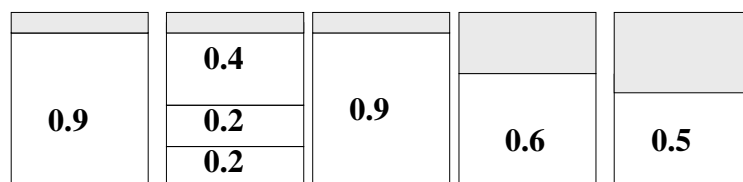


Figure 1.1: Harmonic Fit algorithm for the BPP.

$L = \{0.9, 0.2, 0.2, 0.9, 0.4, 0.6, 0.5\}$



**NF algorithm**



**FF algorithm**

Figure 1.2: Next Fit and First Fit algorithms for the BPP.

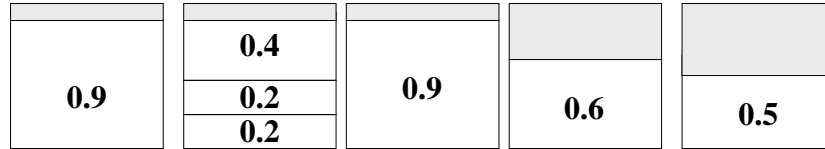
The semi-online algorithms relax the constraint that the items cannot be repacked. It allows the packed items to move after the packing. The number of movement depends on the design of algorithms. It can yield a better algorithm than the online algorithm. For example, the Harmonic Fit (HF) which is similar to HF of the online algorithm except that the packed items are allowed to move after the packing.

The offline algorithms, which have different assumptions when compared to the online algorithms, are applied after the last item arrives and allows the items to be sorted before packing. This results in a better worst-case performance compared to the online or semi-online algorithms as discussed by Coffman et al. [12]. The existing offline algorithms are shown as follows:

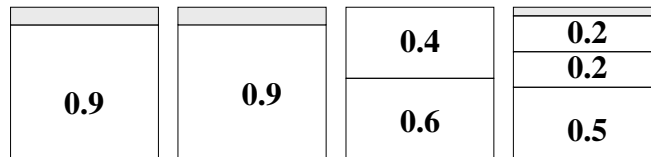
1. Next Fit Decreasing (NFD): First, index items in non-increasing order according to their sizes. Then, apply NF algorithm.
2. First Fit Decreasing (FFD): First, index items in non-increasing order according to their sizes. Then, apply FF algorithm. Figure 1.3 shows the packing by First Fit and First Fit Decreasing algorithms of a list  $L$ .
3. Best Fit Decreasing (BFD): Index items in non-increasing order according to their sizes. Then, apply BF algorithm.

The research presented above focuses on the unit-capacity bin packing problem. On the other hand, the variable-sized bin packing is more general.

$$L = \{0.9, 0.2, 0.2, 0.9, 0.4, 0.6, 0.5\}$$



**FF algorithm**



**FFD algorithm**

Figure 1.3: First Fit and First Fit Decreasing algorithms for the BPP.

### Bounded Space Algorithm

A bounded space algorithm packs items such that the maximum number of bins opened at any time is no more than  $k$  bins, where  $k \geq 1$ . For example, the Next Fit algorithm is a bounded space algorithm with  $k = 1$ . On the other hand, the Best Fit and First Fit algorithms are the unbounded space algorithms with  $k = \infty$ .

## 1.4 Heuristic Algorithms for the Open Bin Packing Problem

The Open Bin Packing Problem (OBPP) is proved by Leung et al. [37] to be NP-hard. Instead of solving OBPP to optimality, we propose heuristic algorithms that provide acceptable

solutions in a short time.

In general, there are three types of algorithms: online, semi-online, and offline algorithms as described in the previous section.

## 1.5 The Worst-Case Analysis

Due to the difficulty of solving OBPP to optimality, heuristic algorithms are of interest to solve the problem. The performance of an algorithm can be measured by the worst-case and average-case performance as in Bramel and Simchi-Levi [8]. Worst case analysis shows how poor a heuristic algorithm can be when it builds the worst solution compared to the best possible one. The deviation from the worst-case solution to the optimal solution represents the performance of the heuristic. However, a heuristic that builds a poor worst-case solution may not generate a poor solution in general. Consequently, the average-case analysis, which is the probabilistic analysis, is an alternative method to determine the performance of the heuristics methods. For example, when the distribution of the size of items is known, there are two ways to determine the average-case performance. First, the analytical method which is very difficult. Second, the empirical tests which run on several instances and replications.

Our study will focus on the asymptotic worst-case ratio. There are two types of worst-case performance ratios as in Coffman et al. [12]. For a given list  $L$  and an algorithm  $A$ , let  $A(L)$  be the number of bins used when algorithm  $A$  is applied to  $L$ . Let  $OPT(L)$  be the minimum required number of bins for a packing of  $L$ , and denote  $R_A(L) = A(L)/OPT(L)$ . First, the

absolute worst-case performance ratio  $R_A$  for an algorithm  $A$  is defined as follows:

$$R_A \equiv \inf \{r \geq 1 : R_A(L) \leq r \text{ for all list } L\} = \sup\{R_A(L)\}.$$

This ratio is often applied to small instances. Simchi-Levi [54] shows that the absolute worst-case ratio for the FF and BF is at least 1.7 and no more than  $7/4$ , and is exactly  $3/2$  for the FFD and BFD algorithms. Chu and La [11] study the absolute worst-case ratios for four approximation algorithms of the variable-Sized bin packing problem. Second, the asymptotic worst-case performance ratio ( $R_A^\infty$ ) is a measurement for the large size problems. It is defined by Vliet [57] as follows:

$$R_A^\infty \equiv \overline{\lim}_{k \rightarrow \infty} \max_L \{R_A(L) | OPT(L) = k.\}$$

In other words,  $R_A^\infty$  is the minimum number of  $\alpha$  such that  $A(L) \leq \alpha OPT(L) + o(OPT(L))$ . Bramel and Simchi-Levi [8] mention that  $R_A^\infty \leq R_A$ . Baker and Coffman [4] and Chu and La [11] define the asymptotic worst-case analysis of the form:  $A(L) \leq \alpha OPT(L) + \beta$  for any  $L$ , where  $\alpha$  and  $\beta$  are constants independent of  $L$ .

The rest of this dissertation is organized as follows: In Chapter 2, we review related literature of the BPP and OBPP including the performance analysis. In Chapter 3, we first implement the First Fit Decreasing algorithm (FFD), which is a well-known algorithm for the BPP, to solve the OBPP. Second, we propose the Modified First Fit Decreasing heuristic algorithm (MFFD) as an alternative. We show that the MFFD algorithm performs better than the FFD and we analyze their asymptotic worst-case performances. Finally, in Chapter 4, we present the empirical study for the FFD and MFFD algorithms.



# Chapter 2

## Literature Review

Our research is related to the BPP and Worst-Case Analysis. We briefly review the related literature in Section 2.1 and briefly review the literature on the OBPP in Section 2.2.

### 2.1 Literature on the Bin Packing Problem (BPP)

Coffman et al. [12] [16] present details of the BPP and heuristic algorithms.

We review some traditional heuristics for the BPP. Each algorithm indexes bins  $1, 2, \dots, n$  in the order they are opened. The online algorithm is applied as soon as the item arrives and that item cannot be repacked. Yao [59] shows that no online algorithm has an asymptotic worst case ratio better than  $3/2$ . He also proposes the online algorithm called Refined First Fit that has  $O(n \log n)$  time with an asymptotic worst-case ratio no greater than  $5/3$ . Liang [38] presents the lower bound for the online packings equal to 1.53635. The existing online algorithms are Next Fit (NF), Harmonic Fit (HF), First Fit (FF), and Best Fit (BF) as described in Chapter

### 1.3.

The semi-online method relaxes the constraint that the items cannot be repacked. It allows the packed items to move after the packing. The number of movements depend on the design of the algorithms. The semi-online algorithm can yield a better algorithm than the online algorithm. Two semi-online algorithms are discussed as follows:

1. Harmonic Fit (HF): Gambosi et al. [25] modify the Harmonic algorithm from an online to a semi-online by allowing the packed item to be repacked no more than once. They propose two types of the HF as follows: i) the four-partition HF in which they prove that its asymptotic worst-case ratio equals to  $3/2$  and it requires  $O(n)$  operations; and ii) the six-partition HF in which they prove that its asymptotic worst-case ratio equals to  $5/4$  and it requires  $O(n \log n)$  operations.
2. The MMP algorithm: Ivkovic and Lloyd [30] propose the MMP and prove that its asymptotic worst-case ratio equals to  $5/4$ . The complexity of MMP is  $O(\log n)$  by the insertion or deletion of items. The MMP is more complex than the algorithm of Gambosi et.al. [25].

The offline algorithm, which has different assumptions than the online algorithm, is applied after the last item arrives and allows the items to be sorted before packing. This results in a better worst-case performance compared to the online or semi-online algorithms as discussed by Coffman et al. [12]. The existing offline algorithms are Next Fit Decreasing (NFD), First Fit Decreasing (FFD) and Best Fit Decreasing (BFD) as described in Chapter 1.3. Table 2.1 summarizes the heuristics and their asymptotic worst-case ratio for the bin packing problem.

Table 2.1: Summary of heuristics for the Bin Packing Problem.

Type	Algorithm	$\bar{R}_A^\infty$	References
<b>Online</b>	Next Fit	2	Johnson [32], Hofri [29]
	Next k Fit	17/10	Mao [42]
	Best k Fit	17/10	Mao [41]
	First Fit	17/10	Johnson [32], Johnson et al. [31]
	Best Fit	17/10	Johnson [32], Johnson et al. [31], Coffman et al. [14]
	Harmonic Fit	1.6910...	Lee and Lee [36]
	Harmonic Fit	1.58887...	Richey [51]
	Harmonic Fit	1.58889...	Seiden[52]
	Bounded Best k Fit	17/10	Csirik and Johnson[19]
	Refined First Fit	5/3	Yao [59]
	Any online algorithm	$\geq 3/2$	Yao [59]
	Any online algorithm	$\geq 1.536...$	Liang [38]
Any online algorithm	$\geq 1.54...$	Vliet [56]	
<b>Semi-Online</b>	Harmonic Fit with 4 Partition	3/2	Gambosi et al. [25]
	Harmonic Fit with 6 Partition	5/4	Gambosi et al. [25]
	MMP	5/4	Ivkovic and Lloyd [30]
<b>Offline</b>	Next Fit Decreasing	1.691...	Baker and Coffman [4]
	First Fit Decreasing	11/9	Johnson [32], Johnson et al. [31], Baker [3] and Yue [60]
	Best Fit Decreasing	11/9	Johnson [32], Johnson et al. [31]
	Refined First Fit Decreasing	$< 11/9$	Yao [59]
	Best two Fit	5/4	Frisen and Langston [24]
	First Fit Decreasing-Best two Fit	6/5	Frisen and Langston [24]

We further review the First Fit Decreasing algorithm. First, Johnson et al. [31], and Johnson [32] show that the asymptotic worst-case performance ratio for First Fit (FF) and Best Fit (BF) is no more than 17/10, and no more than 11/9 for either First Fit Decreasing (FFD) or Best Fit Decreasing (BFD). They prove that  $FFD(L) \leq 11/9 \text{ OPT}(L) + 4$ ,  $\forall L$ , where  $L$  is a list of items in the BPP,  $\text{OPT}(L)$  is the minimum required number of bins

for a list  $L$ , and  $FFD(L)$  is the number of bins used by the FFD heuristic. They use a weighting function to show the worst-case performance of these heuristics. The weighting function depends on the item sizes and the packing location. The proof by Johnson [32] is lengthy.

Later, Baker [3] proves the theorem that  $FFD(L) \leq 11/9 \text{ OPT}(L) + 3, \forall L$ . In his proof, he partitions the last item size into subintervals, and proves that each subinterval does not violate the theorem by the weighting function. He shows that in the FFD bins, an item of size greater than one third requires the same number of bins as would the optimal solution. Moreover, an item of size strictly less than  $2/11$  automatically satisfies the theorem. This reduces the length of the proof by considering only four intervals of the size of the last item in a list  $L$ .

Frisen and Langston [24] use a new technique called the weighting function averaging to prove the worst-case performances of FFD and B2F (Best two fit) algorithms which are equal to  $11/9$  and  $5/4$  respectively. They propose a compound algorithm, which combines both FFD and B2F algorithms in regions in which they are superior. Then, they prove by a weighting function that its worst-case performance is no greater than  $6/5$ . They shorten the length of the proof by reducing the size of the last item to an interval of  $(1/6, 1)$ .

Yue [60] proposes a simpler proof and provides a tighter bound with  $FFD(L) \leq 11/9 \text{ OPT}(L) + 1, \forall L$ . The proof is based on a weighting function and minimal counter example. He shows that the counter example does not exist for his theorem. In addition, he reduces the number of intervals of the size of the last item to three which is similar to the work of Baker [3].

The research mentioned above focuses on the unit-capacity bin packing problem. On

the other hand, the variable-sized bin packing is more general. The details are discussed in Csirik [18], Friesen and Langton [23], Chu and La [11] and Steven et al. [53].

Some researchers are interested in the empirical study instead of the analytical study as discussed in Falkenauer [21], Martello and Toth [44] and Gent [28]. They show the empirical results of solving the BPP by a genetic algorithm and a complete search method. In addition, the Two-Dimensional BPP is discussed in Karp [35], Dowsland [20] and Berkey and Wang [5], while Martello et al. [43], Lim and Ying [39], Raidl [48], and Miyazawa and Wakabayashi [45], [46] study the Three-Dimensional BPP. Sweeney and Paternoster [55] list the published articles, books and dissertations in Cutting and Packing Problems and their applications from 1940 to 1990.

The applications of the BPP can be found in various problems as follows: i) the Scheduling Problems as discussed in Garey et al. [26] and Coffman et al. [13]; ii) the Vehicle Routing and Scheduling Problems as discussed in Federgruen and Ryzin [22]; iii) the Vehicle Routing Problem as discussed in Anily and Bramel [1]; and iv) the Inventory-Routing Problem as discussed in Chan et al. [9].

Anily et al. [2] and Coffman and Leuker [16] state that the BPP is equivalent to the Vehicle Routing Problem (VRP) as follows: Let  $L = (s_1, s_2, \dots, s_n)$  be a list of  $n$  real numbers where  $0 < s_i \leq 1$  is the size of item  $i$ . The items are allocated in a one unit capacity bin so as to minimize the total cost of all bins. The cost of a bin is the function of the numbers of items allocated with the monotonic increasing and concavity properties. In VRP, the number of vehicles used is equivalent to the number of bins used.

Bramel and Simchi-Levi [8] show the worst-case and average-case analyses of the heuristics

that minimize the number of bins for the BPP. Bramel et al. [7] show that the Capacitated VRP is a special case of the BPP with a vehicle capacity equal to one and the demand for each customer is no greater than one. The demand of each customer has to be filled by one vehicle and cannot be split. This is also known as the Capacitated VRP with Unsplit Demand. Since the demand cannot be split, we can introduce the BPP to solve this problem. The cost of a route depends on the length of a tour and the number of customers in each tour. This problem is more realistic than the VRP with Equal Demand, which described by Anily and Bramel [1] that it is the problem of determining how many items are delivered to a fixed capacity vehicle from a warehouse to which set of customers. The demands are assumed to be identical and equal for each customer. Hence, the demand can be split among the vehicles.

Bramel et al. [7] found that traditional BPP heuristics such as Next-Fit, First-Fit, Best-Fit, First-Fit Decreasing, Best-Fit Decreasing, Next-Fit Increasing and Next-Fit Decreasing have the worst-case performance ratios less than two. The average-case analysis of the VRP is done by solving the bin packing problem such that the vehicle capacity equal to one and the demand for each customer is less than one. There exists a solution of the capacitated VRP corresponding to the BPP for every route. Bienstock et al. [6] show that the algorithm for the VRP with unsplit demand is similar to the Next Fit for the BPP which is not asymptotically optimal.

Karmarkar [33], Rhee [49] [50], Ong et al. [47], Coffman and Leuker [16], and Coffman et al. [17] discuss the probabilistic performance of heuristic bin packings. Loulou [40] studies the probabilistic behavior of an optimal bin packing.

Bramel and Simchi-Levi [8] show that in the BPP,  $R_{FFD}$  is no more than  $3/2$ , while

Johnson et al. [31] show that  $R_{FFD}^{\infty} = 11/9$ . Chan et al. [10] show that the optimal solution to the BPP is no more than  $4/3$  of the optimal solution value obtained from solving the linear programming relaxation of the set-partitioning formulation.

## 2.2 Literature on the Open Bin Packing Problem (OBPP)

While there exists abundant research on the BPP, there is much less research on the OBPP. Initially, the OBPP is proposed by Leung et al. [37] who model the ticketing system at the subway station in Hong Kong using the OBPP. They prove that *i*) the OBPP is a strongly NP-hard problem; *ii*) the Next Fit heuristic (NF) has the asymptotic worst-case ratio of two; and *iii*) no online algorithm for OBPP has the asymptotic worst-case ratio less than two. They claim that the optimal properties for OBPP are such that the largest items in the list are assigned to be the last items in each bin, and the total weight of the items is strictly less than the bin capacity after removing the last item. They also propose an offline algorithm for the OBPP that works as follows: Initially, an upper bound (UB) and a lower bound (LB) on the number of optimal bins are calculated as the total weight and half total weight of items in the list, respectively. The algorithm searches for the smallest number  $M \in [LB, UB]$  such that all items can be packed into at most  $(1 + \epsilon) * M$  bins, where  $\epsilon > 0$  is a small amount. Then, the algorithm removes  $M$  items out of the list and packs the remaining items by Karmarkar and Karp [34] algorithm. If it uses no more than  $(1 + \epsilon) * M$  bins, then the algorithm terminates. Otherwise, the algorithm increases  $M$  by one and repeats the process. The algorithm complexity is polynomial and uses  $(1 + \epsilon) * OPT(L)$  bins at most. However,

Leung et al. [37] do not provide the analytical proof for this algorithm. Finally, they show that there exists a polynomial approximation scheme for the OBPP.

Later on, Zhang [61] extends the work of Leung et al. [37]. He considers the parametric ( $m$ ) OBPP, where the item sizes are in the interval of  $(0, 1/m)$  instead of  $(0, 1)$ , where  $m$  is any positive integer. He shows that the Next Fit heuristic has an asymptotic worst-case ratio of  $(m + 1)/m$  and gives a worst-case example. He also proposes a Harmonic algorithm which separates the large and small items into two sets and applies Next Fit algorithm to each set separately. Then, he proves the asymptotic worst-case ratio of the NF and Harmonic in the parametric OBPP and shows that they are lower bounds of that of any online algorithm.

Recently, Yang and Leung [58] study the Ordered Open Bin Packing Problem which extends the requirement corresponding to the order of passenger's itinerary in a subway station problem. In their problem, the item sizes are in  $(0, 1]$ . Three online algorithms are proposed: Mixed Fit, Next Fit and Modified Best Fit. They define 1-piece as an item of size equal to one. They show that with 1-piece, the asymptotic worst-case ratio is greater than 1.630297, and without 1-piece is greater than 1.415715. This ratio is  $25/13$  and  $35/18$  for the Mixed Fit algorithm with and without 1-piece, respectively. Furthermore, this ratio is greater than  $3/2$  with 1-piece and falls in the interval of  $(27/20, 3/2)$  without 1-piece items for the offline algorithm, called Greedy Look-Ahead Next Fit, respectively. Another offline algorithm, called Divide-and-Pack (DP), has an average-case performance equal to one. The result from the asymptotic performance analysis shows that the performance of DP is not better than that of the optimal, but the limit of the ratio approaches one. They also derive a lower bound for any arbitrary algorithm and show that the asymptotic ratio of the expected number of



optimal bins and the number of items equals  $2 - \sqrt{3}$ . They simulate three online algorithms with  $(0,1]$  uniformly distributed item sizes and confirm their analytical results. They also present an average-case analysis of the offline algorithms from simulation without the analytical proof. Their work shows that without 1-pieces, the worst-case ratio is improved. Table 2.2 summarizes the heuristics and their asymptotic worst-case ratios for the OBPP.

Table 2.2: Summary of heuristics for the Open Bin Packing Problem.

Type	Algorithm	$\bar{R}_A^\infty$	References	Note
<b>Online</b>	Next Fit	2	Leung et al. [37]	
	Parametric Next Fit (m)	$(m+1)/m$	Zhang [61]	
	Any online algorithm	$\geq 1.415715\dots$	Yang and Leung [58]	with an ordered requirement
	Next Fit	35/18	Yang and Leung [58]	with an ordered requirement
<b>Offline</b>	Greedy Look-Ahead Next Fit	$(27/20, 3/2)$	Yang and Leung [58]	with an ordered requirement

## Chapter 3

# Offline Algorithms and Analyses for the Open Bin Packing Problem

### 3.1 The First Fit Decreasing (FFD) Algorithm

Leung et al. [37] prove that the Open Bin Packing Problem (OBPP) is NP-Hard. Instead of solving the OBPP to optimality, heuristic algorithms can be used as an attractive alternative. In the previous analytical and empirical results, it is suggested that the well-known heuristics designed so far for the Bin Packing Problem (BPP) are the First Fit Decreasing (FFD) and Best Fit Decreasing (BFD). Both algorithms never require more than 22 % ( $2/9$ ) more bins than the number of optimal bins. Moreover, the FFD is more convenient to use than the BFD from an implementation point of view.

In the following section, we discuss the implementation of the FFD, which is an offline algorithm for the OBPP, and prove that its asymptotic worst-case ratio is no more than  $3/2$ .

Note that  $3/2$  is more than that ratio of the FFD for the BPP ( $11/9$ ). Since the BPP and OBPP are different, so the FFD that performs well in the BPP does not necessary to perform well in the OBPP.

Given an item size  $s_j$  and a list  $L$  of items sizes, where  $L = \{s_1, s_2, \dots, s_n\}$  from the interval  $(0, 1)$ . We index the items in non-increasing order of their sizes. That is  $s_1 \geq s_2 \geq \dots \geq s_n$ . Let  $N = \{1, 2, \dots, n\}$  be a set of item indices.

### **The First Fit Decreasing (FFD) algorithm**

The FFD algorithm consists of three steps, as described below.

Step 1:  $i = 1$ .

Step 2: Pack item  $i$  into the lowest indexed opened bin in which the total size of items already assigned and item  $i$  except the largest item in it is strictly less than one. Open a new bin if it cannot be assigned to any existing bin. (Index the bins in the order they are opened)

Step 3: If  $i = n$ , end; otherwise  $i = i + 1$  and go to Step 3.

