IEMS Vol. 7, No. 2, pp. 126-132, September 2008.

The Asymptotic Worst-Case Ratio of the Bin Packing Problem by Maximum Occupied Space Technique

Pornthipa Ongkunaruk†
Program of Agro-Industry Technology Management
Faculty of Agro-Industry, Kasetsart University, Bangkok 10900 THAILAND
+662-562-5000 Ext. 5363, E-mail: fagipoo@ku.ac.th

Selected paper from APIEM 2006

Abstract. The bin packing problem (BPP) is an NP-Complete Problem. The problem can be described as there are N = {1, 2, ⋯, n } which is a set of item indices and L = {s1, s2, ⋯, sn} be a set of item sizes sj, where 0 <sj ≤ 1,∀ j ∈ 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. In the past, many researchers put on effort to find the heuristic algorithms instead of solving the problem to optimality. Then, the quality of solution may be measured by the asymptotic worst-case ratio or the average-case ratio. The First Fit Decreasing (FFD) is one of the algorithms that its asymptotic worst-case ratio equals to 11/9. Many researchers prove the asymptotic worst-case ratio by using the weighting function and the proof is in a lengthy format. In this study, we found an easier way to prove that the asymptotic worst-case ratio of the First Fit Decreasing (FFD) is not more than 11/9. The proof comes from two ideas which are the occupied space in a bin is more than the size of the item and the occupied space in the optimal solution is less than occupied space in the FFD solution. The occupied space is later called the weighting function. The objective is to determine the maximum occupied space of the heuristics by using integer programming. The maximum value is the key to the asymptotic worst-case ratio.

Keywords: The Bin Packing Problem, Heuristic, Asymptotic Worst-case Ratio, Maximum Occupied Space

1. INTRODUCTION

In a Bin Packing Problem (BPP), let N = {1, 2, ⋯, n} be a set of item indices and L = {s_1, s_2, ⋯, s_n} be a set of item sizes s_j, where $0 < s_j \leq 1$; ∀ j ∈ 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 (1979) and Coffman *et al*. (1997), a BPP is an NP-Complete Problem. We review some traditional heuristics for the BPP. Each algorithm indexes bins {1, 2, ⋯, n} in the order they are opened.

2. LITERATURE REVIEW

The research focuses on the unit-capacity bin packing problem and the First Fit Decreasing (FFD) algorithm. First, Johnson *et al*. (1974), and Johnson (1973) show that the asymptotic worst-case performance ratio for First Fit (FF) and Best Fit (BF) is not more than 17/10, and not more than 11/9 for either FFD or Best Fit Decreasing (BFD). They prove that $FFD(L) \leq 11/9$ OPT (L) + 4, ∀ L, where L is a list of items in the BPP, $OPT(L)$ 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.

Later, Baker (1983) proves the theorem that FFD(L) ≤ 11/9 OPT (L) + 3, ∀ 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 (1991) 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.

† : Corresponding Author

They propose a compound algorithm, which combines both FFD and B2F algorithms in regions in which they are superior. Then, they prove by using a weighting function whose 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 (1991) proposes a simpler proof and provides a tighter bound with FFD(L) ≤ 11/9 OPT(L) + 1, ∀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. THEOREM AND PROOF

Theorem 1 $\lim_{z^*(L)\to\infty} \frac{z^H(L)}{z^*(L)} \le \frac{11}{9}$, *and the ratio is tight.*

Theorem 1 implies that the total number of bins generated by the FFD is not more than 22.22 % more bins than the number of optimal bins for BPP.

3.1 Preliminary

In this section, we present properties and lemmas to support the proof of Theorem 1. We will prove Theorem 1 by contradiction. Suppose there exists a list that violates the theorem. Let *L* be a minimum list that violates the theorem. That is $z^H(L) > 11/9\ z^*(L)$, but $z^H(L') \le 11/9\ z^*(L')$, for $L' \subset L$. Define $z^*(L)$ be the minimum number of bins used to pack items in a list L and $z^H(L)$ be the number of bins used by FFD algorithm to pack items in a list L. Define J_b be a set of items in any FFD bin b = 1, ⋯, $z^H(L)$ and J^*_b be a set of items in any optimal bin b = 1, ⋯, $z^*(L)$. However, we will call J and J* for the short notation of any FFD bin and optimal bin, respectively. After sorting items in a list L, let s_i be the size of the item i^{th} and s_n be the size of the last item. Then, we propose the following lemmas to support the proof.