Next, we give an example for a list  $L = \{0.9, 0.9, 0.9, 0.7, 0.6, 0.5, 0.4, 0.2, 0.2\}$ . Then, the FFD packs items by using total four bins as shown in Figure 3.1. Let  $z^*(L)$  be the minimum number of bins required and  $z^{FFD}(L)$  be the number of bins used in the FFD solution to pack all items in list  $L$ , respectively.

#### **Lemma 3.1.1**

- (i) *There exists an optimal solution in which each bin contains one of the  $z^*(L)$  largest items.*
- (ii)  $z^*(L) \geq j$ , if  $\sum_{i=j+1}^n s_i \geq j - 1$ .
- (iii) *Let  $j = z^*(L) + 1$ . If  $s_j + s_n \geq 1$ , then item  $j$  belongs to a bin that contains exactly two*

$$L = \{0.9, 0.9, 0.9, 0.7, 0.6, 0.5, 0.4, 0.2, 0.2\}$$

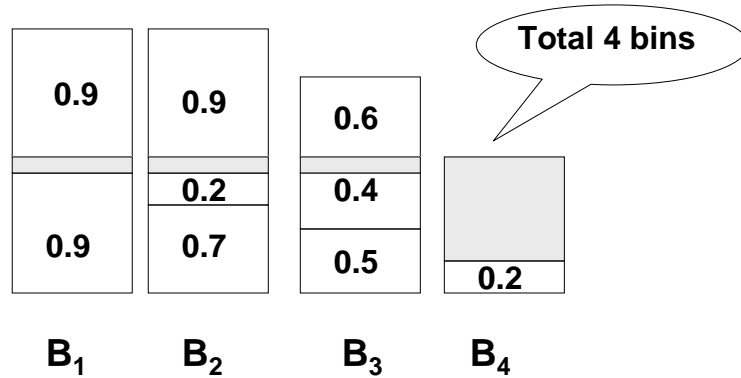
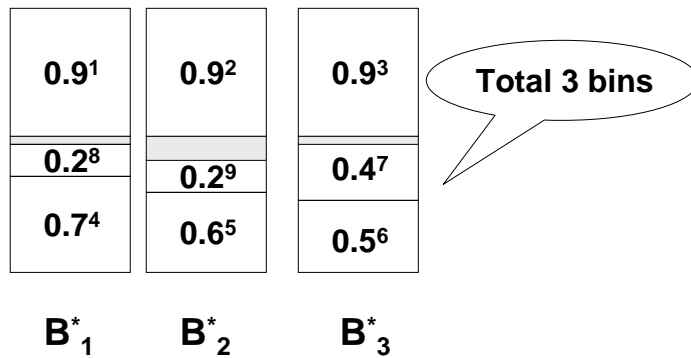


Figure 3.1: An example of the First Fit Decreasing solution for the OBPP.

$$L = \{0.9^1, 0.9^2, 0.9^3, 0.7^4, 0.6^5, 0.5^6, 0.4^7, 0.2^8, 0.2^9\}$$



**Note:** The superscript indicates the index of item.

Figure 3.2: An example of the optimal solution for the OBPP.

items in an optimal solution that satisfies the condition specified in (i).

*Proof:*

(i) If an optimal solution already consists of bins with one of the  $z^*(L)$  largest items each,

then (i) is proved. Otherwise, we index the bins according to the non-increasing order of the largest item in the bins. Then, if we swap item  $j$  with the largest item in bin  $j$ , then we can get another optimal solution. When we repeat this process until the last bin, we have an optimal solution that satisfies (i).

(ii) Consider some  $j$  such that  $\sum_{i=j+1}^n s_i \geq j - 1$ , and suppose  $z^*(L) < j$ . From (i), items  $z^*(L) + 1, z^*(L) + 2, \dots, n$  are packed into  $z^*(L)$  bins without reaching the unit capacity of each one of them. Hence,

$$\sum_{i=z^*(L)+1}^n s_i < z^*(L). \quad (3.1)$$

On the other hand,

$$\sum_{i=z^*(L)+1}^n s_i > \sum_{i=j+1}^n s_i \geq j - 1 \geq z^*(L). \quad (3.2)$$

Because (3.1) contradicts (3.2), we can conclude that  $z^*(L) \geq j$ .

(iii) Suppose  $j$  belongs to the same optimal bin as another item  $i > z^*(L)$ . Then, according to (i),  $s_j + s_i < 1$ . Since this contradicts  $s_j + s_i \geq s_j + s_n \geq 1$ , there are exactly two items in the optimal bin which  $j$  belongs to. ■

For the remaining of the dissertation, an optimal solution refer to one that satisfies the condition in Lemma 3.1.1 (i). Next, we use Lemma 3.1.1 to prove the following theorem.

**Theorem 3.1.1**  $\lim_{z^*(L) \rightarrow \infty} \frac{z^{FFD}(L)}{z^*(L)} \leq \frac{3}{2}$ .

*Proof:* Let  $m = \lfloor \frac{z^{FFD}(L)}{3} \rfloor$ , so that  $z^{FFD}(L) = 3m + j$ , where  $j \in \{0, 1, 2\}$ . From Lemma 3.1.1(iii), in the optimal solution every bin contains at least two items except the last bin. Thus, there are at least  $2m$  items in the first  $m$  bins. Since the total size of all items except

the these  $2m$  items are at least  $2m - 1$ . We have  $\sum_{i=2m+1}^n s_i \geq 2m - 1$ . By Lemma 3.1.1(ii),  $z^*(L) \geq 2m$ . Thus,  $\lim_{z^*(L) \rightarrow \infty} \frac{z^{FFD}(L)}{z^*(L)} \leq \frac{3m+j}{2m} = \frac{3}{2}$ . ■

An instance illustrated in Figure 3.3 is the worst-case solution generated by the FFD. This example shows that this bound is tight.

**A worst-case example:** Suppose there are  $N$  items of size  $1 - \delta$  and  $N^2$  items of size  $\frac{1}{N} - \delta$ , where  $\delta$  is a positive amount less than  $\frac{1}{N^2}$  and  $N$  is any even and positive integer.

**The optimal solution:** The optimal solution requires  $N$  bins. Each bin contains one item of size  $1 - \delta$  and  $N$  items of size  $\frac{1}{N} - \delta$ .

**The FFD solution:** The FFD solution uses  $\frac{3N}{2}$  bins. There are  $\frac{N}{2}$  bins each containing two items of size  $1 - \delta$ , and  $\lceil \frac{N^2}{N+1} \rceil$  bins each containing  $N + 1$  items of size  $\frac{1}{N} - \delta$ . Since the number of bins must be integer, i.e.,  $\lim_{z^*(L) \rightarrow \infty} \left\{ \frac{N}{2} + \frac{N^2}{N+1} \right\} = \lim_{N \rightarrow \infty} \left\{ \frac{N}{2} + \frac{N^2}{N+1} \right\} = \frac{N}{2} + N = \frac{3N}{2}$ .

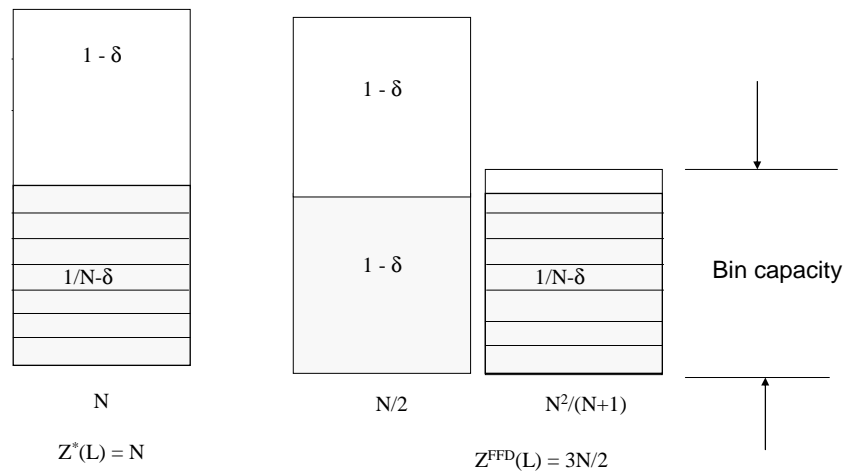


Figure 3.3: A worst-case example of the First Fit Decreasing algorithm for the OBPP.

## 3.2 The Modified First Fit Decreasing (MFFD) Algorithm

In the previous section, we analyze the asymptotic worst-case performance of the FFD. In an offline algorithm, we can improve the performance ratio provided by Lemma 3.1.1(ii) by removing the largest items from a list  $L$  before applying the FFD to the remaining items in the list  $L$ . We call this enhanced version the Modified First Fit Decreasing algorithm. Let  $L_k = \{s_{k+1}, \dots, s_n\}$  and  $N_k = \{k+1, \dots, n\}$ .

### The Modified First Fit Decreasing (MFFD) algorithm

The MFFD algorithm also consists of four steps, as described below.

Step 1: Determine the smallest integer  $k$  such that  $\sum_{j=k+2}^n s_j < k$  and set  $i = k + 1$ .

Step 2: Pack the item  $i$  into the lowest indexed opened bin in which the total size of items already assigned to it is strictly less than one. Open a new bin if it cannot be assigned to any existing bin. Bins are indexed according to the order they are opened.

Step 3: If  $i < n$ , then set  $i = i + 1$  and go to Step 2; otherwise go to Step 4.

Step 4: If the number of bins used in Step 2 exceeds  $k + 1$ , then set  $k = k + 1$ , and return to Step 1. Otherwise, assign item  $b$  to the  $b^{\text{th}}$  opened bin for  $b = 1, 2, \dots, k$ .

Let  $z^{\text{MFFD}}(L)$  be the number of bins used in the MFFD solution to pack all items in list  $L$ . The next theorem specifies the asymptotic worst-case ratio of an MFFD algorithm.

**Theorem 3.2.1**  $\lim_{z^*(L) \rightarrow \infty} \frac{z^{\text{MFFD}}(L)}{z^*(L)} \leq \frac{91}{80}$  and this ratio is tight.

Theorem 3.2.1 implies that the total number of bins generated by the MFFD is no more

than 13.75 % more bins than the number of optimal bins for OBPP. The following example illustrates the tightness of this bound.

**A worst-case example:**

Consider the problem where the list of items includes  $160N$  items each of size  $\frac{5}{18}$ ,  $80N$  items each of size  $\frac{5}{24}$ , and  $240N$  items each of size  $\frac{1}{6}$ , where  $N$  is any positive integer.

**The Optimal solution:** The optimal solution requires  $80N$  bins. Each bin contains two items of size  $\frac{5}{18}$ , one item of size  $\frac{5}{24}$ , and three items of size  $\frac{1}{6}$ .

**The MFFD solution:** First, calculate  $k_1$  from Step 1, resulting in  $k_1 = 80N$ . Second, pack item in  $N_{80N+1}$  and the MFFD uses greater than  $80N + 1$  bins. Since for  $i < 11N$ ,  $k_i = 80N + i < \frac{160 - (80N+i)}{3} + \frac{80N-3}{4} + \frac{240N-4}{5}$ , then the algorithm will continue. When  $i = 11N$ , then  $k_i = 91N = \frac{80N-11N}{3} + \frac{80N-3}{4} + \frac{240N-4}{5} = 91N$ , then the algorithm stops and the MFFD solution uses  $91N$  bins. There are 23 bins each containing four items of size  $\frac{5}{18}$ ; 20 bins each containing one item of size  $\frac{5}{18}$  and four items of size  $\frac{5}{24}$ ; and 48 bins each containing one item of size  $\frac{5}{18}$  and five items of size  $\frac{1}{6}$ . This example shows that this bound is tight.

### 3.2.1 Preliminary

In this section, we present properties and lemmas to support the proof of Theorem 3.2.1. In the following discussion, an optimal solution refers to one that satisfies the condition specified in Lemma 3.1.1(i). Furthermore, we refer an MFFD solution to the one that has  $z^{MFFD}(L) - 1$  largest items packed as the last items and has one item in the last bin. We will prove Theorem 3.2.1 by contradiction. Suppose there exists a list that violates the theorem. Let  $L$  be a minimum list that violates the theorem. That is  $z^{MFFD}(L) > \frac{91}{80}z^*(L)$ , but  $z^{MFFD}(\tilde{L}) \leq$



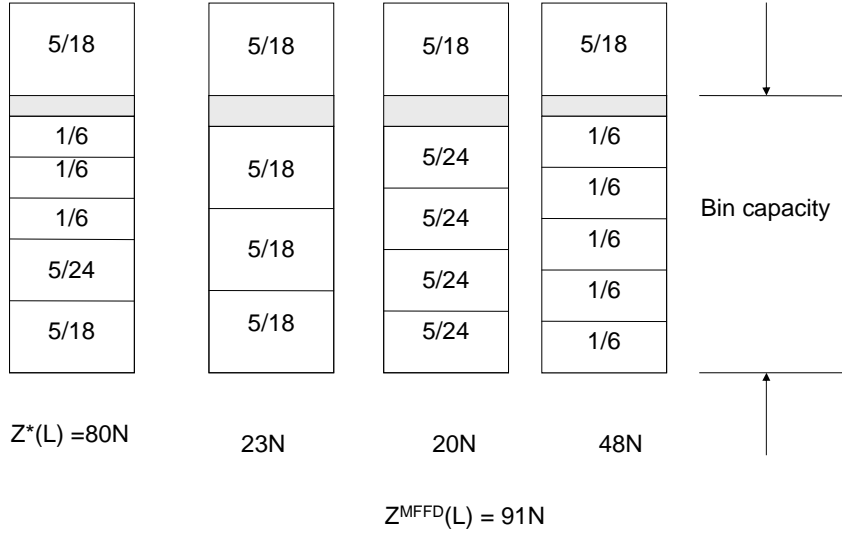


Figure 3.4: A worst-case example of the Modified First Fit Decreasing algorithm for the OBPP.

$\frac{91}{80}z^*(\tilde{L})$  for any  $\tilde{L} \subset L$ .

Denote  $k_i$  as the value of  $k$  in the  $i^{\text{th}}$  iteration of step 2 and  $k_f$  as the final value of  $k$ . Let us first examine the following properties.

**Property 3.2.1**

(i)  $z^*(L) \geq k_1$ .

(ii)  $f > 1$  and at least  $z^{MFFD}(L)$  bins are opened before Step 4 for list  $L_{z^{MFFD}(L)-1}$ .

*Proof:*

(i) The choice of  $k_1$  implies  $\sum_{j=k_1+1}^n s_j \geq k_1 - 1$ , and by Lemma 3.1.1(ii)  $z^*(L) \geq k_1$ . This shows that  $k_1$  is the lower bound of the number of optimal bins.

(ii) Suppose  $k_f = k_1$ , then  $z^{MFFD}(L) \leq k_1 + 1$ . Together with (i), we have  $z^{MFFD}(L) \leq z^*(L) + 1$ . Since this contradicts with  $z^{MFFD}(L) > \frac{91}{80}z^*(L) + 1$ , thus  $k_f > k_1$ . Hence, either  $k_f = z^{MFFD}(L) - 1$  with exactly  $z^{MFFD}(L)$  bins opened before Step 4, or at least

$[z^{MFFD}(L) - 1] + 2$  bins are opened for the list  $L_{z^{MFFD}(L)-1}$  at the second last iteration before Step 4. If  $k_f \neq z^{MFFD}(L) - 1$ , then the solution from the second last iteration before Step 4 after removing all items from bin  $z^{MFFD}(L)^{th}$  to the last bin except the first item of bin  $z^{MFFD}(L)^{th}$  has  $z^{MFFD}(L)$  bins and there are  $z^{MFFD}(L) - 1$  largest items on the top. In addition, the optimal number of bins is not larger, so to assume that  $k_f = z^{MFFD}(L) - 1$  does not reduce the worst-case ratio. ■

Then, we can analyze the MFFD performance under the assumption that its solution is  $z^{MFFD}(L)$  bins and there are  $d = z^{MFFD}(L) - 1$  largest items packed as the last items with only one item in the last bin. After the first item is assigned to the  $z^{MFFD}(L)^{th}$  bin, removing the higher indexed items from the list does not change the number of the bins used by MFFD. Consequently, we can assume that item  $n$  is the first item assigned to the  $(d + 1)^{th}$  bin. In summary, we can analyze the list of items in the MFFD solution, which has  $d$  largest items on the top and has one item in the last bin and call this new list,  $\tilde{L}, \tilde{L} \subset L$ .

**Lemma 3.2.1** *For a minimum list  $\tilde{L}$  with  $\tilde{L}_d$  using exactly  $d + 1$  bins in Step 2 of the MFFD algorithm, then  $z^*(\tilde{L}) < \frac{80}{91}(d + 1)$ . Then, for  $\tilde{\tilde{L}} \subset \tilde{L}$  with  $\tilde{\tilde{L}}_d$  using exactly  $\tilde{d} + 1$  bins in Step 2 of the MFFD algorithm, then  $z^*(\tilde{\tilde{L}}) \geq \frac{80}{91}(\tilde{d} + 1)$ .*

*Proof:* If for a list  $L$ ,  $\frac{z^{MFFD}(L)}{z^*(L)} > \frac{91}{80}$ . Then, for  $\tilde{L} \subset L$  with  $\tilde{L}_d$  using exactly  $d + 1$  bins in Step 2 of the MFFD algorithm, then  $z^*(\tilde{L}) < \frac{80}{91}(d + 1)$ . Since  $\tilde{\tilde{L}} \subset \tilde{L}$  and  $\tilde{L}$  is a minimum list that violates Theorem 3.2.2, then this implies  $z^*(\tilde{\tilde{L}}) \geq \frac{80}{91}(\tilde{d} + 1)$  with  $\tilde{\tilde{L}}_d$  using exactly  $\tilde{d} + 1$  bins in Step 2 of the MFFD algorithm.

Next, let  $k = k_f$  be the value of the last iteration of  $k$  in Step 4. Furthermore, let  $J_b$

$\subseteq \{k+1, \dots, n\}$  be a set of items assigned to the  $b^{\text{th}}$  bin before Step 4 for  $b = 1, 2, \dots, d+1$ , and  $I_b = J_b \cup \{b\}$  for  $b = 1, 2, \dots, d+1$ . Consider an optimal solution for a list  $L = \{s_1, s_2, \dots, s_n\}$ . Let  $I_b^* \subseteq N = \{1, 2, \dots, n\}$  be a set of items in optimal bin  $b$  and  $J_b^* = I_b^* \setminus \{b\}$  for  $b = 1, 2, \dots, z^*(\tilde{L})$ . We also provide the properties that support the proof as follows.

**Property 3.2.2**  $|J_b| \geq 2$ , for any  $b = 1, 2, \dots, \tilde{d}$ .

*Proof:* Consider the list  $\tilde{L}$  with  $\tilde{L}_d$  using exactly  $d+1$  bins in Step 2 of the MFFD algorithm. Suppose  $J_b = \{j\}$  for some  $b = 1, 2, \dots, d$ , then  $s_j + s_s \geq 1$ . Lemma 3.1.1(iii) implies  $J_e^* = \{j\}$  for some  $e = 1, 2, \dots, z^*(\tilde{L})$ . First, remove items  $e$  and  $j$  from the list  $\tilde{L}$ . Then,  $z^*(\tilde{\tilde{L}}) = z^*(\tilde{L}) - 1$ . Removing  $e$  of an optimal bin is equivalent to removing  $b$  of an MFFD bin that contains item  $j$ . Repeat the process until  $|J_b| > 1, \forall b = 1, 2, \dots, d$ . Suppose  $q$  is the number of bins that  $|J_b| = 1$  in a list  $\tilde{L}$ . After the process, we call a new list  $\tilde{\tilde{L}}$  with  $\tilde{\tilde{L}}_d$  using exactly  $\tilde{d}+1$  bins in Step 2 of the MFFD algorithm. Thus,  $\tilde{d}+1 = d+1-q > \frac{91}{80}z^*(\tilde{L}) - q \geq \frac{91}{80}(z^*(\tilde{\tilde{L}})+q) - q \geq \frac{91}{80}z^*(\tilde{\tilde{L}}) + \frac{11q}{80}$ . By Lemma 3.2.1,  $\tilde{L}$  cannot be a minimum list that violates Theorem 3.2.1, but  $\tilde{\tilde{L}}$ . Hence, for  $\tilde{\tilde{L}}$  being a minimum list that violates Theorem 3.2.1,  $|J_b| \geq 2$  for  $b = 1, 2, \dots, \tilde{d}-1$ . ■

**Property 3.2.3** <sup>1</sup> For any  $b = 1, 2, \dots, z^*(\tilde{\tilde{L}})$ ,  $|J_b^*| \geq 3$ .

*Proof:* Consider the list  $\tilde{\tilde{L}}$  with  $\tilde{\tilde{L}}_d$  using exactly  $d+1$  bins in Step 2 of the MFFD algorithm. Suppose  $J_b^* = \{i, j\}$ , where  $i < j$  for some  $b^* = 1, 2, \dots, z^*(\tilde{\tilde{L}})$ . Suppose  $J_b = \{i, k, \dots\}$  where  $i < k$  for some  $b = 1, 2, \dots, d$ .

First, if  $s_j \leq s_k$ , then remove items in bin  $I_b = J_b \cup b$  from a list  $\tilde{\tilde{L}}$ . Repeat the process until  $|J_b^*| > 2, \forall b = 1, 2, \dots, z^*(\tilde{\tilde{L}})$ . Suppose  $r$  is the number of bins that  $|J_b^*| = 2$  in a list  $\tilde{\tilde{L}}$ .

<sup>1</sup>This proof is similar to Yue [60]'s proof.

After the process, we call a new list  $\tilde{\tilde{L}}$  with  $\tilde{\tilde{L}}_{\tilde{d}}$  using exactly  $\tilde{d} + 1$  bins in Step 2 of the MFFD algorithm. Then,  $\tilde{d} + 1 = d + 1 - r$ . Now, consider the optimal bins, replace items  $k$  and  $b$  with  $j$  and  $b^*$ , respectively, thus,  $z^*(\tilde{\tilde{L}}) \leq z^*(\tilde{L}) - r$ . Now,  $\tilde{d} + 1 = d + 1 - r > \frac{91}{80}z^*(\tilde{L}) - r = \frac{91}{80}(z^*(\tilde{L}) + r) - r > \frac{91}{80}z^*(\tilde{L}) + \frac{11r}{80}$ . By Lemma 3.2.1,  $\tilde{L}$  cannot be a minimum list that violates Theorem 3.2.1, but  $\tilde{\tilde{L}}$ . Hence, for  $\tilde{\tilde{L}}$  being a minimum list that violates Theorem 3.2.1,  $|J_b^*| \geq 3$  for  $b = 1, 2, \dots, z^*(\tilde{\tilde{L}})$ .

Second, if  $s_k < s_j$ , by the FFD rule, there exists a bin  $\hat{b}$  with  $J_{\hat{b}} = \{h, j, \dots\}$  where  $s_h > s_i$  in the MFFD bins and  $b$  is any item. Remove items in bin  $I_{\hat{b}} \equiv J_{\hat{b}} \cup \hat{b}$  from a list  $\tilde{L}$ . Repeat the process until  $|J_b^*| > 2, \forall b = 1, 2, \dots, z^*(\tilde{L})$ . After the process, we call a new list  $\tilde{\tilde{L}}$  with  $\tilde{\tilde{L}}_{\tilde{d}}$  using exactly  $\tilde{d} + 1$  bins in Step 2 of the MFFD algorithm. Then,  $\tilde{d} + 1 = d + 1 - r$ . Now, consider the optimal bins, replace the items  $h$  and  $\hat{b}$  with  $i$  and  $b^*$ , thus,  $z^*(\tilde{\tilde{L}}) \leq z^*(\tilde{L}) - r$ . Using a similar argument as the first case, we have for  $\tilde{\tilde{L}}$  being a minimum list that violates Theorem 3.2.1,  $|J_b^*| \geq 3$  for  $b = 1, 2, \dots, z^*(\tilde{\tilde{L}})$ . ■

For the remaining of this dissertation, we denote  $z^H = \tilde{d} + 1$  as the number of bins used in the MFFD solution and  $z^*$  as the minimum number of bins required to pack all items in list  $\tilde{\tilde{L}}$ . Next, we show two lemmas for the proof of Theorem 3.2.1 as follows.

**Lemma 3.2.2** *If  $s_n \geq \frac{1}{3}$  and  $|J_b| \geq 2$  for  $b = 1, 2, \dots, z^H - 1$ , then  $z^H = z^*$ .*

*Proof:* If  $s_n \geq \frac{1}{3}$ , then  $\sum_{j \in J_b^*} s_j < 1$  implies  $|J_b^*| \leq 2$  for  $b = 1, 2, \dots, z^*$ . Thus,  $|J_b| \geq 2$  for  $b = 1, 2, \dots, z^H - 1$  implies  $z^H \leq z^*$ , then  $z^H = z^*$ . ■

**Lemma 3.2.3** *If  $s_n < \frac{1}{8}$ , then  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{9}{8}$ .*

*Proof:* Let  $m = \lfloor \frac{z^H}{9} \rfloor$ , and set  $z^H = 9m + j$ , where  $j = \{0, 1, \dots, 8\}$ . Consider the total size of

the items except the largest one in each bin. By the FFD rule, the item  $n$  with  $s_n < \frac{1}{8}$  cannot be packed in any bin  $b$  in which  $\sum_{i \in J_b} s_i > \frac{7}{8}$ , where  $b \in \{1, \dots, z^H - 1\}$ . Thus, the total size of items in the first  $8m$  bins excluding the lowest indexed items is  $> \frac{7}{8} * 8m = 7m$ . Next, the last  $m + j$  bins including the largest items has the total size at least  $m - 1 + j$ . Thus,

$$\begin{aligned} \sum_{i=8m+1}^n s_i &= \sum_{b=1}^{8m} (\sum_{i \in I_b} s_i - s_b) + \sum_{b=8m+1}^{z^H} \sum_{i \in I_b} s_i. \\ &\geq 7m + m + j - 1 \geq 8m - 1. \end{aligned}$$

By Lemma 3.1.1(ii),  $z^* \geq 8m$ . Thus,  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{9m+j}{8m} = \frac{9}{8}$ . ■

### 3.2.2 Weighting Function

In this section, we present the weighting function that will be used for the rest of the proof. In the literature, Johnson [32], Johnson et al. [31], Baker [3], Frisen and Langston [24] and Yue [60] use the weighting function to prove the FFD worst-case performance of FFD. It shows that the weighting function is an effective method to prove the worst-case performance of an offline algorithm for the bin packing problem. Hence, we use the weighting function in our proof.

In the MFFD packing, the actual size of the item does not indicate the space required for the packing. On the other hand, the weight of the item represents the space required. For example, with  $s_n = \frac{1}{6}$ , if there are three items each of size  $\frac{2}{7}$  with the total size of  $\frac{6}{7}$ , then we assign the weight of these items to  $\frac{1}{3}$ . It shows that each item occupies the equal space of  $\frac{1}{3}$  instead of  $\frac{2}{7}$ . Notice that the weight of an item is not smaller than the size of that item.

Let  $W(i)$  be the weighting function of the item type of item indexed  $i$ ,  $W(\tilde{L})$  be the total

weight of item  $i$ ,  $\forall i \in \tilde{L}$ .

In this section, we describe the weighting function to be used to prove Theorem 3.2.1, where  $\frac{1}{8} \leq s_n < \frac{1}{3}$ .

Next, denote  $R_j$  as the range of the item sizes,  $G$  as an item of size  $\geq \frac{1}{2}$ , and non- $G$  items as the items of size less than  $\frac{1}{2}$ . Moreover, the sizes of items presented in  $J$  and  $J^*$  are non-increasing, i.e., for  $J_b = \{i, j, k\}$ ,  $s_i \geq s_j \geq s_k$ . By the FFD rule, we define the types of the MFFD bins as follows:

1. A Pure Bin. It is a bin in which  $J$  contains only items of the same range ( $R_j$ ), which occurs only to a pure item ( $p_i$ -item), where a  $p_i$ -item is an item that is packed in a bin when all higher indexed bins are empty. There are  $i$  items of type  $p_i$  in  $J$ . The size of a  $p_i$ -item is determined such that MFFD packs  $i$  items of  $p_i$  into a  $p_i$ -bin.
2. A Non-Pure Bin. It is a bin in which  $J$  contains items of different ranges. There are two types of non-pure bins defined as follows.
  - (a) A Fallback Bin. It contains  $i$  items of non-pure items ( $n_i$ -item) and at least one fallback items, where a non-pure item is an item that is packed in a bin with smaller items. A fallback item is an item that is packed in a bin when the higher indexed bin is not empty. There are two types of a fallback bin: an  $n_1$  bin containing a  $G$  item and smaller items, and an  $n_i$  bin, for  $i \geq 2$  containing  $i$  items of  $n_i$  and one or more fallback items.
  - (b) A Transition Bin. It is a bin with the highest indexed item of any of item types except type  $n_1$ . The total number of transition bins is no more than the total

number of the  $p_i$  and  $n_i$  types of bins.

For the remaining of the dissertation, we consider the MFFD packing before Step 4. Denote a bin by its first item type. For example,  $n_1$ -bin ( $G$  bin),  $p_2$ -bin, and  $n_3$ -bin represent the bins that contain  $n_1$ ,  $p_2$  and  $n_3$  as the first item in the bin, respectively. To avoid the numerous values of the weight of different items, we unify the weight of the item type i.e. the weight of the same item type is unique as the following rules:

1. If item  $j$  is of type  $p_i$ , then  $W(j) = \frac{1}{i}$ ,  $\forall i = \{2, \dots, 7\}$ .
2. For the last item ( $n$ ),  $W(n) = \frac{1}{m}$ , where  $m$  is the positive integer such that  $\frac{1}{m+1} \leq s_n < \frac{1}{m}$ ,  $m = \{3, \dots, 7\}$ .
3. If item  $j$  is of type  $n_1$ , then  $W(j) = \frac{3}{4}$ .
4. If item  $j$  is of type  $n_i$ , for  $i \geq 2$ , then  $W(j) = \frac{m-1}{im}$ , where  $m = \{3, \dots, i+1\}$ .

Next, we define the ranges of item sizes for  $\frac{1}{m+1} \leq s_n < \frac{1}{m}$  as follows:

1. To determine the maximum size of an item type  $n_1$  or  $G$ , by Property 3.2.3, we have  $|J_b^*| \geq 3$ . Thus,  $s_{n_1} < 1 - 2 * s_n$ .
2. For a  $p_i$  item,  $1 - s_n \leq i s_{p_i} < 1$ ,  $\forall i < m$ . Thus,  $\frac{1-s_n}{i} \leq s_{p_i} < \frac{1}{i}$ .
3. For an  $n_i$  item  $s_{p_{i+1}} \leq s_{n_i} < s_{p_i}$ ,  $\forall i = \{2, \dots, m-1\}$ .
4. The last range is  $s_n \leq s_{p_i} < s_{n_{i-1}}$ , for  $i = m$ .

Next, we derive the weighting function as the following lemma.

**Lemma 3.2.4**  $z^H - \tau \leq \sum_{b=1}^{z^*} \sum_{i \in J_b^*} W(i) - \sum_{i=z^*+1}^{z^H-1} W(i) - \sum_{b=1}^{z^H} (\sum_{i \in J_b} W(i) - 1)$ , where  $\tau$  is the number of transition bins.

*Proof:* Since  $W(\tilde{L}) = \sum_{b=1}^{z^H} \sum_{i \in I_b} W(i) = \sum_{b=1}^{z^*} \sum_{i \in I_b^*} W(i)$ , we have

$$\begin{aligned} \sum_{b=1}^{z^H} \sum_{i \in I_b} W(i) &= \sum_{b=1}^{z^*} \sum_{i \in I_b^*} W(i). \\ \sum_{b=1}^{z^H} \sum_{i \in I_b} W(i) - \sum_{i=1}^{z^H-1} W(i) &= \sum_{b=1}^{z^*} \sum_{i \in I_b^*} W(i) - \sum_{i=1}^{z^*} W(i) - \sum_{i=z^*+1}^{z^H-1} W(i). \\ \sum_{b=1}^{z^H} \sum_{i \in I_b} W(i) &= \sum_{b=1}^{z^*} \sum_{i \in J_b^*} W(i) - \sum_{i=z^*+1}^{z^H-1} W(i). \\ \sum_{b=1}^{z^H} \sum_{i \in J_b} W(i) - \tau &\leq \sum_{b=1}^{z^*} \sum_{i \in J_b^*} W(i) - \sum_{i=z^*+1}^{z^H-1} W(i). \\ z^H - \tau &\leq \sum_{b=1}^{z^*} \sum_{i \in J_b^*} W(i) - \sum_{i=z^*+1}^{z^H-1} W(i) - \sum_{b=1}^{z^H} (\sum_{i \in J_b} W(i) - 1). \end{aligned}$$

From now on, we consider the packing of items in  $J^*$  and  $J$ , where  $J^*$  and  $J$  is a set of the items in any optimal bin and MFFD bin without their lowest indexed items, respectively.

Next, we determine the maximum value of  $\sum_{i \in J^*} W(i)$  so that we can show that  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{91}{80}$ .

If  $\sum_{i \in J} W(i) > 1$ , we need to adjust the weight so that  $\sum_{i \in J} W(i) = 1$ . For example, if  $\sum_{i \in J} W(i) = 1 + a$ , we subtract the extra weight ( $a$ ) from the MFFD bin and the optimal bin containing the corresponding item. The explanation of the weight adjustment is in Section 3.2.4.

### 3.2.3 The Integer Program for the Maximum Weight of the Optimal Bins

In this section, we show how to determine the maximum value of  $\sum_{i \in J^*} W(i)$  by solving an integer program. The objective is to determine the maximum weight of the optimal packing



so that the total size of the item in a bin is strictly less than one. For  $\frac{1}{m+1} \leq s_n < \frac{1}{m}$ ,  $m = \{3, \dots, 7\}$ , we formulate the following mathematical program.

Problem P1:  $\max \sum_i w(i)x_i.$

$$\text{s.t.} \quad \sum_i LS_i(m)x_i < 1. \quad (3.3)$$

$$x_i \geq 0, \text{ integer } \forall i \in \{n_j, p_{j+1}\}, \forall j = \{1, \dots, m-1\}. \quad (3.4)$$

Where  $x_i$  is the number of item type  $i$  in  $J^*$ ,  $LS_i(m) \in LS(m)$ , where  $LS(m)$  is a set of the smallest size of item types, which is a function of  $s_n$ , where  $s_n$  is a function of  $m$ . Furthermore,  $w(i)$  is the weighting function of the item type  $i$ . Constraint 3.3 requires that the total content of items in  $J^*$  must be strictly less than one. Constraint 3.4 requires that the number of items of type  $i$  in  $J^*$  must be a positive integer. The objective is to determine the maximum weight in  $J^*$ . There are two modifications to transform the mathematical program to the integer program as follows:

1. The Right Hand Side (RHS) Constraint Modification: We modify the right hand side of Constraint 3.1 to  $\sum_{i \in J^*} LS_i(m)x_i \leq 1 - \epsilon$ , where  $\epsilon$  is a small number. To ensure that  $\epsilon$  is small enough, let  $\epsilon = \frac{1}{2c}$ , where  $c$  is the least common factor of the denominators of  $LS(m)$ . For example, for  $s_n = \frac{1}{m+1} = \frac{1}{4}$ ,  $LS(3) = \{\frac{3}{8}, \frac{1}{3}, \frac{1}{4}\}$ , then  $c = \frac{1}{24}$ , and  $\epsilon = \frac{1}{2}(\frac{1}{24}) = \frac{1}{48}$ .
2. The Left Hand Side (LHS) Constraint Modification: Since the value of  $LS_i(m)$  depends on  $s_n \in [\frac{1}{m+1}, \frac{1}{m})$ , and let  $LS_i(m) = a_i + b_i s_n$ . Then, consider the following two cases:
  - (a) If  $b \geq 0$ , then we use the lower bound of  $s_n$  because  $\sum_i (a_i + b_i(\frac{1}{m+1})x_i) \leq \sum_i (a_i +$

$b_i s_n x_i \leq 1 - \epsilon_1$ , where  $\epsilon_1 = \frac{1}{2c_1}$  and  $c_1$  is the least common factor of the denominators of  $LS(m)$ , for  $s_n = \frac{1}{m+1}$ .

(b) If  $b < 0$ , then we use the upper bound of  $s_n$  because  $\sum_i (a_i + b_i(\frac{1}{m} - \delta)x_i) \leq \sum_i (a_i + b_i s_n)x_i \leq 1 - \epsilon_2$ . Let  $l$  be the last item type, which  $LS_l(m) = \frac{1}{m} - \delta$ . Then, the constraint is  $\sum_{i \neq l} LS_i(m)x_i + (\frac{1}{m} - \delta)x_l < 1$ . Hence,  $\sum_{i \neq l} LS_i(m)x_i < 1 - \frac{x_l}{m} - x_l \delta$ . Then,  $\delta \leq \frac{1}{2m x_l c_2}$ , where  $c_2$  is the least common factor of the denominators of  $a_i$ ,  $\forall i \neq l$ . Since  $x_l \leq m$ ,  $\delta \leq \frac{1}{2m^2 c_2}$ . Hence, let  $\delta = \frac{1}{2m^2 c_2}$ . Similar to the previous case, let  $\epsilon_2 = \frac{1}{2c_3}$ , where  $c_3$  is the least common factor of the denominators of  $LS(m)$ , for  $s_n = \frac{1}{m} - \delta$ , respectively.

We can use an upper bound and lower bound of the interval of  $s_n$  to be a set of the item sizes in the integer program and choose the maximum solution of the two solutions because this solution already provides the maximum value.

Now, we formulate the integer program as follows.

Problem P2:  $\max \sum_i w(i)x_i$ .

$$\text{s.t.} \quad \sum_i LS_i(m)x_i \leq 1 - \epsilon \quad (3.5)$$

$$0 \leq x_i \leq v_i \quad \forall i. \quad (3.6)$$

$$x_i = \text{integer} \quad \forall i \in \{n_j, p_{j+1}\}, \forall j = \{1, \dots, m-1\}. \quad (3.7)$$

Constraint 3.5 requires that the total content of items in  $J^*$  must be strictly less than one. Constraint 3.6 sets the maximum number of items of type  $i$  in  $J^*$ . Constraint 3.7 requires that the number of items of type  $i$  must be an integer. After solving the integer program (we

use Lingo 9.0), we can identify  $\sum_{i \in J^*} W(i)$  and item types in  $J^*$ . The advantage of using the integer program is that we can manipulate the number of items of type  $i$  in  $J^*$  by modifying constraint 3.6. For example, we can obtain the maximum value of  $\sum_{i \in J^*} W(i)$  when there is no  $p_2$  item in  $J^*$  by adding  $x_{p_2} = 0$  to the constraint.

Table 3.1: Summary of item sizes and weight for  $\frac{1}{4} \leq s_n < \frac{1}{3}$ .

Items Types	Range of Item Sizes	Item Weight	$v_i$
$p_2$	$[\frac{1-s_n}{2}, \frac{1}{2})$	$\frac{1}{2}$	2
$n_2$	$[\frac{1}{3}, \frac{1-s_n}{2})$	$\frac{1}{3}$	2
$p_3$	$[s_n, \frac{1}{3})$	$\frac{1}{3}$	3

### An Example of the Integer Program

We give an example of the integer program when  $\frac{1}{4} \leq s_n < \frac{1}{3}$ . As the item sizes and weight shown in Table 3.1, if  $s_n = \frac{1}{4}$ , then  $LS(3) = \{\frac{3}{8}, \frac{1}{3}, \frac{1}{4}\}$  and  $\epsilon = \frac{1}{48}$ . First, we formulate the following integer program.

$$\text{Problem P3:} \quad \max \quad \frac{1}{2}x_{p_2} + \frac{1}{3}x_{n_2} + \frac{1}{3}x_{p_3}.$$

$$\text{s.t.} \quad \frac{3}{8}x_{p_2} + \frac{1}{3}x_{n_2} + \frac{1}{4}x_{p_3} \leq 1 - \frac{1}{48}. \quad (3.8)$$

$$0 \leq x_{p_2} \leq 2 \quad (3.9)$$

$$0 \leq x_{n_2} \leq 2 \quad (3.10)$$

$$0 \leq x_{p_3} \leq 3 \quad (3.11)$$

$$x_{p_2}, x_{n_2}, x_{p_3} = \text{integer}. \quad (3.12)$$

Second, when  $m = 3$ , for  $s_n = \frac{1}{m} - \delta = \frac{1}{3} - \frac{1}{108} = \frac{35}{108}$ ,  $LS(3) = \{\frac{73}{216}, \frac{1}{3}, \frac{35}{108}\}$  and  $\epsilon = \frac{1}{432}$ .

Then, modify Constraint 3.8 to  $\frac{73}{216}x_{p_2} + \frac{1}{3}x_{n_2} + \frac{35}{108}x_{p_3} \leq 1 - \frac{1}{432} = \frac{431}{432}$ . After resolving

the problem, the optimal solution is  $J^* = (p_2, n_2, p_3)$  and  $\sum_{i \in J^*} W(i) = \frac{7}{6}$ . Finally, modify Constraint 3.9 to  $x_{p_2} = 0$ , then the optimal solutions become  $J^* = (2n_2, p_3)$ ,  $(n_2, 2p_3)$  and  $(3p_3)$  and  $\sum_{i \in J^*} W(i) = 1$ .

### 3.2.4 The Proof of Theorem 3.2.1

In this section, we use the weighting function as described in Section 3.2.2 for the proof of Theorem 3.2.1, where  $\frac{1}{8} \leq s_n < \frac{1}{3}$ . We partition the size of the last item ( $s_n$ ) into five ranges in  $[\frac{1}{m+1}, \frac{1}{m})$ , where  $m = \{3, \dots, 7\}$ . In each range, we know the maximum number of items in  $J^*$  or  $v_i$ . For example, if  $\frac{1}{4} \leq s_n < \frac{1}{3}$ , then  $v_i \leq 3$ . Then, we combine the integer program and the weighting function to complete the proof with the following lemmas.

**Lemma 3.2.5** *If  $\frac{1}{8} \leq s_n < \frac{1}{3}$ , then  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{91}{80}$ .*

*Proof:* We show the proof of five intervals in the following sections:

**3.2.5** **If  $\frac{1}{4} \leq s_n < \frac{1}{3}$ , then  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{10}{9}$ .**

*Proof:* If  $s_n \geq \frac{1}{4}$  and  $\sum_{j \in J^*} s_j < 1$ , then we have  $|J^*| \leq 3$ . Furthermore, together with Property 3.2.3, we have  $|J^*| = 3$ . Since  $s_{n_1} + 2s_n \geq \frac{1}{2} + 2 * \frac{1}{4} = 1$ ,  $n_1 \notin J^*$ . Table 3.1 shows the item sizes and weight for  $\frac{1}{4} \leq s_n < \frac{1}{3}$ . Note that we assign the weight of items such that  $\sum_{i \in J} W(i) = 1$ .

Figure 3.5 shows an example of the MFFD packing.

According to the FFD rule and discounting the large items that are assigned to the bins in Step 4, the MFFD algorithm will pack two items of type  $p_2$ , two items of type  $n_2$  with an item of type  $p_3$ , or three items of type  $p_3$ ; except the bins with the highest indexed item of

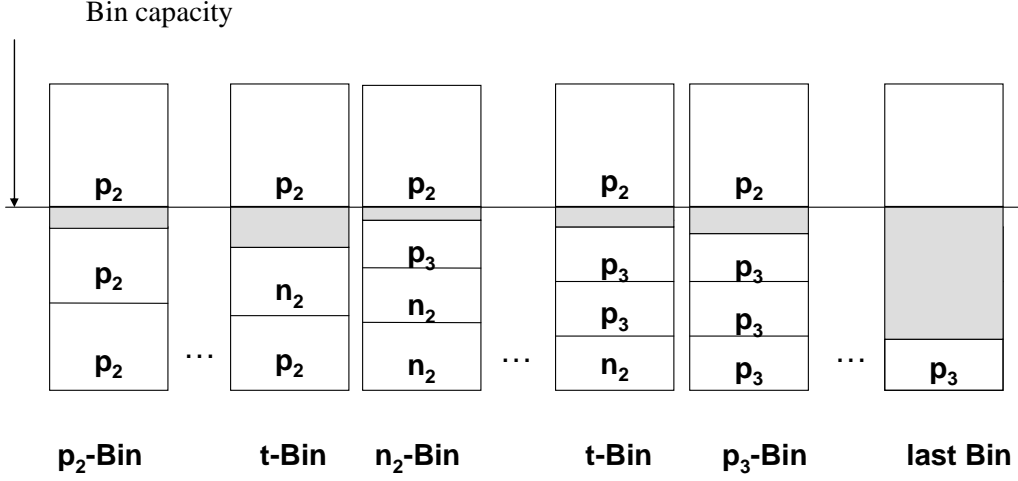


Figure 3.5: An example of the MFFD packing that contains a  $p_2$ , transition (t),  $n_2$ ,  $p_3$ , and the last bins, respectively.

any of three types called the transition bin, which has total weight  $\leq 3$ . Other MFFD bins have  $\sum_{i \in J} W(i) = 1$ . From the integer program in Section 3.2.3, we have  $\sum_{i \in J^*} W(i) \leq \frac{7}{6}$  if there is at most one item of type  $p_2$  in  $J^*$  and  $\sum_{i \in J^*} W(i) \leq 1$  if there is no item of type  $p_2$  in  $J^*$ . By

Lemma 3.2.4, we have

$$\begin{aligned}
 z^H - 3 &\leq \sum_{b=1}^{z^*} \sum_{i \in J_b^*} W(i) - \sum_{i=z^*+1}^{z^H-1} W(i) - \sum_{b=1}^{z^H} (\sum_{i \in J_b} W(i) - 1). \\
 &\leq \frac{7}{6} z^* - \frac{1}{2} (z^H - 1 - z^*). \\
 \frac{3}{2} z^H &\leq \frac{5}{3} z^* + \frac{7}{2}.
 \end{aligned}$$

Then,  $\frac{z^H}{z^*} \leq \frac{10}{9} + \frac{7}{2z^*}$  and  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{10}{9} < \frac{91}{80}$ . ■

**3.2.6** If  $\frac{1}{5} \leq s_n < \frac{1}{4}$ , then  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{91}{80}$ .

*Proof:* According to the FFD rule and discounting the large items that are assigned to the bins in Step 4, the MFFD algorithm will pack an item of type  $G$  with the other one or two of the smaller items, two items of type  $p_2$ , two items of type  $n_2$  with a smaller item, three items of type  $p_3$ , three items of type  $n_3$  with a smaller item or four items of type  $p_4$ ; except the five transition bins, which has total weight less than five, other MFFD bins have total weight equal to one. Let  $G$  bin be a bin that includes an item of type  $n_1$  in  $J^*$ , non- $G$  bin be a bin that does not include an item of type  $n_1$  in  $J^*$  and  $\alpha$  be the number of  $G$  bins in the optimal solution. Furthermore, define a non- $G$  item of size  $\geq \frac{1-s_n}{3}$  as an item of type  $m$  (that is an item of type  $p_2, n_2$  or  $p_3$ ). Also, define a non- $G$  item of size  $< \frac{1-s_n}{3}$  as an item of type  $t$  (that is an item of type  $n_3$  or  $p_4$ ). Table 3.2 summarizes item sizes and weight for  $\frac{1}{5} \leq s_n < \frac{1}{4}$ .

Table 3.2: Summary of item sizes and weight for  $\frac{1}{5} \leq s_n < \frac{1}{4}$ .

Items Types	Range of Item Sizes	Item Weight	$v_i$
$n_1$	$[\frac{1}{2}, \frac{3}{5})$	$\frac{3}{4}$	1
$p_2$	$[\frac{1-s_n}{2}, \frac{1}{2})$	$\frac{1}{2}$	2
$n_2$	$[\frac{1}{3}, \frac{1-s_n}{2})$	$\frac{3}{8}$	2
$p_3$	$[\frac{1-s_n}{3}, \frac{1}{3})$	$\frac{1}{3}$	3
$n_3$	$[\frac{1}{4}, \frac{1-s_n}{3})$	$\frac{1}{4}$	3
$p_4$	$[s_n, \frac{1}{4})$	$\frac{1}{4}$	4

Let  $W$  be the total weight of items in  $J^*$  from the integer program,  $W'$  be the total weight of items in  $J^*$  after the weight adjustment and  $W''$  is the average weight of items in  $J^*$  after the weight adjustment and matching. Then, consider two cases as illustrated in Figure 3.6.

**Case 1:**  $\alpha \leq z^H - 1 - z^*$

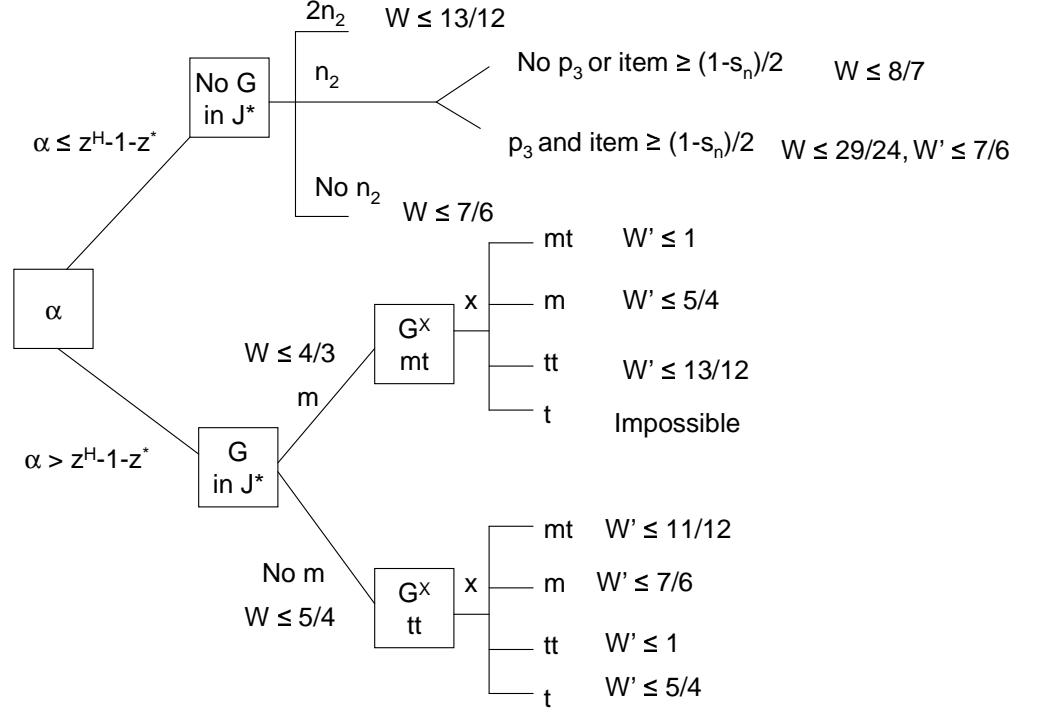


Figure 3.6: Summary of the weight adjustment when  $\frac{1}{5} \leq s_n < \frac{1}{4}$ .

In this case, there is no  $G$  bin in the MFFD solution. We can partition the cases as shown in Figure 3.6. There is the weight adjustment as follows: If there are one item of type  $n_2$ , one item of type  $p_3$  and one item of size  $\geq \frac{1-s_n}{2}$  in  $J^*$ , then  $\sum_{i \in J^*} W(i) \leq \frac{29}{24}$ . Since  $s_{n_2} < \frac{1-s_n}{2}$ , in the MFFD solution, two of  $n_2$  can be packed with  $p_3$ , then  $\sum_{i \in J} W(i) = 2(\frac{3}{8}) + \frac{1}{3} = 1 + \frac{1}{12}$ . By Lemma 3.2.4, we can adjust the weight so that  $\sum_{i \in J} W(i) = 1$ . Since each optimal bin has one of  $n_2$ , then we can adjust half of the extra weight that the MFFD bin with two of  $n_2$  has. Hence, we adjust  $\frac{1}{2}(\frac{1}{12}) = \frac{1}{24}$  from the maximum weight. After the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{29}{24} - \frac{1}{24} = \frac{7}{6}$ . In summary, there are at most  $\alpha$  bins that  $\sum_{i \in J^*} W(i) \leq \frac{4}{3}$  and  $\sum_{i \in J^*} W(i) \leq \frac{7}{6}$  without an item

of type  $G$ . By Lemma 3.2.4, we have

$$\begin{aligned}
z^H - 5 &\leq \sum_{b=1}^{z^*} \sum_{i \in J_b^*} W(i) - \sum_{i=z^*+1}^{z^H-1} W(i) - \sum_{b=1}^{z^H} (\sum_{i \in J_b} W(i) - 1). \\
&\leq \alpha \left( \frac{4}{3} - \frac{3}{4} \right) + \frac{7}{6} (2z^* - z^H + 1) + \left( \frac{29}{24} - \frac{3}{8} \right) (z^H - z^* - \alpha - 1). \\
&\leq \frac{7}{6} (2z^* - z^H) + \frac{5}{6} (z^H - z^*) + \frac{1}{3}. \\
\frac{8}{6} z^H &\leq \frac{9}{6} z^* + \frac{16}{3}.
\end{aligned}$$

Then,  $\frac{z^H}{z^*} \leq \frac{9}{8} + \frac{16}{3z^*}$  and  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{9}{8}$ .

**Corollary 3.2.1** *If  $\sum_{i \in J^*} W(i) \leq \frac{7}{6}$ , then  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{9}{8}$ .*

**Case 2:**  $\alpha > z^H - 1 - z^*$

In this case, we present a weight adjustment of according to Lemma 3.2.3 for the  $G$  bin. Let  $G_Y^X$  be a set of items of type  $G$  packed with items of set  $X$  in  $J$ , but packed with items of set  $Y$  in  $J^*$ . For example, any item  $i \in G_{mt}^m$  is packed with an item of type  $m$  only in  $J$ , but packed with an item of type  $m$  and  $t$  in  $J^*$ . Since an item of type  $m$  is larger than that of type  $t$ , by the FFD rule,  $G_{mt}^t = \emptyset$ . For any item  $i$ , let  $u(i)$  be the bin that  $i$  is assigned in an MFFD solution and  $u^*(i)$  be the bin that  $i$  is assigned in an optimal solution. Next, consider the following weight adjustments:

1. If a  $G$  item is packed with an item of type  $m$  and  $t$  in  $J^*$ , then  $\sum_{i \in J^*} W(i) \leq \frac{4}{3}$ . Then,

consider the MFFD solution as the following possibilities:

- (a) If  $G$  is packed with an item of type  $m$  and  $t$  in  $J$ , then  $\sum_{i \in J^*} W(i) \leq \frac{4}{3}$  and  $\sum_{i \in J} W(i) \geq$

$\frac{4}{3} = 1 + \frac{1}{3}$ . We adjust the weight of the optimal bin by  $\frac{1}{3}$ . After the adjustment,

$$\sum_{i \in J^*} W(i) \leq \frac{4}{3} - \frac{1}{3} = 1. \text{ Hence, for any } i \in G_{mt}^m, \sum_{i \in J^*} W(i) \leq 1.$$



(b) If  $G$  is packed with an item of type  $m$  in  $J$ , then consider the following cases:

- i. If there exists  $J^*$  with an item of type  $n_2$ , then the MFFD can pack an item of type  $G$  with  $n_2$ , and  $\sum_{i \in J} W(i) = \frac{3}{4} + \frac{3}{8} = 1 + \frac{1}{8} = 1 + \frac{1}{12} + \frac{1}{24}$ . In Case 1, we adjust the weight of the item of type  $n_2$  in the non- $G$  bin by  $\frac{1}{24}$ , then we adjust  $\frac{1}{12}$  for the  $G$  bin. Hence, we subtract  $\frac{1}{12}$  from the maximum weight. After the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{4}{3} - \frac{1}{12} = \frac{5}{4}$ .
- ii. If there does not exist  $J^*$  with one item of type  $n_2$  in the optimal solution, then there are two cases as follows:

- If there does not exist  $J^*$  with an item of type  $p_3$ , then  $\sum_{i \in J^*} W(i) \leq \frac{5}{4}$ .
- If there exists  $J^*$  with an item of type  $p_3$ , then the MFFD can pack an item of type  $G$  with  $p_3$ , and  $\sum_{i \in J} W(i) = \frac{3}{4} + \frac{1}{3} = 1 + \frac{1}{12}$ . Hence, we subtract  $\frac{1}{12}$  from the maximum weight. After the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{4}{3} - \frac{1}{12} = \frac{5}{4}$ .

Hence, for any  $i \in G_{mt}^m$ ,  $\sum_{i \in J^*} W(i) \leq \frac{5}{4}$ .

(c) If  $G$  is packed with two items of type  $t$  in  $J$ , then  $\sum_{i \in J} W(i) \geq \frac{5}{4} = 1 + \frac{1}{4}$ . We adjust the weight of the optimal bin by  $\frac{1}{4}$ . After the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{4}{3} - \frac{1}{4} = \frac{13}{12}$ .

Hence, for any  $i \in G_{mt}^{tt}$ ,  $\sum_{i \in J^*} W(i) \leq \frac{13}{12}$ .

(d) If  $G$  is packed with one item of type  $t$  in  $J$ , then by the FFD rule, a  $G$  item must be packed with an item of type  $m$  before an item of type  $t$ . Thus, this case does not exist.

2. If a  $G$  item is packed with two items of type  $t$  in  $J^*$ , then  $\sum_{i \in J^*} W(i) \leq \frac{5}{4}$ . Then, consider the MFFD solution as the following possibilities:

(a) If  $G$  is packed with an item of type  $m$  and  $t$  in  $J$ , then  $\sum_{i \in J} W(i) \geq \frac{4}{3} = 1 + \frac{1}{3}$ .

We adjust the weight of the optimal bin by  $\frac{1}{3}$ . After the adjustment,  $\sum_{i \in J^*} W(i) \leq$

$$\frac{5}{4} - \frac{1}{3} = \frac{11}{12}. \text{ Hence, for any } i \in G_{tt}^{mt}, \sum_{i \in J^*} W(i) \leq \frac{11}{12}.$$

(b) If  $G$  is packed with an item of type  $m$  in  $J$ , then  $\sum_{i \in J} W(i) \geq \frac{13}{12} = 1 + \frac{1}{12}$ . We adjust

the weight of the optimal bin by  $\frac{1}{12}$ . After the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{5}{4} - \frac{1}{12} = \frac{7}{6}$ .

$$\text{Hence, for any } i \in G_{tt}^m, \sum_{i \in J^*} W(i) \leq \frac{7}{6}.$$

(c) If  $G$  is packed with two items of type  $t$  in  $J$ , then  $\sum_{i \in J} W(i) \geq \frac{5}{4} = 1 + \frac{1}{4}$ . We adjust

the weight of the optimal bin by  $\frac{1}{4}$ . After the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{5}{4} - \frac{1}{4} = 1$ .

$$\text{Hence, for any } i \in G_{tt}^{tt}, \sum_{i \in J^*} W(i) \leq 1.$$

(d) If  $G$  is packed with an item of type  $t$  in  $J$ , then  $\sum_{i \in J} W(i) \geq 1$ . Hence, for any  $i \in$

$$G_{tt}^t, \sum_{i \in J^*} W(i) \leq \frac{5}{4}.$$

After the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{5}{4}$  with an item of type  $G$  and  $\sum_{i \in J^*} W(i) \leq \frac{7}{6}$  without an item of type  $G$ . However, the proof is not completed without the matching of the  $G$  bins in the optimal and MFFD solutions. When  $i \in G_{mt}^m$  or  $G_{tt}^t$ , the weight of  $J^*$  is maximized. Next, consider the following property.

**Property 3.2.4** *There does not exist an item of type  $t$  belonging to both  $\{j \in J_{u^*(i)}^* : i \in G_{tt}^t\}$  and  $\{j \in J_{u(i)} : i \in G_{tt}^t\}$ .*

*Proof:* Recall that  $J_b^*$  is a set of items in an optimal bin  $b$ . Now, let  $J_b^*$  represent the type of items in it. For example,  $J_1^* = \{G_1, t'_1, t''_1\}$  and  $J_2^* = \{G_2, t'_2, t''_2\}$ , where the items in  $J^*$  indicates an item type, and a bin index. For example  $G_1$  is the item of type  $G$ ,  $t'_1$  and  $t''_1$  are the first and second item of the item of type  $t$  of the first bin, respectively. Suppose  $t'_2$  belongs

to both  $\{j \in J_{u^*(i)}^* : i \in G_{tt}^t\}$  and  $\{j \in J_{u(i)} : i \in G_{tt}^t\}$ . Then,  $J_1 = \{G_1, t_2'\}$ . Since  $t_2'$  is packed with  $G_1$  only in  $J$  but with  $G_2$  and  $t_2''$  in  $J^*$ ,  $s_{G_1} > s_{G_2}$ . By the FFD rule, an item of type  $t$  packed with  $G_2$  in  $J$  must be too large to be packed with  $G_1$ , then  $s_{G_1} + s_t \geq 1$ . In addition, since  $s_{G_1} < 1 - 2s_n$ , then  $s_t \geq 1 - s_{G_1} = 2s_n$ . For  $s_n \leq \frac{1}{6}$ ,  $s_t \geq \frac{2}{6}$ . However, the size of  $t$  is  $< \frac{1-s_n}{3} = \frac{5}{18} < \frac{1}{3}$ , then there is a contradiction. ■

This property shows that the item of type  $t$ , which is packed with  $G$  in the MFFD solution, is not from the bin with  $i \in G_{tt}^t$  in the optimal solutions. Next, consider the following property.

**Property 3.2.5** *For any item  $i \in G_{tt}^t$  and  $j \in G_{mt}^m$ ,  $s_i < s_j$ .*

*Proof:* Since item  $i$  is packed with an item of type  $t$  in  $J$ , then  $s_i \geq 1 - \frac{1-s_n}{3} - s_n$ , but item  $j$  is packed with one item of type  $m$  and another item of type  $t$  in  $J^*$ , then  $s_j < 1 - \frac{1-s_n}{3} - s_n \leq s_i$ . ■

Next, consider the matching as illustrated in Figure 3.7. Next, we calculate the average weight of the matching as described in following cases:

**Case 2.1:**  $G_{mt}^m = \emptyset$  and  $G_{tt}^t \neq \emptyset$

Let  $\gamma_i$  be the number of optimal bins that  $J^*$  contains at most  $i$  items of type  $t$ . By Property 3.2.4,  $t$  is not packed with an item of type  $G_{tt}^t$ , then it must be packed with an item of type  $t$  from either  $G$  bin with  $\sum_{i \in J^*} W(i) \leq \frac{7}{6}$  or non- $G$  bin with  $\sum_{i \in J^*} W(i) \leq \frac{7}{6}$ .

Consider the matching, since an item of type  $t$  in  $J$  can come from two types of optimal bins. First, if it is from the bin with at most two items of type  $t$  in  $J^*$  with  $\sum_{i \in J^*} W(i) \leq \frac{7}{6}$ , then at most two  $G$  bins can be packed with these items. Thus, the average weight is  $\leq \frac{5}{4}(\frac{2}{3}) + \frac{7}{6}(\frac{1}{3}) = \frac{11}{9}$ . Similarly, if an item of type  $t$  in  $J$  is from the bin with at least three items of type  $t$  in  $J^*$  with  $\sum_{i \in J^*} W(i) \leq \frac{9}{8}$ , then the average weight is  $\leq \frac{5}{4}(\frac{3}{4}) + \frac{9}{8}(\frac{1}{4}) = \frac{39}{32}$ . In each

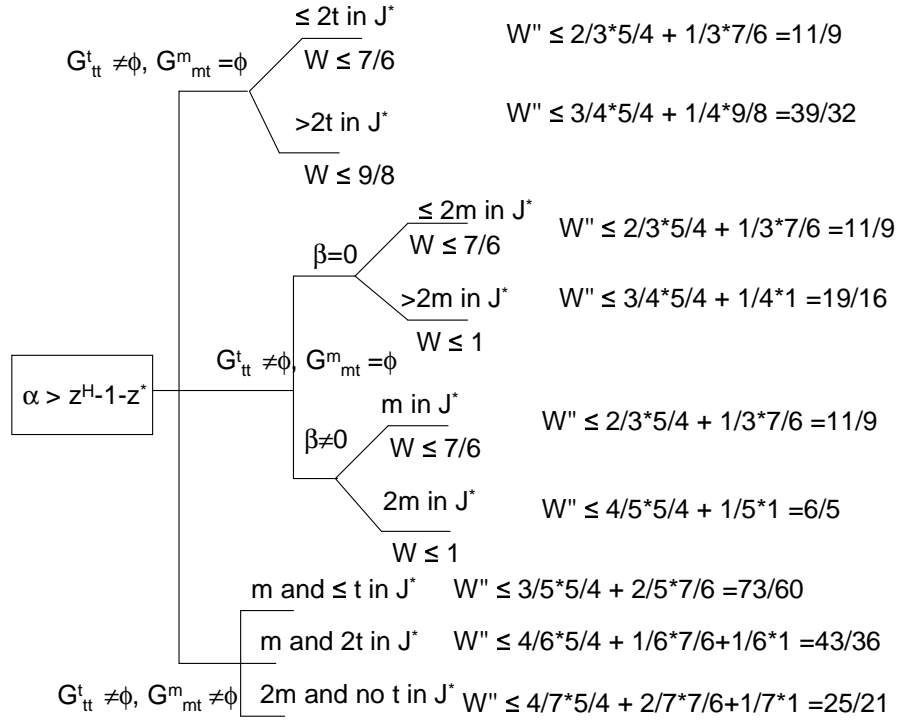


Figure 3.7: Summary of the matching when  $\frac{1}{5} \leq s_n < \frac{1}{4}$ .

case, we can add the constraint in the integer program. Since  $\frac{11}{9} > \frac{39}{32}$ , then the average weight is  $\leq \frac{11}{9}$ . In summary, after the weight adjustment and matching as illustrated in Figure 3.7, the average weight is  $\leq \frac{11}{9}$ . By Lemma 3.2.4, we have

$$\begin{aligned}
z^H - 5 &\leq \sum_{b=1}^{z^*} \sum_{i \in J_b^*} W(i) - \sum_{i=z^*+1}^{z^H-1} W(i) - \sum_{b=1}^{z^H} \left( \sum_{i \in J_b} W(i) - 1 \right). \\
&\leq \frac{5}{4}(\alpha - (z^H - z^* - 1)) + \frac{7}{6}(z^* - \alpha) + \left(\frac{4}{3} - \frac{3}{4}\right)(z^H - z^* - 1). \\
&\leq \frac{5}{4}|G_{tt}^t| + \frac{8}{7}\gamma_1 + \frac{7}{6}\gamma_2 + \frac{9}{8}\gamma_3 + \gamma_4 + \frac{7}{6}(z^* - |G_{tt}^t| - \gamma_1 - \gamma_2 - \gamma_3 - \gamma_4 + z^* - z^H) \\
&\quad + \left(\frac{4}{3} - \frac{3}{4}\right)(z^H - z^*) + \frac{2}{3},
\end{aligned}$$

with  $\gamma_1 + 2\gamma_2 + 3\gamma_3 + 4\gamma_4 \geq |G_{tt}^t|$  and  $2z^* - z^H - |G_{tt}^t| - \gamma_1 - \gamma_2 - \gamma_3 - \gamma_4 \geq 0$ .

$$\leq \frac{5}{4}|G_{tt}^t| + \frac{8}{7}\gamma_1 + \frac{7}{6}\gamma_2 + \frac{9}{8}\gamma_3 + \gamma_4 + \frac{7}{12}(z^H - z^*) + \frac{2}{3},$$

$$\begin{aligned}
& \text{with } \gamma_1 + 2\gamma_2 + 3\gamma_3 + 4\gamma_4 \geq |G_{tt}^t| \text{ and } |G_{tt}^t| - \gamma_1 - \gamma_2 - \gamma_3 - \gamma_4 = 2z^* - z^H. \\
& \leq \frac{5}{4}|G_{tt}^t| + \frac{7}{6}\gamma_2 + \frac{7}{12}(z^H - z^*) + \frac{2}{3}, \text{ with } 2\gamma_2 \geq |G_{tt}^t| \text{ and } |G_{tt}^t| + \gamma_2 = 2z^* - z^H. \\
& \leq \frac{5}{4}\left(\frac{2(2z^* - z^H)}{3}\right) + \frac{7}{6}\left(\frac{(2z^* - z^H)}{3}\right) + \frac{7}{12}(z^H - z^*) + \frac{2}{3}.
\end{aligned}$$

the RHS is maximized when  $\gamma_1 = \gamma_3 = \gamma_4 = 0$ .

$$\begin{aligned}
& \leq \frac{11}{9}(2z^* - z^H) + \frac{7}{12}(z^H - z^*) + \frac{2}{3}, \text{ the maximum average weight is } \frac{11}{9}. \\
& \leq \frac{67}{36}z^* - \frac{23}{36}z^H + \frac{2}{3}.
\end{aligned}$$

Then,  $\frac{z^H}{z^*} \leq \frac{67}{59} + \frac{17}{3z^*}$  and  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{67}{59} < \frac{91}{80}$ .

**Corollary 3.2.2** *If  $\sum_{i \in J^*} W(i) \leq \frac{11}{9}$ , then  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{67}{59}$ .*

**Case 2.2:**  $G_{mt}^m \neq \emptyset$  and  $G_{tt}^t = \emptyset$

Let  $\beta$  be the number of items of type  $m$  belonging to both  $j \in J_{u^*(i)}^* : i \in G_{mt}^m$  and  $j \in J_{u(i)} : i \in G_{mt}^m$ . Since  $G$  is not packed with an item of type  $t$  in the MFFD solution, then it must be packed with an item of type  $m$  only. There are two possibilities as follows:

A. If  $\beta = 0$ , then there is no item of type  $m$  belonging to both  $\{j \in J_{u^*(i)}^* : i \in G_{mt}^m\}$  and  $\{j \in J_{u(i)} : i \in G_{mt}^m\}$ . Since  $G$  is not packed with an item of type  $t$  in the MFFD solution, then it must be packed with an item of type  $m$  only. Since an item of type  $m$  in  $J$  can come from two types of optimal bins. First, if the bin has at most two items of type  $m$  in  $J^*$ , then at most two  $G$  bins can be packed with these items. Thus, the average weight is  $\leq \frac{11}{9}$ . Second, if the bin has at least three items of type  $m$  in  $J^*$ , then the average weight is  $\leq \frac{19}{16}$ . Since  $\frac{11}{9} > \frac{19}{16}$ . By Corollary 3.2.2,  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{67}{59} < \frac{91}{80}$ .

B. If  $\beta \neq 0$ , then there exists an item  $k$  of type  $m$  belonging to both  $j \in J_{u^*(i)}^* : i \in G_{mt}^m$  and  $j \in J_{u(i)} : i \in G_{mt}^m$ .

Thus, there must exist at least two sets of  $G_{mt}^m$  bin and one of them contains an item  $k$ . Let  $J_1^* = \{G_1, m_1, t_1\}$  and  $J_2^* = \{G_2, k, t_2\}$ . Then,  $J_1 = \{G_1, k\}$ . Since  $k$  is packed with  $G_1$  only in the MFFD solution but with  $G_2$  and  $t_2$  in the optimal solution,  $s_{G_1} > s_{G_2}$ . For  $J_2 = \{G_2, m_2\}$ , by the FFD rule,  $m_2$  must be too large to be packed with  $G_1$ ,  $s_{G_1} + s_{m_2} > 1$ . In addition,  $s_{G_1} < 1 - s_{m_1} - s_{t_1} < 1 - \frac{1-s_n}{3} - s_n$ , then  $s_{m_2} \geq 1 - s_{G_1} = \frac{1-s_n}{3} + s_n = \frac{4}{15} + \frac{1}{5} = \frac{7}{15}$ . Thus,  $m_2$  is not from the  $G$  bin because  $s_G + s_{m_2} + s_t \geq \frac{1}{2} + \frac{7}{15} + \frac{1}{5} > 1$ . There are at most two items of type  $m$  of size  $\geq \frac{7}{15}$  in  $J^*$ .

An item of type  $m$  can come from two types of optimal bins. First, if the bin has two items of type  $m$  of size  $\geq \frac{7}{15}$  in  $J^*$ . Then, there are at most two portions of  $G_{mt}^m$  with an item  $k$  and two portions of  $G_{mt}^m$  without an item  $k$  and one non- $G$  bin. Thus, the average weight is  $\leq \frac{6}{5}$ . Second, if an item of type  $m$  is from the bin with at most one item of type  $m$  of size  $\geq \frac{7}{15}$  in  $J^*$ , then there are at most one portion of  $G_{mt}^m$  with an item  $k$  and one portion of  $G_{mt}^m$  without an item  $k$  and one non- $G$  bin. Since  $\frac{11}{9} > \frac{6}{5}$ , the average weight is  $\leq \frac{11}{9}$ . By Corollary 3.2.2,  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{67}{59} < \frac{91}{80}$ .

**Case 2.3:**  $G_{mt}^m \neq \emptyset$  and  $G_{tt}^t \neq \emptyset$

In Case 2.1 and 2.2, we show that if there is no matching of items in a  $G_{mt}^m$  bin and  $G_{tt}^t$  bin, then  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{67}{59}$ . In this case, we consider when there is a matching of items in these sets i.e. in the MFFD bin, an item of type  $t$  in a  $G_{tt}^t$  bin comes from the optimal  $G_{mt}^m$  bin.

When  $G_{mt}^m \neq \emptyset$  and  $G_{tt}^t \neq \emptyset$ , let  $J_1^* = \{G_1, t'_1, t''_1\}$ ,  $J_2^* = \{G_2, m_2, t_2\}$ . By Property 3.2.5,  $s_{G_1} > s_{G_2}$ , by the FFD rule,  $G_1$  must be packed with  $m_2$  before an item of type  $t$ . If  $G_{tt}^m = \emptyset$ , then  $G_1$  is not packed with  $m_2$  in the MFFD solution. This implies  $s_{G_1} + s_{m_2} \geq 1$ ,  $s_{G_1} \geq 1 - s_{m_2} = 1 - \frac{1-s_n}{3} = \frac{2+s_n}{3}$ . This is a contradiction to the size of an item of type  $G$ .

Thus,  $s_{G_1} + s_{m_2} < 1$  and  $G_{tt}^m \neq \emptyset$ . This implies that one portion of  $G_{mt}^m$  matches with one portion of  $G_{tt}^m$ . Furthermore, one portion of  $G_{tt}^t$  matches with one portion of  $G_{mt}^m$ .

Consider one portion of a set of  $G_{tt}^m$ ,  $G_{tt}^t$  and  $G_{mt}^m$ . Let  $G_1 \in G_{tt}^m$ ,  $G_2 \in G_{tt}^t$  and  $G_3 \in G_{mt}^m$ , respectively. Hence,  $J_1^* = \{G_1, t'_1, t''_1\}$ ,  $J_2^* = \{G_2, t'_2, t''_2\}$ ,  $J_3^* = \{G_3, m_3, t_3\}$ . By Property 3.2.5 and the FFD rule,  $G_1$  must be packed with an item of type  $m$  before  $G_3$ . Since  $s_{m_3} < 1 - s_{G_3} - s_{t_3} - \frac{1}{2} - \frac{1}{5} = \frac{3}{10} < 2s_n$ ,  $m_3$  can be packed with  $G_1$  in the MFFD solution and  $J_1 = \{G_1, m_3\}$ . Next,  $G_2$  must be packed with an item of type  $t$  in  $J$  instead of an item of type  $m$ , then  $s_{G_2} + s_m \geq 1$ ,  $s_m \geq 1 - s_{G_2}$ . Since  $G_2$  is packed with two items of type  $t$  in  $J^*$ ,  $s_{G_2} < 1 - 2s_n$  and hence  $s_m \geq 2s_n$ . Thus, an item of type  $m$  must be packed with  $G_3$  in the MFFD solution,  $J_3 = \{G_3, m_3\}$  and there exists a bin with  $m_3$  in  $J^*$  in which no item of type  $G$  in it (because  $s_m + s_t + s_G \geq 2s_n + s_n + \frac{1}{2} > 1$ ). In addition, since  $s_m + 3s_t \geq 2s_n + 3s_n \geq 1$ , there are at most two items of type  $t$  in the bin containing an item of type  $m$  of size  $\geq 2s_n$ . Furthermore, in the MFFD solution, these  $t$  items from a non- $G$  bin and a  $t$  item from  $G_{mt}^m$  can be packed with  $i \in G_{tt}^t$ . Since  $m$  is packed with  $i \in G_{tt}^m$  before  $j \in G_{tt}^t$ , by the FFD rule,  $s_i > s_j$ , and hence an item of type  $t$ , which is packed with  $i \in G_{tt}^t$  in the MFFD solution, cannot come from the optimal bin with  $i \in G_{tt}^m$ .

Consider three cases: First, if there are two items of type  $t$  and one item of type  $m$  of size  $\geq 2s_n$  in  $J^*$ , then there are at most three portions of  $G_{tt}^t$  (one from the bin containing  $G_{mt}^m$  and two from the non- $G$  bins) with  $\sum_{i \in J^*} W(i) \leq \frac{5}{4}$ , one of  $G_{mt}^m$  with  $\sum_{i \in J^*} W(i) \leq \frac{5}{4}$ , one of  $G_{tt}^m$  with  $\sum_{i \in J^*} W(i) \leq \frac{7}{6}$  and one non- $G$  bin with  $\sum_{i \in J^*} W(i) \leq 1$ . Thus, the average weight is  $\leq \frac{43}{36}$ . Second, if there is at most one item of type  $t$  and one item of type  $m$  of size  $\geq 2s_n$  in  $J^*$ , then there are at most two portions of  $G_{tt}^t$  (one from the bin containing  $G_{mt}^m$  and one from the

non- $G$  bin) with  $\sum_{i \in J^*} W(i) \leq \frac{5}{4}$ , one of  $G_{mt}^m$  with  $\sum_{i \in J^*} W(i) \leq \frac{5}{4}$ , one of  $G_{tt}^m$  with  $\sum_{i \in J^*} W(i) \leq \frac{7}{6}$  and one non- $G$  bin with  $\sum_{i \in J^*} W(i) \leq \frac{7}{6}$ . Hence, the average weight is  $\leq \frac{73}{60}$ . Finally, if there are two items of type  $m$  of size  $\geq 2s_n$  in  $J^*$ , then there are at most two portions of  $G_{tt}^t$  from the bin containing  $G_{mt}^m$  with  $\sum_{i \in J^*} W(i) \leq \frac{5}{4}$ , two of  $G_{mt}^m$  with  $\sum_{i \in J^*} W(i) \leq \frac{5}{4}$ , two of  $G_{tt}^m$  with  $\sum_{i \in J^*} W(i) \leq \frac{7}{6}$  and one non- $G$  bin with  $\sum_{i \in J^*} W(i) \leq 1$ . Thus, the average weight is  $\leq \frac{25}{21}$ .

Since  $\frac{11}{9} > \frac{73}{60} > \frac{43}{36} > \frac{25}{21}$ , by Corollary 3.2.2,  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{67}{59} < \frac{91}{80}$ .

**Case 2.4:** If  $G_{tt}^t = \emptyset$  and  $G_{mt}^m = \emptyset$ , then  $\sum_{i \in J^*} W(i) \leq \frac{7}{6}$ . By Corollary 3.2.1,  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} < \frac{67}{59}$ . ■

### 3.2.7 If $\frac{1}{6} \leq s_n < \frac{1}{5}$ , then $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{91}{80}$ .

*Proof:* According to the FFD rule and discounting the large items that are assigned to the bins in Step 4, the MFFD algorithm will pack an item of type  $G$  with the other one or two smaller items, two items of type  $p_2$ , two items of type  $n_2$  with an item of type  $p_3$  or smaller, three items of type  $p_3$ , three items of type  $n_3$  with an item of type  $p_4$  or smaller, four items of type  $p_4$ , four items of type  $n_4$  with an item of type  $p_5$ , or five items of type  $p_5$ ; except the seven transition bins, which has total weight less than seven, other MFFD bins have total weight equal to one before Step 4. Recall that  $\alpha$  is the number of  $G$  bins in the optimal packing. Define an item of type non- $G$  bins of size  $\geq \frac{1-s_n}{3}$  as an item of type  $m$  (that is an item of type  $p_2$ ,  $n_2$  or  $p_3$ ). Furthermore, define an item of type non- $G$  bins of size  $< \frac{1-s_n}{3}$  as an item of type  $t$  (that is an item of type  $n_3$ ,  $p_4$ ,  $n_4$  or  $p_5$ ). Table 3.3 summarizes item sizes and weight for  $\frac{1}{6} \leq s_n < \frac{1}{5}$ .

Recall that  $W$  is the total weight of items in  $J^*$  from the integer program,  $W'$  is the total weight of items in  $J^*$  after the weight adjustment and  $W''$  is the average weight of items in



Table 3.3: Summary of item sizes and weight for  $\frac{1}{6} \leq s_n < \frac{1}{5}$ .

Items	Types	Range of Item Sizes	Item Weight	$v_i$
$n_1$		$[\frac{1}{2}, \frac{2}{3})$	$\frac{3}{4}$	1
$p_2$		$[\frac{1-s_n}{2}, \frac{1}{2})$	$\frac{1}{2}$	2
$n_2$		$[\frac{1}{3}, \frac{1-s_n}{2})$	$\frac{2}{5}$	2
$p_3$		$[\frac{1-s_n}{3}, \frac{1}{3})$	$\frac{1}{3}$	3
$n_3$		$[\frac{1}{4}, \frac{1-s_n}{3})$	$\frac{4}{15}$	3
$p_4$		$[\frac{1-s_n}{4}, \frac{1}{4})$	$\frac{1}{4}$	4
$n_4$		$[\frac{1}{5}, \frac{1-s_n}{4})$	$\frac{1}{5}$	4
$p_5$		$[s_n, \frac{1}{5})$	$\frac{1}{5}$	5

$J^*$  after the weight adjustment and matching. Then, consider the two cases as illustrated in Figure 3.8.

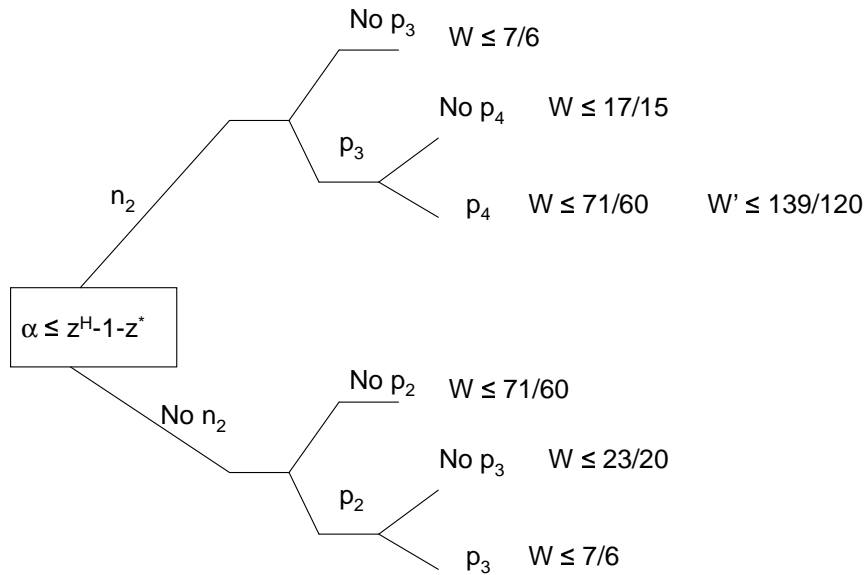


Figure 3.8: Summary of the weight adjustment when  $\frac{1}{6} \leq s_n < \frac{1}{5}$ .

**Case 1:**  $\alpha \leq z^H - 1 - z^*$

In this case, there is no  $G$  bin in the MFFD solution. We can partition the cases as shown in Figure 3.8. There is the weight adjustment as follows: If there exists  $J^*$  with one item of type  $n_2$ , one of type  $p_3$ , and one of type  $p_4$ , then  $\sum_{i \in J^*} W(i) \leq \frac{71}{60}$ . Since  $s_{p_3} + s_n \geq \frac{1-s_n}{3} + s_n > \frac{1-s_n}{2} = s_{n_2}$ , in the MFFD solution, two of smaller size of  $n_2$  can be packed with an item of type  $p_4$ . Thus, the total weight is  $2(\frac{2}{5}) + \frac{1}{4} = 1 + \frac{1}{20}$ . We can adjust the weight with  $\frac{1}{2}(\frac{1}{20}) = \frac{1}{40}$ . After adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{71}{60} - \frac{1}{40} = \frac{139}{120}$ . Hence,  $\sum_{i \in J^*} W(i) \leq \frac{71}{60}$  without an item of type  $G$ . By Lemma 3.2.4, we have

$$\begin{aligned}
z^H - 7 &\leq \sum_{b=1}^{z^*} \sum_{i \in J_b^*} W(i) - \sum_{i=z^*+1}^{z^H-1} W(i) - \sum_{b=1}^{z^*} (\sum_{i \in J_b} W(i) - 1). \\
&\leq \alpha \left( \frac{4}{3} - \frac{3}{4} \right) + \frac{71}{60} (2z^* - z^H + 1) + \left( \frac{71}{60} - \frac{1}{3} \right) (z^H - z^* - \alpha - 1). \\
&\leq \frac{7}{12} \alpha + \frac{71}{60} (2z^* - z^H) + \frac{51}{60} (z^H - z^* - \alpha) + \frac{1}{3}. \\
&\leq \frac{71}{60} (2z^* - z^H) + \frac{51}{60} (z^H - z^*) + \frac{1}{3}. \\
\frac{80}{60} z^H &\leq \frac{91}{60} z^* + \frac{22}{3}.
\end{aligned}$$

Then,  $\frac{z^H}{z^*} \leq \frac{91}{80} + \frac{22}{3z^*}$  and  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{91}{80}$ . Note that the asymptotic worst-case ratio is tight.

**Case 2:**  $\alpha > z^H - 1 - z^*$

According to the definition of  $G_Y^X$  as specified in a previous lemma. Then, consider the following matching:

1. If a  $G$  item is packed with an item of type  $m$  and  $t$  in  $J^*$ , then  $\sum_{i \in J^*} W(i) \leq \frac{4}{3}$ . Then, consider the MFFD solution as the following possibilities:

- (a) If  $G$  is packed with an item of type  $m$  and  $t$  in  $J$ , then  $\sum_{i \in J^*} W(i) \leq \frac{4}{3}$  and  $\sum_{i \in J} W(i) \geq$

$\frac{77}{60} = 1 + \frac{17}{60}$ . We adjust the weight of the optimal bin by  $\frac{17}{60}$ . After the adjustment,

$$\sum_{i \in J^*} W(i) \leq \frac{4}{3} - \frac{17}{60} = \frac{21}{20}. \text{ Hence, for any } i \in G_{mt}^m, \sum_{i \in J^*} W(i) \leq \frac{21}{20}.$$

(b) If  $G$  is packed with an item of type  $m$  in  $J$ , then  $\sum_{i \in J} W(i) \geq \frac{13}{12} = 1 + \frac{1}{12}$ . We adjust

the weight of the optimal bin by  $\frac{1}{12}$ . After the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{4}{3} - \frac{1}{12} = \frac{5}{4}$ .

Hence, for any  $i \in G_{mt}^m$ ,  $\sum_{i \in J^*} W(i) \leq \frac{5}{4}$ .

(c) If  $G$  is packed with two items of type  $t$  in  $J$ , then  $\sum_{i \in J} W(i) \geq \frac{23}{20} = 1 + \frac{3}{20}$ . We adjust

the weight of the optimal bin by  $\frac{3}{20}$ . After the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{4}{3} - \frac{3}{20} = \frac{71}{60}$ .

Hence, for any  $i \in G_{mt}^{tt}$ ,  $\sum_{i \in J^*} W(i) \leq \frac{71}{60}$ .

(d) If  $G$  is packed with one item of type  $t$  in  $J$ . Similar to previous section. This case

does not exist. Thus,  $G_{mt}^t = \emptyset$

2. If a  $G$  item is packed with two items of type  $t$  in  $J^*$ , then  $\sum_{i \in J^*} W(i) \leq \frac{19}{15}$ . Then, consider

the MFFD solution as the following possibilities:

(a) If there is an item of type  $n_3$  in  $J^*$ , then consider the following cases:

i. If  $G$  is packed with an item of type  $m$  and  $t$  in  $J$ , then  $\sum_{i \in J} W(i) \geq \frac{77}{60} = 1 + \frac{17}{60}$ .

After the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{19}{15} - \frac{17}{60} = \frac{59}{60}$ .

ii. If  $G$  is packed with an item of type  $m$  in  $J$ , then  $\sum_{i \in J} W(i) \geq \frac{3}{4} + \frac{1}{3} = 1 + \frac{1}{12}$ .

Then, we can adjust  $\frac{1}{12}$  from  $\frac{19}{15}$ . After the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{19}{15} - \frac{1}{12} = \frac{71}{60}$ .

iii. If  $G$  is packed with two items of type  $t$  in  $J$ , then  $\sum_{i \in J} W(i) \geq \frac{23}{20} = 1 + \frac{3}{20}$ . After

the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{19}{15} - \frac{3}{20} = \frac{67}{60}$ .

iv. If  $G$  is packed with an item of type  $t$  in  $J$ , then  $\sum_{i \in J} W(i) \leq \frac{61}{60} = 1 + \frac{1}{60}$ . After

the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{19}{15} - \frac{1}{60} = \frac{5}{4}$ .

(b) If there is no item of type  $n_3$  in  $J^*$ , then  $\sum_{i \in J^*} W(i) \leq \frac{5}{4}$ . Next, consider the following cases:

i. If  $G$  is packed with an item of type  $m$  and  $t$  in  $J$ , then  $\sum_{i \in J} W(i) \geq \frac{77}{60} = 1 + \frac{17}{60}$ .

After the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{5}{4} - \frac{17}{60} = \frac{58}{60}$ .

ii. If  $G$  is packed with two items of type  $m$  in  $J$ , then  $\sum_{i \in J} W(i) \geq \frac{13}{12}$ . Then, after

the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{5}{4} - \frac{1}{12} = \frac{7}{6}$ .

iii. If  $G$  is packed with two items of type  $t$  in  $J$ , then  $\sum_{i \in J} W(i) \geq \frac{23}{20} = 1 + \frac{3}{20}$ . After

the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{5}{4} - \frac{3}{20} = \frac{11}{10}$ .

iv. If  $G$  is packed with an item of type  $t$  in  $J$ , then there are two cases: First, if there exists an item of type  $p_4$  in  $J^*$ , then  $\sum_{i \in J^*} W(i) \leq \frac{5}{4}$ . Second, if there does not

exist an item of type  $p_4$  in  $J^*$ , then  $\sum_{i \in J^*} W(i) \leq \frac{6}{5}$  and  $\sum_{i \in J} W(i) \geq \frac{19}{20} = 1 - \frac{1}{20}$ .

After the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{6}{5} - (-\frac{1}{20}) = \frac{5}{4}$ .

The summary of the weight adjustment is illustrated in Figure 3.9. After the adjustment,

$\sum_{i \in J^*} W(i) \leq \frac{5}{4}$  with an item of type  $G$  and  $\sum_{i \in J^*} W(i) \leq \frac{71}{60}$  without an item of type  $G$ .

Then, consider the matching as illustrated in Figure 3.10. Similar to previous lemma, we consider the following cases:

**Case 2.1:**  $G_{tt}^t \neq \emptyset$  and  $G_{mt}^m = \emptyset$

Similar to Case 2.1 of Section 3.2.6, there are two cases: First, if an item of type  $t$  in  $J$  is from the bin with at most two  $t$  in  $J^*$ , then at most two  $G$  bins can be packed with these items. Thus, the average weight is  $\leq \frac{11}{9}$ . Second, if an item of type  $t$  in  $J$  is from the bin with

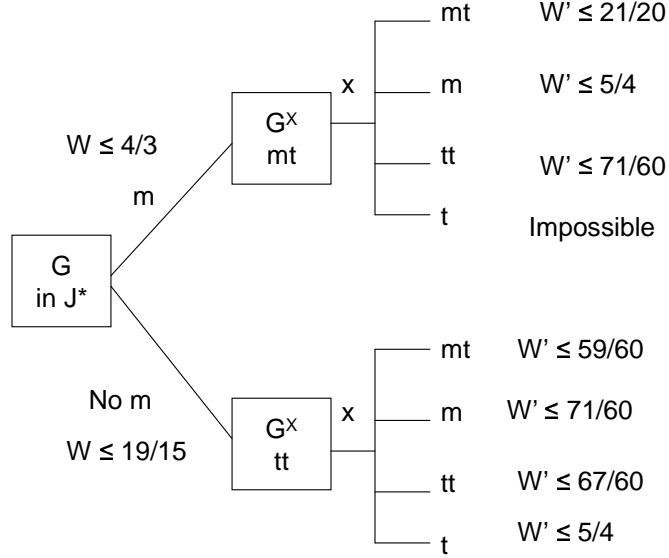


Figure 3.9: Summary of the weight adjustment when  $\frac{1}{6} \leq s_n < \frac{1}{5}$ .

more than two items of type  $t$  in  $J^*$ , then the average weight is  $\leq \frac{49}{40}$ . Since,  $\frac{49}{40} > \frac{11}{9}$ , then

$\sum_{i \in J^*} W(i) \leq \frac{49}{40}$ . By Lemma 3.2.4, we have

$$\begin{aligned}
z^H - 7 &\leq \sum_{b=1}^{z^*} \sum_{i \in J_b^*} W(i) - \sum_{i=z^*+1}^{z^H-1} W(i) - \sum_{b=1}^{z^H} (\sum_{i \in J_b} W(i) - 1). \\
&\leq \frac{5}{4}(\alpha - (z^H - z^* - 1)) + \frac{71}{60}(z^* - \alpha) + \left(\frac{4}{3} - \frac{3}{4}\right)(z^H - z^* - 1), \\
&\leq \frac{49}{40}(2z^* - z^H) + \frac{7}{12}(z^H - z^*) + \frac{2}{3}. \\
&\leq \frac{224}{120}z^* - \frac{57}{120}z^H + \frac{2}{3}.
\end{aligned}$$

Then,  $\frac{z^H}{z^*} \leq \frac{224}{197} + \frac{23}{3z^*}$  and  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{224}{197} < \frac{91}{80}$ .

**Corollary 3.2.3** *If  $\sum_{i \in J^*} W(i) \leq \frac{49}{40}$ , then  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{224}{197} < \frac{91}{80}$ .*

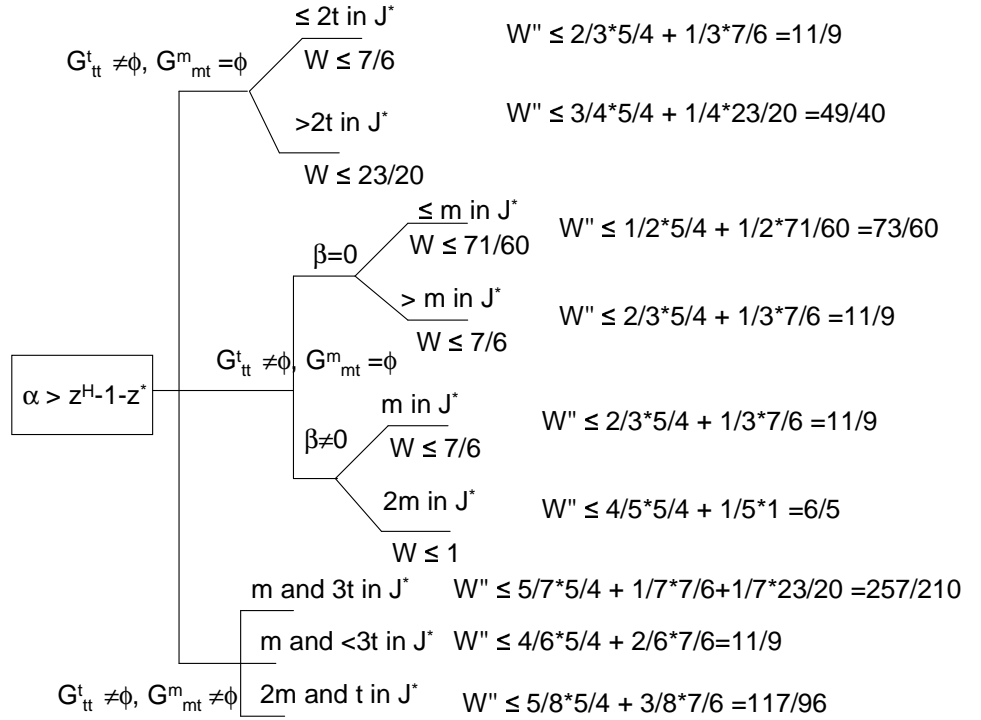


Figure 3.10: Summary of the matching when  $\frac{1}{6} \leq s_n < \frac{1}{5}$ .

**Case 2.2:**  $G_{mt}^m \neq \emptyset$  and  $G_{tt}^t = \emptyset$

Recall that  $\beta$  is the number of items of type  $m$  belonging to both  $j \in J_{u^*(i)}^* : i \in G_{mt}^m$  and  $j \in J_{u(i)} : i \in G_{mt}^m$ . Since  $G$  is not packed with an item of type  $t$  in the MFFD solution, then it must be packed with an item of type  $m$  only. Consider the following cases:

A. If  $\beta = 0$ , then there is no item of type  $m$  belonging to both  $\{j \in J_{u^*(i)}^* : i \in G_{mt}^m\}$  and  $\{j \in J_{u(i)} : i \in G_{mt}^m\}$ . By the integer program, we have  $\sum_{i \in J^*} W(i) \leq \frac{71}{60}$  if there is no item of type  $G$  but at most one item of type  $m$  in  $J^*$ ,  $\sum_{i \in J^*} W(i) \leq \frac{7}{6}$  if there is no item of type  $G$  but at most two items of type  $m$  in  $J^*$ , and  $\sum_{i \in J^*} W(i) \leq 1$  if there are three items of type  $m$  in  $J^*$ . If there are two items of type  $m$  in  $J^*$ , then  $\sum_{i \in J^*} W(i) \leq \frac{7}{6}$ .

Consider two cases: First, if an item of type  $m$  in  $J$  is from the bin with at least two items

of type  $m$  in  $J^*$ , then the average weight is  $\leq \frac{11}{9}$ . Second, if an item of type  $m$  in  $J$  is from the bin with one item of type  $m$  in  $J^*$ , then the average weight is at most  $\frac{73}{60}$ . Since  $\frac{49}{40} > \frac{11}{9} > \frac{73}{60}$ , by Corollary 3.2.3,  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} < \frac{91}{80}$ .

B. If  $\beta \neq 0$ , then there must exist at least two sets of  $G_{mt}^m$  bin and one of them contains an item  $k$ . Let  $J_1^* = \{G_1, m_1, t_1\}$  and  $J_2^* = \{G_2, k, t_2\}$ . Then,  $J_1 = \{G_1, k\}$ . Since  $k$  is packed with  $G_1$  only in the MFFD solution but with  $G_2$  and  $t_2$  in the optimal solution,  $s_{G_1} > s_{G_2}$ . For  $J_2 = \{G_2, m_2\}$ , by the FFD rule,  $m_2$  must be too large to be packed with  $G_1$ ,  $s_{G_1} + s_{m_2} \geq 1$ . In addition,  $s_{G_1} < 1 - s_{m_1} - s_{t_1} < 1 - \frac{1-s_n}{3} - s_n$ , then  $s_{m_2} \geq 1 - s_{G_1} = \frac{1-s_n}{3} + s_n = \frac{5}{18} + \frac{1}{6} = \frac{4}{9}$ . Thus,  $m_2$  is not from the  $G$  bin because  $s_G + s_{m_2} + s_t \geq \frac{1}{2} + \frac{4}{9} + \frac{1}{6} > 1$ . There are at most two items of type  $m$  of size  $\geq \frac{4}{9}$  in  $J^*$ .

Hence, there are two cases: First, if an item of type  $m$  in  $J$  is from the bin with two items of type  $m$  of size  $\geq \frac{4}{9}$  in the non- $G$  bin in  $J^*$ , then there are at most two portions of  $G_{mt}^m$  with an item  $k$  and two portions of  $G_{mt}^m$  without an item  $k$  and one non- $G$  bin. Thus, the average weight is  $\leq \frac{6}{5}$ . Second, if an item of type  $m$  in  $J$  is from the bin with at most one item of type  $m$  of size  $\geq \frac{4}{9}$  in the non- $G$  bin in  $J^*$ , then there are at most one portion of  $G_{mt}^m$  with an item  $k$  and one portion of  $G_{mt}^m$  without an item  $k$  and one non- $G$  bin. Thus, the average weight is  $\leq \frac{11}{9}$ . Since  $\frac{49}{40} > \frac{11}{9} > \frac{6}{5}$ , by Corollary 3.2.3,  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} < \frac{91}{80}$ .

**Case 2.3:  $G_{mt}^m \neq \emptyset$  and  $G_{tt}^t \neq \emptyset$**

Similar to Case 2.3 in Section 3.2.6, consider one portion of a set of  $G_{tt}^m$ ,  $G_{tt}^t$  and  $G_{mt}^m$ . Let  $G_1 \in G_{tt}^m$ ,  $G_2 \in G_{tt}^t$  and  $G_3 \in G_{mt}^m$ , respectively. Hence,  $J_1^* = \{G_1, t'_1, t''_1\}$ ,  $J_2^* = \{G_2, t'_2, t''_2\}$ ,  $J_3^* = \{G_3, m_3, t_3\}$ . By Property 3.2.5 and the FFD rule,  $G_1$  must be packed with an item of type  $m$  before  $G_3$ . Since  $s_{m_3} < 1 - s_{G_3} - s_{t_3} = 1 - \frac{1}{2} - \frac{1}{6} = \frac{1}{3} \leq 2s_n$ ,  $m_3$  can be packed with

$G_1$  in the MFFD solution and  $J_1 = \{G_1, m_3\}$ . Next,  $G_2$  must be packed with an item of type  $t$  in  $J$  instead of an item of type  $m$ , then  $s_{G_2} + s_m \geq 1$ ,  $s_m \geq 1 - s_{G_2}$ . Since  $G_2$  is packed with two items of type  $t$  in  $J^*$ ,  $s_{G_2} < 1 - 2s_n$  and hence  $s_m \geq 2s_n$ . Thus, an item of type  $m$  must be packed with  $G_3$  in the MFFD solution,  $J_3 = \{G_3, m_3\}$  and there exists a bin with  $m_3$  in  $J^*$  in which no item of type  $G$  in it (because  $s_m + s_t + s_G \geq 2s_n + s_n + \frac{1}{2} > 1$ ). In addition, since  $s_m + 4s_t \geq 2s_n + 4s_n \geq 1$ , there are at most three items of type  $t$  in the bin containing an item of type  $m$  of size  $\geq 2s_n$ . Furthermore, these  $t$  items from a non- $G$  bin and a  $t$  item from  $J_3^*$  can be packed with  $i \in G_{tt}^t$ . Similarly, if there are two items of type  $m$  of size  $> 2s_n$ , then at most one item of type  $t$  can be in the bin.

Hence, there are three cases: First, if there are three items of type  $t$  and one item of type  $m$  of size  $\geq 2s_n$  in  $J^*$ , then at most four portions of  $G_{tt}^t$  (one from a  $G_{mt}^m$  and three from the non- $G$  bin), one of  $G_{mt}^m$ , one of  $G_{tt}^m$  and one non- $G$  bin containing three items of type  $t$  and an item of type  $m$ . Thus, the average weight is  $\leq \frac{257}{210}$ . Second, if there are at most two items of type  $t$  and one item of type  $m$  of size  $\geq 2s_n$  in  $J^*$ , then at most three portions of  $G_{tt}^t$  (one from a  $G_{mt}^m$  and two from the non- $G$  bins), one of  $G_{mt}^m$ , one of  $G_{tt}^m$  and one non- $G$  bin containing two items of type  $t$  and an item of type  $m$  of size  $\geq 2s_n$ . Thus, the average weight is at most  $\frac{11}{9}$ . Finally, if there are two items of type  $m$  of size  $\geq 2s_n$  and one item of type  $t$  in  $J^*$ , then at most three portions of  $G_{tt}^t$  (two from a  $G_{mt}^m$  and one from the non- $G$  bin), two of  $G_{mt}^m$ , two of  $G_{tt}^m$  and one non- $G$  bin containing an item of type  $t$  and two items of type  $m$  of size  $\geq 2s_n$ . Thus, the average weight is  $\leq \frac{117}{96}$ .



Since  $\frac{257}{210} > \frac{11}{9} > \frac{117}{96}$ , then the average weight is  $\leq \frac{257}{210}$ . By Lemma 3.2.4, we have

$$\begin{aligned} z^H - 7 &\leq \sum_{b=1}^{z^*} \sum_{i \in J_b^*} W(i) - \sum_{i=z^*+1}^{z^H-1} W(i) - \sum_{b=1}^{z^H} \left( \sum_{i \in J_b} W(i) - 1 \right). \\ &\leq \frac{257}{210} (2z^* - z^H) + \frac{7}{12} (z^H - z^*) + \frac{2}{3}. \\ &\leq \frac{783}{420} z^* - \frac{269}{420} z^H + \frac{2}{3}. \end{aligned}$$

Then,  $\frac{z^H}{z^*} \leq \frac{783}{689} + \frac{23}{3z^*}$  and  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{783}{689} < \frac{91}{80}$ .

**Case 2.4:** If  $G_{tt}^t = \emptyset$  and  $G_{mt}^m = \emptyset$ , then  $\sum_{i \in J^*} W(i) \leq \frac{71}{60}$ . Since  $\frac{49}{40} > \frac{71}{60}$ , by Corollary 3.2.3,

$$\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} < \frac{91}{80}. \quad \blacksquare$$

**3.2.8** If  $\frac{1}{7} \leq s_n < \frac{1}{6}$ , then  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{91}{80}$ .

*Proof:* According to the FFD rule and discounting the large items that are assigned to the bins in Step 4, the MFFD algorithm will pack two items of type  $p_2$ , two items of type  $n_2$  with a smaller item, three items of type  $p_3$ , three items of type  $n_3$  with a smaller item, four items of type  $p_4$ , four items of type  $n_4$  with a smaller item, five items of type  $p_5$ , five items of item type  $n_5$  with a smaller item, or six items of type  $p_6$ ; except the nine transition bins.

Hence, by the weight assigned in Table 3.4,  $\sum_{i \in J} W(i) \geq 1$ . From the integer program, we have

$\sum_{i \in J^*} W(i) \leq \frac{4}{3}$ ,  $\sum_{i \in J^*} W(i) \leq \frac{6}{5}$  without an item of type  $G$ ,  $\sum_{i \in J^*} W(i) \leq \frac{71}{60}$  without an item of type  $G$  or  $n_2$ , and  $\sum_{i \in J^*} W(i) \leq \frac{7}{6}$  without an item of type  $G$  or  $p_5$ . Recall that  $W$  is the total weight of items in  $J^*$  from the integer program and  $W'$  is the total weight of items in  $J^*$  after the weight adjustment. Then, consider the weight adjustment as illustrated in Figure 3.11.

Then, consider the following cases:

**Case 1:**  $\alpha \leq z^H - 1 - z^*$

Table 3.4: Summary of item sizes and weight for  $\frac{1}{7} \leq s_n < \frac{1}{6}$ .

Items	Types	Range of Item Sizes	Item Weight	$v_i$
$n_1$		$[\frac{1}{2}, \frac{5}{7})$	$\frac{3}{4}$	1
$p_2$		$[\frac{1-s_n}{2}, \frac{1}{2})$	$\frac{1}{2}$	2
$n_2$		$[\frac{1}{3}, \frac{1-s_n}{2})$	$\frac{5}{12}$	2
$p_3$		$[\frac{1-s_n}{3}, \frac{1}{3})$	$\frac{1}{3}$	3
$n_3$		$[\frac{1}{4}, \frac{1-s_n}{3})$	$\frac{5}{18}$	3
$p_4$		$[\frac{1-s_n}{4}, \frac{1}{4})$	$\frac{1}{4}$	4
$n_4$		$[\frac{1}{5}, \frac{1-s_n}{4})$	$\frac{5}{24}$	4
$p_5$		$[\frac{1-s_n}{5}, \frac{1}{5})$	$\frac{1}{5}$	5
$n_5$		$[\frac{1}{6}, \frac{1-s_n}{5})$	$\frac{1}{6}$	5
$p_6$		$[s_n, \frac{1}{6})$	$\frac{1}{6}$	6

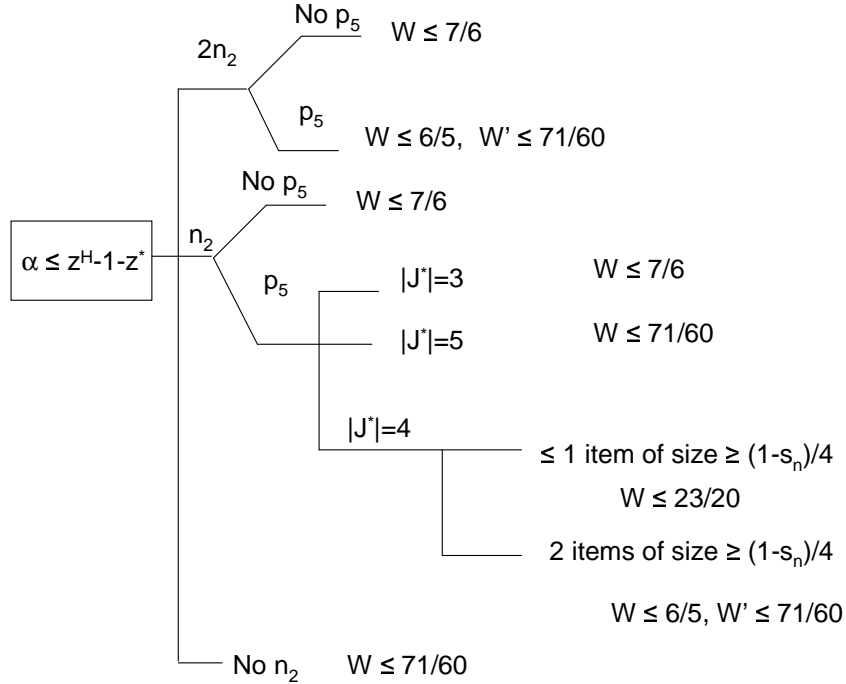


Figure 3.11: Summary of the weight adjustment when  $\frac{1}{7} \leq s_n < \frac{1}{6}$ .

Consider the following adjustments: First, if there are two items of type  $n_2$  and one item of type  $p_5$  in  $J^*$ , then  $\sum_{i \in J^*} W(i) \leq \frac{6}{5}$ . Since in the MFFD solution, the MFFD can pack two items

of a smaller size of type  $n_2$  with  $p_5$ , then  $\sum_{i \in J} W(i) = 2(\frac{5}{12}) + \frac{1}{5} = 1 + \frac{1}{30}$ . By Lemma 3.2.4, we can adjust the weight so that  $\sum_{i \in J} W(i) = 1$ . Then, we adjust the weight of the smaller size of type  $n_2$  in  $J^*$ . Since each optimal bin has one item of a smaller size of  $n_2$ , we adjust half of the number of items of type  $n_2$  with  $\frac{1}{30}$ . Hence, we subtract  $\frac{1}{2}(\frac{1}{30})$  from the maximum weight. After the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{6}{5} - \frac{1}{2}(\frac{1}{30}) = \frac{71}{60}$ .

Second, if there are four items in  $J^*$  with one item of type  $n_2$ , one of type  $p_5$  and the remaining items are of size  $\geq \frac{1-s_n}{4}$ , then  $\sum_{i \in J^*} W(i) \leq \frac{6}{5}$ . Since the total space of the remaining two items is at least  $2(\frac{1-s_n}{4}) = \frac{1-s_n}{2}$ , then the space of a bin is greater than the size of an item of type  $n_2$ , this implies that two items of type  $n_2$  and one item of type  $p_5$  can be packed in the same bin. Hence, in the MFFD solution, two items of a smaller size of type  $n_2$  can be packed with an item of type  $p_5$ . Then,  $\sum_{i \in J} W(i) = 2(\frac{5}{12}) + \frac{1}{5} = 1 + \frac{1}{30}$ . In addition, because each optimal bin has one item of type  $n_2$ , we adjust half of the number of items of type  $n_2$  with  $\frac{1}{30}$ . Hence, we subtract  $\frac{1}{2}(\frac{1}{30})$  from the maximum weight. After adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{6}{5} - \frac{1}{60} = \frac{71}{60}$ .

Let  $\kappa_1$  be the number of the optimal bins with two items of type  $n_2$ , one item of type  $p_5$  and one more item,  $\kappa_2$  be the number of the optimal bins with one item of type  $n_2$ , one of type  $p_5$  and the remaining items are of size  $\geq \frac{1-s_n}{4}$ . If  $\kappa_1 > 0$  and  $\kappa_2 > 0$ , then the number of the MFFD bins with two items of type  $n_2$  and one item of type  $p_5$  is  $\frac{\kappa_1 + \kappa_2}{2}$ . This does not affect the weight adjustment. After the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{71}{60}$  without an item of type  $G$ . By Lemma 3.2.4, we have

$$\begin{aligned} z^H - 9 &\leq \sum_{b=1}^{z^*} \sum_{i \in J_b^*} W(i) - \sum_{i=z^*+1}^{z^H-1} W(i) - \sum_{b=1}^{z^H} (\sum_{i \in J_b} W(i) - 1). \\ &\leq \alpha \left( \frac{4}{3} - \frac{3}{4} \right) + \frac{71}{60} (2z^* - z^H + 1) + \left( \frac{71}{60} - \frac{1}{3} \right) (z^H - 1 - z^* - \alpha). \end{aligned}$$

$$\begin{aligned}
&\leq \frac{7}{12}\alpha + \frac{71}{60}(2z^* - z^H) + \frac{51}{60}(z^H - z^* - \alpha) + \frac{1}{3}. \\
&\leq \frac{71}{60}(2z^* - z^H) + \frac{51}{60}(z^H - z^*) + \frac{1}{3}. \\
\frac{4}{3}z^H - 9 &\leq \frac{91}{60}z^* \frac{1}{3}.
\end{aligned}$$

Note that  $\frac{71}{60}(2z^* - z^H) + (\frac{71}{60} - \frac{1}{3})(z^H - z^*) \geq \frac{23}{20}(2z^* - z^H) + (\frac{23}{20} - \frac{1}{4})(z^H - z^*)$ . Hence,  $\frac{z^H}{z^*} \leq \frac{91}{80} + \frac{28}{3z^*}$  and  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{91}{80}$ .

**Case 2:**  $\alpha > z^H - 1 - z^*$

In this case, we can apply Lemma 3.1.1 (ii) instead of using the weighting function like previous lemmas. Let  $m = \lfloor \frac{z^H}{9} \rfloor$ , and set  $z^H = 9m + j$ , where  $j = \{0, 1, \dots, 8\}$ . The idea is that if  $\alpha > z^H - 1 - z^*$  and  $\frac{1}{7} \leq s_n < \frac{1}{6}$ , then the total size of the last  $m$  bin is large enough to fill the space of the first  $8m$  bins.

Consider the total size of bins except the largest items. By the FFD rule, for  $s_n < \frac{1}{6}$ , it implies that the last item cannot be packed in any previous bin in which  $\sum_{i \in I_b} s_i > \frac{5}{6}$ . If the total size of the items excluding the first  $8m$  items is at least  $8m - 2$ , then by Lemma 3.1.1 (ii), the number of bins required is at least  $8m - 1$ . The detail is as follows.

$$\begin{aligned}
\sum_{i=8m+1}^n s_i &= \sum_{b=1}^{8m} (\sum_{i \in I_b} s_i - s_b) + \sum_{b=8m+1}^{z^H} \sum_{i \in I_b} s_i. \\
&> 8m(1 - s_n) + (z^H - 1 - 8m)(1 - s_n + \frac{1}{2}). \\
&> 8m(1 - \frac{1}{6}) + (m - 1)(\frac{3}{2} - \frac{1}{6}). \\
&= 8m - \frac{4}{3} \geq 8m - 2.
\end{aligned}$$

By Lemma 3.1.1(ii),  $z^* \geq 8m - 1$ . Thus,  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{9m+j}{8m-1} = \frac{9}{8}$ . ■

**3.2.9** If  $\frac{1}{8} \leq s_n < \frac{1}{7}$ , then  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{91}{80}$ .

*Proof:* According to the FFD rule and discounting the large items that are assigned to the bins in Step 4, the MFFD algorithm will pack two items of type  $p_2$ , two items of type  $n_2$  with a smaller item, three items of type  $p_3$ , three items of type  $n_3$  with a smaller item, four items of type  $p_4$ , four items of type  $n_4$  with a smaller item, five items of item type  $p_5$ , five items of type  $n_5$  with a smaller item, six items of type  $p_6$ , six items of type  $n_6$  with a smaller item, or seven items of type  $p_7$ ; except eleven transition bins. Hence, by the weight assigned in Table 3.5,  $\sum_{i \in J} W(i) \geq 1$ . From the integer program, we have  $\sum_{i \in J^*} W(i) \leq \frac{113}{84}$  with an item of type  $G$  in  $J^*$ ,  $\sum_{i \in J^*} W(i) \leq \frac{257}{210}$  without an item of type  $G$ , but two items of type  $n_2$  in  $J^*$ ,  $\sum_{i \in J^*} W(i) \leq \frac{251}{210}$  without an item of type  $G$  and at most one item of type  $n_2$  in  $J^*$ , and  $\sum_{i \in J^*} W(i) \leq \frac{23}{20}$  without an item of type  $G$  or  $p_5$  in  $J^*$ . Recall that  $W$  is the total weight of items in  $J^*$  from the

Table 3.5: Summary of item sizes and weight for  $\frac{1}{8} \leq s_n < \frac{1}{7}$ .

Items	Types	Range of Item Sizes	Item Weight	$v_i$
$n_1$		$(\frac{1}{2}, \frac{3}{4})$	$\frac{3}{4}$	1
$p_2$		$(\frac{1-s_n}{2}, \frac{1}{2})$	$\frac{1}{2}$	2
$n_2$		$(\frac{1}{3}, \frac{1-s_n}{2})$	$\frac{6}{14}$	2
$p_3$		$(\frac{1-s_n}{3}, \frac{1}{3})$	$\frac{1}{3}$	3
$n_3$		$(\frac{1}{4}, \frac{1-s_n}{3})$	$\frac{6}{21}$	3
$p_4$		$(\frac{1-s_n}{4}, \frac{1}{4})$	$\frac{1}{4}$	4
$n_4$		$(\frac{1}{5}, \frac{1-s_n}{4})$	$\frac{6}{28}$	4
$p_5$		$(\frac{1-s_n}{5}, \frac{1}{5})$	$\frac{1}{5}$	5
$n_5$		$(\frac{1}{6}, \frac{1-s_n}{5})$	$\frac{6}{35}$	5
$p_6$		$(\frac{1-s_n}{6}, \frac{1}{6})$	$\frac{1}{6}$	6
$n_6$		$(\frac{1}{7}, \frac{1-s_n}{6})$	$\frac{1}{7}$	6
$p_7$		$(s_n, \frac{1}{7})$	$\frac{1}{7}$	7

integer program and  $W'$  is the total weight of items in  $J^*$  after the weight adjustment. Then, we consider the weight adjustment as illustrated in Figure 3.12. Let  $\lambda$  be the number of items of size  $\geq \frac{1-s_n}{2}$ . Consider the following cases:

**Case 1:**  $\lambda \leq z^H - 1 - z^*$

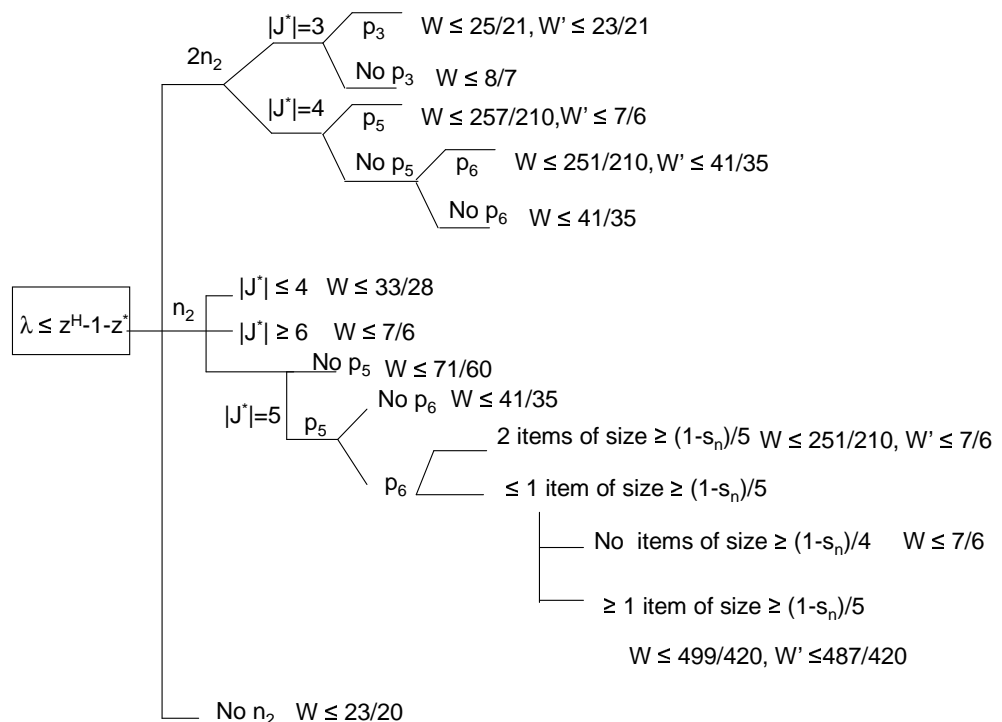


Figure 3.12: Summary of the weight adjustment when  $\frac{1}{8} \leq s_n < \frac{1}{7}$ .

Consider the following adjustments: First, if there are at most three items in  $J^*$  with two items of type  $n_2$  and one of type  $p_3$ , then  $\sum_{i \in J^*} W(i) \leq \frac{25}{21}$ . Hence, in the MFFD solution, two items of a smaller size of type  $n_2$  can be packed with an item of type  $p_3$ . Then,  $\sum_{i \in J} W(i) = 2(\frac{6}{14}) + \frac{1}{3} = 1 + \frac{4}{21}$ . Since each optimal bin has one item of a smaller size of  $n_2$ , we adjust half of the number of items of type  $n_2$  with  $\frac{4}{21}$ . Hence, we subtract  $\frac{1}{2}(\frac{4}{21})$  from the maximum weight. After the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{25}{21} - \frac{1}{2}(\frac{4}{21}) = \frac{23}{21}$ .

Second, if there are four items in  $J^*$  with two items of type  $n_2$ , one item of type  $p_5$  and one item of type  $p_6$ , then  $\sum_{i \in J^*} W(i) \leq \frac{257}{210}$ . We calculate the maximum size of  $n_2$ ,  $s_{n_2} \leq 1 - s_{n_2} - s_{p_5} - s_n = 1 - \frac{1}{3} - \frac{1-s_n}{5} - s_n = \frac{7}{15} - \frac{4s_n}{5}$ . Since the space after packing two items of the largest size of item of type  $n_2$  is at least  $1 - 2(\frac{7}{15} - \frac{4s_n}{5}) = \frac{1}{15} + \frac{8s_n}{5} = \frac{1}{15} + \frac{8(\frac{1}{8})}{5} = \frac{4}{15} > s_{p_5}$ . This implies that in the MFFD solution, two items of the largest size of an item type  $n_2$  and one item of type  $p_5$  can be packed in the same bin. Then,  $\sum_{i \in J} W(i) = 2(\frac{6}{14}) + \frac{1}{5} = 1 + \frac{2}{35}$ . Hence, we subtract  $\frac{2}{35}$  from the weight. After the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{257}{210} - \frac{2}{35} = \frac{7}{6}$ .

Third, if there are four items in  $J^*$  with two items of type  $n_2$ , one item of type  $p_6$ , but no item of type  $p_5$ , then  $\sum_{i \in J^*} W(i) \leq \frac{251}{210}$ . We calculate the maximum size of  $n_2$ ,  $s_{n_2} \leq 1 - s_{n_2} - s_{p_6} - s_n = 1 - \frac{1}{3} - \frac{1-s_n}{6} - s_n \geq 1 - \frac{1}{3} - \frac{1-\frac{1}{8}}{6} - \frac{1}{8} = \frac{19}{48}$ . Since the space after packing two items of the largest size of item of type  $n_2$  is at least  $1 - 2(\frac{19}{48}) = \frac{5}{24} > s_{p_6}$ . This implies that in the MFFD solution, two items of the largest size of an item type  $n_2$  and one item of type  $p_6$  can be packed in the same bin. Then,  $\sum_{i \in J} W(i) = 2(\frac{6}{14}) + \frac{1}{6} = 1 + \frac{1}{42}$ . Hence, we subtract  $\frac{1}{42}$  from the maximum weight. After the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{251}{210} - \frac{1}{42} = \frac{41}{35}$ .

Next, if there are five items in  $J^*$  with one item of type  $n_2$ ,  $p_5$ ,  $p_6$  and the remaining items are of size  $\geq \frac{1-s_n}{5}$ , then  $\sum_{i \in J^*} W(i) \leq \frac{251}{210}$ . Since the total space of  $p_6$  and the remaining two items is at least  $\frac{1-s_n}{6} + 2(\frac{1-s_n}{5}) > \frac{1-s_n}{2}$ . Then, the space of a bin is greater than the size of item type  $n_2$ , this implies that two items of type  $n_2$  and one item of type  $p_5$  can be packed in the same bin. Hence, in the MFFD solution, two items of a smaller size of type  $n_2$  can be packed with an item of type  $p_5$ . Then,  $\sum_{i \in J} W(i) = 2(\frac{6}{14}) + \frac{1}{5} = 1 + \frac{2}{35}$ . Since each optimal bin has one item of a smaller size of  $n_2$ , we adjust half of the number of item type  $n_2$  with  $\frac{2}{35}$ . Hence, we subtract  $\frac{1}{2}(\frac{2}{35})$  from the maximum weight. After the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{251}{210} - \frac{1}{2}(\frac{2}{35}) = \frac{7}{6}$ .

Finally, if there are five items in  $J^*$  with one item of type  $n_2, p_5, p_6$  and at most one of the remaining items is of size  $\geq \frac{1-s_n}{4}$ , then  $\sum_{i \in J^*} W(i) \leq \frac{499}{420}$ . Since the total space of  $p_6$  and the remaining items is at least  $\frac{1-s_n}{4} + \frac{1-s_n}{6} + s_n > \frac{1-s_n}{2}$ . Then, the space of a bin is greater than the size of an item of type  $n_2$ , this implies that two items of type  $n_2$  and one item of type  $p_5$  can be packed in the same bin. Hence, in the MFFD solution, two items of a smaller size of type  $n_2$  can be packed with an item of type  $p_5$ . Then,  $\sum_{i \in J} W(i) = 2(\frac{6}{14}) + \frac{1}{5} = 1 + \frac{2}{35}$ . Since each optimal bin has one item of a smaller size of  $n_2$ , we adjust half of the number of item type  $n_2$  with  $\frac{2}{35}$ . Hence, we subtract  $\frac{1}{2}(\frac{2}{35})$  from the maximum weight. After the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{499}{420} - \frac{1}{2}(\frac{2}{35}) = \frac{487}{420}$ .

Let  $\kappa_1$  be the number of the optimal bins with two items of type  $n_2$ , one item of type  $p_5$  and one more item,  $\kappa_2$  be the number of the optimal bins with two items of type  $n_2$ , one item of type  $n_5$  and one more item,  $\kappa_3$  be the number of the optimal bins with one item of type  $n_2$ , one of type  $p_5$ , one of type  $p_6$  and the rest items are of size  $\geq \frac{1-s_n}{5}$ . If  $\kappa_i > 0$ , for  $i = \{1, 2, 3\}$ , then the number of the MFFD bins with two items of type  $n_2$  and one item of type  $p_5$  is  $\frac{\kappa_1 + \kappa_3}{2}$ , and the number of the MFFD bins with two items of type  $n_2$  and one item of type  $n_5$  is  $\frac{\kappa_2}{2}$ . This does not affect the weight adjustment. In summary, after the adjustment,  $\sum_{i \in J^*} W(i) \leq \frac{71}{60}$  without an item of type  $G$ . By Lemma 3.2.4, we have

$$\begin{aligned}
z^H - 11 &\leq \sum_{b=1}^{z^*} \sum_{i \in J_b^*} W(i) - \sum_{i=z^*+1}^{z^H-1} W(i) - \sum_{b=1}^{z^H} (\sum_{i \in J_b} W(i) - 1). \\
&\leq \lambda \left( \frac{113}{84} - \frac{3}{4} \right) + \frac{71}{60} (2z^* - z^H + 1) + \left( \frac{257}{210} - \frac{6}{14} \right) (z^H - 1 - z^* - \lambda). \\
&\leq \frac{50}{84} \lambda + \frac{71}{60} (2z^* - z^H - \lambda) + \frac{167}{210} (z^H - 1 - z^*) + \frac{163}{420}. \\
&\leq \left( \frac{167}{210} - \frac{71}{60} \right) z^H + \left( \frac{142}{60} - \frac{167}{210} \right) z^* + \frac{163}{420}.
\end{aligned}$$



$$\frac{583}{420}z^H - 11 \leq \frac{660}{420}z^* + \frac{163}{420}.$$

Note that  $\frac{71}{60}(2z^* - z^H) + (\frac{257}{210} - \frac{6}{14})(z^H - z^*) \geq \frac{23}{20}(2z^* - z^H) + (\frac{23}{20} - \frac{1}{4})(z^H - z^*)$ . Hence,

$$\frac{z^H}{z^*} \leq \frac{660}{583} + \frac{4783}{420z^*} \text{ and } \lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{660}{583} < \frac{91}{80}.$$

**Case 2:**  $\lambda > z^H - 1 - z^*$

This case is similar to Case 2 of Section 3.2.8, except that the size of the largest item is at least  $\frac{1-s_n}{2}$  instead of  $\frac{1}{2}$ . Recall  $m = \lfloor \frac{z^H}{9} \rfloor$ , and set  $z^H = 9m + j$ , where  $j = \{0, 1, \dots, 8\}$ . Consider

the total size of bins except the largest items. By the FF rule, the item  $n$  with  $s_n < \frac{1}{7}$  cannot be packed in any bin  $b$  in which  $\sum_{i \in J_b} s_i > \frac{6}{7}$ , where  $b \in \{1, \dots, z^H - 1\}$ . Hence,

$$\begin{aligned} \sum_{i=8m+1}^n s_i &= \sum_{b=1}^{8m} (\sum_{i \in I_b} s_i - s_b) + \sum_{b=8m+1}^{z^H} \sum_{i \in I_b} s_i. \\ &> 8m(1 - s_n) + (z^H - 1 - 8m)(1 - s_n + \frac{1 - s_n}{2}). \\ &\geq 8m(1 - s_n) + (9m - 1 - 8m)(1 - s_n + \frac{1 - s_n}{2}). \\ &= (\frac{3}{2}m - \frac{3}{2} + 8m)(1 - s_n). \\ &\geq (\frac{19}{2}m - \frac{3}{2})(1 - \frac{1}{7}). \\ &\geq \frac{57m}{7} - \frac{9}{7} \geq 8m - 2. \end{aligned}$$

By Lemma 3.1.1(ii),  $z^* \geq 8m - 1$ . Thus,  $\lim_{z^* \rightarrow \infty} \frac{z^H}{z^*} \leq \frac{9m+j}{8m-1} = \frac{9}{8}$ . ■

By all lemmas, the proof of Theorem 3.2.1 is completed. Next, we conduct the numerical study in Chapter 4.

# Chapter 4

## Empirical Study of the FFD and MFFD Algorithms

In Chapter three, we analyze the worst-case performance of the FFD and MFFD algorithms and conclude that the MFFD outperforms the FFD. In this chapter, we study their average-case performance by empirical tests.

### 4.1 Empirical Tests of the FFD and MFFD Algorithms

We prove that OBPP is NP-hard and hence it faces computational difficulty to obtain optimality. Fortunately, we can estimate a lower bound of the number of optimal bins by using  $k_1$  from Step 1 of the MFFD algorithm according to Property 3.2.1(i) (page 26).

In order to numerically compare this lower bound with the solutions generated by the FFD and MFFD algorithms, we conduct the following experiments. First, we generate the item

sizes according to the uniform (0,1) distribution for 100 items. Table 4.1 compares the lower bound with the solutions from the FFD and MFFD algorithms. We then repeat the same experiment for 500, 1000, and 2000 items, and present the results in Tables 4.2, 4.3, and 4.4, respectively.

The programs are coded by C++ and run on a pentium IV 2.8 GHz machine, which completes the computations almost instantaneously. This experiment demonstrates that the MFFD algorithm is asymptotically optimal for the uniformly distributed data set. Table 4.5 shows that the MFFD algorithm outperforms the FFD algorithm in worst-case and average-case performance.

Table 4.1: The average-case performance of the FFD and MFFD algorithms when the number of items is 100.

<b>Replication</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>
<b>Lower bound (<math>k_1</math>)</b>	28	28	25	28	28	26	28	27	27	28
<b>FFD</b>	38	37	34	38	37	36	38	35	37	37
<b>MFFD</b>	28	29	25	30	29	27	29	27	28	29

Table 4.2: The average-case performance of the FFD and MFFD algorithms when the number of items is 500.

<b>Replication</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>
<b>Lower bound (<math>k_1</math>)</b>	136	134	132	134	135	130	140	141	136	130
<b>FFD</b>	181	179	175	178	179	174	183	185	181	175
<b>MFFD</b>	137	135	133	135	136	131	141	142	136	131

Table 4.6 and Table 4.7 summarize the results for the average-case and worst-case scenario of the experiment under a uniformly distributed data and ten replications. Numerical results

Table 4.3: The average-case performance of the FFD and MFFD algorithms when the number of items is 1000.

Replication	1	2	3	4	5	6	7	8	9	10
<b>Lower bound</b> ( $k_1$ )	266	271	264	263	269	266	260	276	270	265
<b>FFD</b>	355	360	350	349	356	353	348	366	359	352
<b>MFFD</b>	266	272	264	264	269	267	260	277	271	265

Table 4.4: The average-case performance of the FFD and MFFD algorithms when the number of items is 2000.

Replication	1	2	3	4	5	6	7	8	9	10
<b>Lower bound</b> ( $k_1$ )	525	536	533	524	523	525	537	530	533	541
<b>FFD</b>	699	710	705	694	695	698	714	704	704	715
<b>MFFD</b>	525	537	534	525	524	526	538	530	533	541

demonstrate that the MFFD algorithm outperforms the FFD algorithm in both worst-case and average-case scenarios. Both algorithms work very well on  $(0,1/8)$  uniformly distributed data. However, for a uniform  $(1/3,1)$  distribution according to Lemma 3.2.5,  $k_1$  is not a tight lower bound for the number of optimal bins. Thus, we adjust the lower bound by using  $z^{MFFD}$  instead of  $k_1$ . The last columns of Tables 4.6 and 4.7 show that with this improved lower bound, the ratios of the worst-case and average-case performance of the FFD and MFFD algorithms dropped by 7% – 11% and 7% – 12%, respectively. Figure 4.1 summarizes the average-case performance of the FFD and MFFD algorithms.

Next, we perform the test of uniform  $(0,1)$  and mixed-uniform  $(0,1)$  distributions for 100 and 1000 items, using fifty replications. We choose the  $(0,1)$  mixed-uniform distributed data to study the performance of the  $(0,1)$  normally distributed data instead of using truncated

Table 4.5: Summary of the average-case performance of the FFD and MFFD algorithms.

The number of items	100	500	1000	2000
<b>FFD</b>	1.3447	1.3281	1.3288	1.3262
<b>MFFD</b>	1.0289	1.0067	1.0019	1.0011

Table 4.6: Summary of the average-case performance of the FFD and MFFD algorithms.

The number of items	The average-case of	U(0,1)	U(1/3,1)	U(0,1/8)	U(1/4,1)	U(1/8,1)	$U'(1/3,1)$
100	FFD	1.34	1.31	1.08	1.34	1.34	1.20
	MFFD	1.03	1.10	1.00	1.06	1.04	1.00
500	FFD	1.33	1.30	1.08	1.31	1.32	1.21
	MFFD	1.01	1.07	1.00	1.02	1.01	1.00
1000	FFD	1.33	1.30	1.08	1.30	1.32	1.21
	MFFD	1.00	1.07	1.00	1.02	1.01	1.00
2000	FFD	1.33	1.29	1.07	1.31	1.32	1.22
	MFFD	1.00	1.07	1.00	1.01	1.01	1.00

Table 4.7: Summary of the worst-case performance of the FFD and MFFD algorithms.

The number of items	The worst-case of	U(0,1)	U(1/3,1)	U(0,1/8)	U(1/4,1)	U(1/8,1)	$U'(1/3,1)$
100	FFD	1.38	1.35	1.09	1.50	1.38	1.24
	MFFD	1.07	1.12	1.02	1.09	1.07	1.00
500	FFD	1.35	1.31	1.08	1.33	1.34	1.22
	MFFD	1.01	1.09	1.00	1.04	1.01	1.00
1000	FFD	1.34	1.30	1.08	1.31	1.33	1.23
	MFFD	1.00	1.08	1.00	1.03	1.01	1.00
2000	FFD	1.33	1.30	1.08	1.31	1.34	1.23
	MFFD	1.00	1.07	1.00	1.03	1.01	1.00

normally distributed data. Hence, we generate uniform random variables of ten intervals, which belong to  $(0, 0.1)$ ,  $[0.1, 0.2)$ , ...,  $[0.9, 1)$ , having 1, 3, 5, 10, 31, 31, 10, 5, 3 and 1 items per interval, respectively. These random variables are called mixed-uniform  $(0, 1)$  distributions

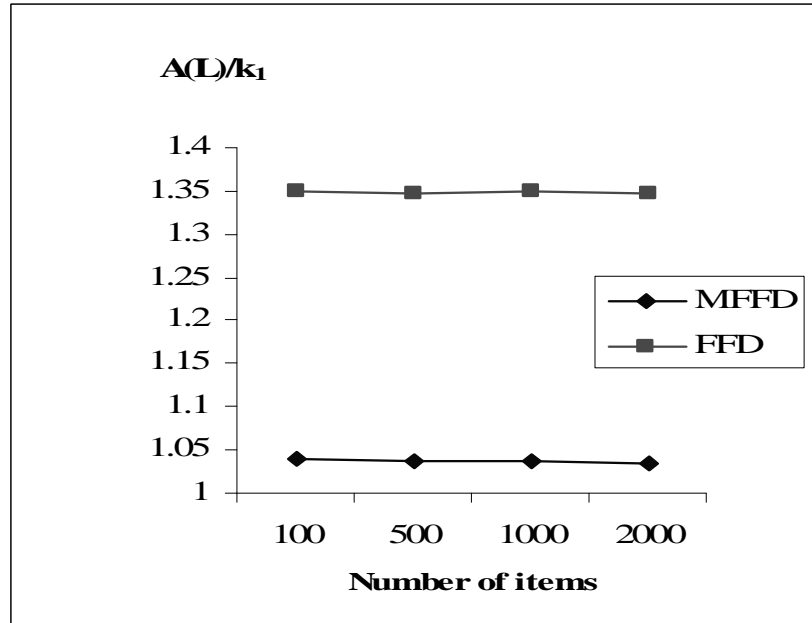


Figure 4.1: Summary of the average-case performance of the FFD and MFFD algorithms.

since they are uniformly distributed in each interval but normally distributed over  $(0, 1)$ .

The parenthetical numbers in Tables 4.8 and 4.9 are  $k_1$ . We found that for the uniformly distributed data, the MFFD generates at most one more bin than the lower bound. Thus, the MFFD is asymptotically optimal. Moreover, both algorithms performs slightly better with a uniformly distributed data. This implies that the MFFD and FFD algorithms perform worse when the data deviate from uniform distribution.

Table 4.8: Summary of the worst-case and average-case performance of the FFD and MFFD algorithms of a uniformly distributed data set.

The number of items	FFD	MFFD	The number of items	FFD	MFFD
100	34(24)	25(24)	1000	355(264)	260(259)
$R_A^\infty$	1.417	1.0417	$R_A^\infty$	1.3447	1.0038
$\bar{R}_A$	1.3467	1.0253	$\bar{R}_A$	1.3266	1.0019

Table 4.9: Summary of the worst-case and average-case performance of the FFD and MFFD algorithms of a mixed-uniformly distributed data set

The number of items	FFD	MFFD	The number of items	FFD	MFFD
<b>100</b>	41(30)	31(30)	<b>1000</b>	401(298)	303(298)
$R_A^\infty$	1.3667	1.0333	$R_A^\infty$	1.3456	1.0168
$\bar{R}_A$	1.3500	1.0333	$\bar{R}_A$	1.3388	1.0134

Finally, we study the worst-case performance of both algorithms by running the algorithm on problems that the optimal number of bins is known. Let  $N = 10$  and  $\delta$  be a very small positive number. Consider eight problems as follows:

1. An example of worst-case performance of the FFD as in Figure 3.1.
2. An example of worst-case performance of the MFFD as in Figure 3.3.
3.  $101N$  items of size  $1/2 + 2\delta$ ,  $34N$  items of size  $1/4 + 4\delta$ ,  $42N$  items of size  $1/4 + 2\delta$ , and  $76N$  items of size  $1/4 - 5\delta$ .
4.  $10N$  items of size  $1 - \delta$  and  $1/3 - \delta$  and  $40N$  items of size  $1/6$ .
5.  $49N$  items of size  $1 - \delta$ ,  $27N$  items of size  $1/2 + \delta$  and  $1/3 - 2\delta$ ,  $22N$  items of size  $1/3 - \delta$ , and  $115N$  items of size  $1/6$ .
6.  $9N$  items of size  $1 - \delta$  and  $5/12 - \delta$ ,  $1/3$ , and  $1/4$ .
7.  $8N$  items of size  $1 - \delta$ ,  $16N$  items of size  $3/10 - \delta$  and  $1/5$ .
8.  $42N$  items of size  $1 - \delta$ ,  $23N$  items of size  $2/7$ ,  $42N$  items of size  $2/7 - \delta$ , and  $336N$  items of size  $1/7$ .

9.  $35N$  items of size  $1 - \delta$  and  $1/4 - 7\delta$ , and  $210N$  items of size  $1/8 + \delta$ .

Table 4.10: Summary of the worst-case of FFD and MFFD algorithms.

<b>Problem</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>
<b>Lower bound</b> ( $k_1$ )	100	80	590	100	490	90	80	420	350
$z^{FFD}$	150	116	847	142	694	128	112	607	508
$z^{MFFD}$	100	91	670	110	540	100	90	460	380
$R_{FFD}^\infty$	1.500*	1.45	1.4356	1.420	1.4163	1.4222	1.400	1.4452	1.4514
$R_{MFFD}^\infty$	1.000	1.1375*	1.1356	1.100	1.1020	1.1111	1.1250	1.0952	1.0857

\* indicate tight asymptotic worst-case ratio.

Table 4.10 shows that the worst-case performance of the FFD and MFFD algorithms can be reached (or the worst-case ratio is tight) as shown in problems 1 and 2. This shows that the MFFD performs very well in general and has only 13.75% more bins than the optimal bins in the worst case scenario. However, for asymptotic number of items as shown in Table 4.5, the average-case performance of the FFD and MFFD algorithms are no more than 33% and 1% of that of the optimal solutions, respectively.

## 4.2 Conclusions and Future Research

The Open Bin Packing Problem (OBPP) is a new variant of a well-known Bin Packing Problem (BPP). In the OBPP, items are packed into bins so that the total content before the last item in each bin is strictly less than the bin capacity. The objective is to minimize the number of bins used. The OBPP is proved to be NP-hard. Instead of solving OBPP to optimality, a heuristic that provides acceptable solutions in a short time is of interest.



We propose two offline algorithms, in which the sizes of items are known in advance. First, we consider the First Fit Decreasing algorithm (FFD), which is a well-known algorithm for the BPP. We prove that its asymptotic worst-case performance ratio is no more than  $3/2$ . We notice that the FFD performance of the OBPP is worse than that of the BPP. Alternatively, we modify the FFD algorithm by adding an algorithm in which the set of the largest items is the set of the last items packed in each bin. Then, we propose the Modified First Fit Decreasing (MFFD) and prove that its asymptotic worst-case performance ratio is no more than  $91/80 \simeq 1.1375$ . Finally, we conduct empirical tests of the FFD and MFFD average-case performance. The results show that in general, the FFD and MFFD use no more than 33% and 1% of the number of bins than that of optimal packing, respectively. In addition, the MFFD is asymptotically optimal when the sizes of items are  $(0, 1)$  uniformly distributed.

Further research can be as follows: the unbounded distribution or  $(0, 1]$  uniform distribution of the sizes of items, the semi-online and online algorithms that are applicable when information about items is not available. The FFD can be modified as a semi-online algorithm by allowing some packed items to be repacked. Then, the lack of information of item sizes causes its asymptotic worst-case performance to be worse than that of the MFFD. In addition, the extension of the MFFD algorithm for the unequal capacity open bin packing problem is of interest.

# Bibliography

- [1] S. Anily and J. Bramel. Quantitative models for supply chain management. In S. Tayur, R. Ganeshan, and M. Magazine, editors, *Vehicle Routing and the Supply Chain*, pages 147–196. Kluwer Academic Publishers, Massachusetts, 1999.
- [2] S. Anily, J. Bramel, and D. Simchi-Levi. Worst-case analysis of heuristics for the bin packing problem with general cost structures. *Operations Research*, 42:287–298, 1994.
- [3] B. Baker. A new proof for the first-fit decreasing bin-packing algorithm. *Journal of Algorithms*, 6:49–70, 1983.
- [4] B. Baker and Jr. E. Coffman. A tight asymptotic bound for next-fit-decreasing bin-packing. *SIAM Journal on Algebraic and Discrete Methods*, 2:147–152, 1981.
- [5] J. Berkey and P. Wang. Two-dimensional finite bin-packing algorithms. *Journal of Operation Research Society*, 38:423–429, 1987.
- [6] D. Bienstock, J. Bramel, and D. Simchi-Levi. A probabilistic analysis of tour partitioning heuristics for the capacitated vehicle routing problem with unsplit demands. *Math O. R.*, 18:786–802, 1993.

- [7] J. Bramel, Jr. E., P. Shor, and D. Simchi-Levi. Probabilistic analysis of the capacitated vehicle routing problem with unsplit demands. *Operations Research*, 40:1095–1106, 1992.
- [8] J. Bramel and D. Simchi-Levi. *The Logic of Logistics: Theory, Algorithms, and Applications for Logistics Management*. Springer, 2001.
- [9] L. M. A. Chan, A. Federgruen, and D. Simchi-Levi. Probabilistic analyses and practical algorithms for inventory-routing models. *Operations Research*, 46:96–106, 1998.
- [10] L. M. A. Chan, D. Simchi-Levi, and J. Bramel. Worst-case analyses, linear programming and the bin-packing problem. *Mathematical Programming*, 83:213–227, 1998.
- [11] C. Chu and R. La. Variable-sized bin packing: Tight absolute worst-case performance ratios for four approximation algorithms. *SIAM Journal on Computing*, 30:2069–2083, 2001.
- [12] Jr. E. Coffman, M. Garey, and D. Johnson. In D. Hochbaum, editor, *Approximation Algorithms for NP-Hard Problems*, pages 46–93. PWS Publishing Company, Massachusetts.
- [13] Jr. E. Coffman, M. Garey, and D. Johnson. An application of bin-packing to multiprocessor scheduling. *SIAM Journal on Computing*, 7:1–17, 1978.
- [14] Jr. E. Coffman, D. Johnson, P. Shor, and R. Weber. Bin-packing with discrete item sizes, part ii: Tight bounds on first fit. *Random Structures and Algorithms*, 10:69–101, 1997.
- [15] Jr. E. Coffman, J. Lenstra, and A. Rinnooy Kan. Computational complexity. In *Handbooks in Operation Research and Management Science*, pages 121–154. North-Holland, 1992.

- [16] Jr. E. Coffman and G. Lueker. Bin packing heuristics. In *Probabilistic Analysis of Packing and Partitioning Algorithms*, pages 121–154. Wiley & Sons, 1991.
- [17] Jr. E. Coffman, G. Lueker, and A. Rinnooy Kan. Asymptotic methods in the probabilistic analysis of sequencing and packing heuristics. *Management Science*, 34:266–290, 1988.
- [18] J. Csirik. An on-line algorithm for variable-sized bin packing. *Acta Inform*, 26:697–709, 1989.
- [19] Jnos Csirik and David S. Johnson. Bounded space on-line bin packing: Best is better than first. *Algorithmica*, 31:115–13, 2001.
- [20] K. Dowsland. The three-dimensional pallet chart:an analysis of the factors affecting the set of feasible layouts for a class of two-dimensional packing problems. *Journal of Operation Research Society*, 35:895–905, 1984.
- [21] E. Falkenauer. A hybrid grouping genetic algorithm for bin packing. *Journal of Heuristics*, 2:5–30, 1996.
- [22] A. Federgruen and G. Ryzin. Probabilistic analysis of a generalized bin packing problem with applications to vehicle routing and scheduling problems. *Operations Research*, 45:596–609, 1997.
- [23] D. Friesen and M. Langston. Variable-sized bin packing. *SIAM Journal on Computing*, 15:222–230, 1986.
- [24] D. Friesen and M. Langston. Analysis of a compound bin packing algorithm. *SIAM Journal Discrete Mathematics*, 4:61–79, 1991.

- [25] G. Gambosi, A. Postiglione, and M. Talamo. New algorithm for online bin packing problem a survey in r. petreschi et al. editor. *Algorithms and complexity : proceedings of the first Italian conference, Rome, Italy, 1-2 October 1990*, 1:44–59, 1990.
- [26] Graham R. Johnson D. Garey, M. and A. Yao. Resource constrained scheduling as generalized bin packing. *Journal of Combinatorial Theory Ser. A*, 21:257–298, 1976.
- [27] M. Garey and D. Johnson. *Computers And Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York, 1979.
- [28] I. Gent. Heuristic solution of open bin packing problems. *Journal of Heuristics*, 3:299–304, 1998.
- [29] M. Hofri. A probabilistic analysis of the next-fit bin packing algorithm. *Journal of Algorithms*, 5:547–556, 1984.
- [30] Z. Ivkovic and E. Lloyd. Fully dynamic algorithms for bin packing: Being (mostly) myopic helps. *SIAM Journal on Computing*, 28:574–611, 1998.
- [31] D. Johnson, A. Demers, J. Ullman, M.Garey, and R. Graham. Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM Journal on Computing*, 3:299–325, 1974.
- [32] David Johnson. *Near-Optimal Bin Packing Algorithms*. PhD thesis, MIT, Cambridge, Massachusetts, June 1973.

- [33] N. Karmarkar, editor. *Probabilistic Analysis of Some Bin Packing Algorithms*, Proceeding of the 23<sup>rd</sup> Annual Symposium on Foundations of Computer Science, Long Beach, CA, 1982. IEEE Computer Society.
- [34] N. Karmarkar and R. Karp, editors. *An Efficient Approximation Scheme for the one-dimensional bin packing problem*, Proceeding of the 23<sup>rd</sup> Annual Symposium on Foundations of Computer Science, Long Beach, CA, 1982. IEEE Computer Society.
- [35] R. Karp, M. Luby, and A. Marchetti-Spaccamela, editors. *A Probabilistic Analysis of Multidimensional Bin Packing Problems*, Proceeding of the 16<sup>th</sup> Annual ACM Symposium on Theory of Computing, 1984.
- [36] C. Lee and D. Lee. A simple on-line bin packing algorithm. *Journal of ACM*, 3:562–572, 1985.
- [37] J. Leung, D. Moshe, and G. Young. Technical note: A note on an open-end bin packing problem. *Journal of Scheduling*, 4:201–207, 2001.
- [38] F. Liang. A lower bound for on-line bin packing. *Information Processing Letter*, 10:76–79, 1980.
- [39] Andrew Lim and Wang Ying. A new method for the three dimensional container packing problem. In *IJCAI*, pages 342–350, 2001.
- [40] R. Loulou. Probabilistic behavior of optimal bin packing. *Operation Research Letter*, 3:129–135, 1984.

- [41] W. Mao. Best-k-fit bin packing. *Computing*, 50:265–270, 1993.
- [42] W. Mao. Tight worst-case performance bounds for next-k-fit bin packing. *SIAM Journal on Computing*, 22:46–56, 1993.
- [43] S. Martello, D. Pisinger, and D. Vigo. The three-dimensional bin packing problem. *Operations Research*, 48:256–267, 2000.
- [44] S. Martello and P. Toth. Lower bounds and reduction procedure for the bin packing problem. *Discrete Applied Mathematics*, 22:59–70, 1990.
- [45] F. K. Miyazawa and Y. Wakabayashi. An algorithm for the three-dimensional packing with asymptotic performance analysis. *Algorithmica*, 18:122–144, 1997.
- [46] F. K. Miyazawa and Y. Wakabayashi. Approximation algorithm for the orthogonal  $z$ -oriented 3-d packing problem. *SIAM Journal on Computing*, 29:1008–1029, 2000.
- [47] H. Ong, M. Magazine, and T. Wee. Probabilistic analysis of bin packing heuristics. *Operations Research*, 32:983–998.
- [48] Gunther R. Raidl. A weight-coded genetic algorithm for the multiple container packing problem. In *Selected Areas in Cryptography*, pages 291–296, 1999.
- [49] W. Rhee. Martingale inequalities and NP-complete problems. *Math. Operation Research*, 12:177–181, 1987.
- [50] W. Rhee. Optimal bin packing with items of random sizes. *Math. Operation Research*, 13:140–151, 1988.

- [51] M. Richey. Improved bounds for harmonic-based bin packing problem algorithms. *Discrete Applied Mathematics*, 34:203–227, 1991.
- [52] S. Seiden. On the online bin packing problem. *Journal of the ACM (JACM)*, 49:640–671, 2002.
- [53] S. Seiden, R. Stee, and L. Epstein. New bounds for variable-sized online bin packing. *SIAM Journal on Computing*, 32:455–469, 2003.
- [54] D. Simchi-Levi. New worst-case results for the bin-packing problem. *Naval Research Logistics*, 41:579–585, 1994.
- [55] P. Sweeney and E. Paternoster. Cutting and packing problems: A categorized application-orientated research bibliography. *Journal of Operation Research Society*, 43:691–706, 1992.
- [56] A. Vliet. An improved lower bound for on-line bin packing algorithms. *Information Processing Letters*, 43:277–284, 1992.
- [57] A. Vliet. On the asymptotic worst-case behavior of harmonic fit. *Journal of Algorithms*, 20:113–136, 1996.
- [58] J. Yang and J. Leung. The ordered open-end bin packing problem. *Operations Research*, 51:759–770, 2003.
- [59] A. Yao. New algorithm for bin-packing. *Journal of ACM*, 27:207–227, 1980.



- [60] M. Yue. A simple proof of the inequality  $\text{FFD}(L) \leq \frac{11}{9} \text{OPT}(L) + 1 \quad \forall L$  for the bin-packing algorithm. *ACTA Mathematicae Applicatae Sinica*, 7:321–331, 1991.
- [61] G. Zhang. Parameterized on-line open-end bin packing. *Computing*, 60:267–273, 1998.

# Vita

Pornthipa Ongkunaruk (Morr) was born in Bangkok, the capital of Thailand, in 1975. She received her B.S. degree in the department of Agro-Industrial Product Development, the faculty of Agro-Industry, Kasetsart University (KU), Thailand in 1997. She was awarded the first class honor in 1997. Later she received a scholarship from the Royal Thai Government for her M.S. degree study in Industrial and Systems Engineering at Asian Institute of Technology (AIT), Thailand. she was awarded the James A. Linen III Memorial Prize in recognition of the most outstanding academic and graduated in 1999.