Lemma 1: *For all b = 1, 2, ⋯, $z^H(L)$-1, we have that $|J_b| \ge 2$.*

Proof: Consider a list L using exactly d+1 bins in FFD. Suppose $J_b = \{j\}$ for some b = 1, 2, ⋯, d. Then, $s_j + s_n > 1$. Thus, $J^*_e = \{j\}$ for some e = 1, 2, ⋯, z^*. First, remove item j from a list L and call the new list L′. Then, $z^*(L') = z^*(L) - 1 \le 9/11\ z^H(L) - 1 \le 9/11\ (z^H(L')+1) - 1 \le 9/11\ z^H(L') - 2/11$. Hence, contradict with L, in which it is a minimum list that violates Theorem 1. □

Lemma 2: *For all b = 1, 2, ⋯, $z^*(L)$-1, we have that $|J^*_b| \ge 3$.*

Proof: The proof is done by Yue (1991). □

Next, we partition the range of the size of the last item (s_n) into five ranges as follows:

1. If $s_n > 1/3$, then $z^H(L) = z^*(L)$. Since $s_n > 1/3$, implies $|J^*_b| \le 2$. Combine with Lemma 1, it implies $z^H(L) = z^*(L)$. □
2. If $s_n < 2/11$, then $z^H(L) \le 11/9\ z^*(L)$. Since the item of size less than 2/11 cannot be packed in any previous bin before the last bin. This implies that the total content of any bin except the last bin must be less than 9/11. Then, the sum of the item sizes is at least $9/11(z^H(L)-1)$. Then, $z^*(L) \ge 9/11(z^H(L)-1)$ or $z^H(L) \le 11/9\ z^*(L)+1$. □

   For $2/11 \le s_n \le 1/3$, we partition into three ranges and we will proof this by using weighting function.
3. If $1/6 < s_n \le 1/5$, then $z^H(L) \le 73/60\ z^*(L)$.
4. 5If $1/5 < s_n \le 1/4$, then $z^H(L) \le 11/9\ z^*(L)$.
5. If $1/4 < s_n \le 1/3$, then $z^H(L) \le 7/6\ z^*(L)$.

3.2 The Maximum Occupied Space Technique

In this section, we present the maximum occupied space technique, using similar idea as the weighting function. In the literature, Johnson (1973), Johnson et al. (1974), Baker (1983), Frisen and Langston (1991) and Yue (1991) 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.

In the FFD packing, the actual size of the item does not indicate the space required for the packing. On the other hand, the weight of an item represents the space occupied by the item to pack in a bin. For example, if s_n = 1/6 and one of the bins has three items each of size 2/7 with the total size of 6/7, then we assign the weight of these items to 1/3. It shows that each item occupies the equal space of 1/3 instead of 2/7. Notice that the weight of an item is not smaller than its size.

Let *W(i)* be the space occupied by an item i, $W(J_b)$ and $W(J^*_b)$ be the space occupied by the FFD and optimal bin b, respectively. Similarly, define *W(L)* to be the total space occupied by items in a list L.

By the FFD rule, we define the types of the FFD bins as follows:

1. A Pure Bin. It is a bin containing 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 so that FFD packs i items of p_i into a p_i-bin.
2. A Non-Pure Bin. It is a bin containing items of different ranges. There are two types of non-pure bins defined as follows:
   2.1 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 which is an item of size $\ge 1/2$ and smaller items, and an n_i bin, for $i \ge 2$ containing i items of n_i and one or more fallback items.

2.2 A Transition Bin. It is a bin containing the highest indexed item of any of item types except type n_1, i.e. it is the last FFD bin of each item type except G bin. The total number of transition bins is not more than the total number of the p_i and n_i types of bins.

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. We call the occupied space for bin packing as the weight and assign the weight of the FFD bin so that the total weight in each bin is equal to one. To avoid the numerous values of the weight of different items, we unify the weight of the item type. For $1/(m+1) < s_n \le 1/m$, where m = 3, 4, 5, the weight of the same item type is unique and follows the following rules:

1. If item j is of type p_i, then $W(j) = 1/i$.
2. If item j is of type n_1 or G, then $W(j) = 1-1/m$.
3. If item j is of type n_i, for $i > 1$, then $W(j) = (m-2)/(i^*(m-1))$.

Next, we define the ranges of item sizes for $1/(m+1) < s_n \le 1/m$ as follows:

1. To determine the maximum size of an item type n_1 or G, by Lemma 2, we have $|J^*_b| \ge 3$. Thus, $s_G < 1 - 2^* s_n$.
2. For a p_i item, $1-s_n \le i^* s_{pi} < 1$, $\forall\ i < m$. Thus, $(1-s_n)/i \le s_{pi} < 1/i$.
3. For an n_i item $s_{pi+1} \le s_{ni} < s_{pi}$, $\forall\ i = \{2, \cdots, m-1\}$.
4. The last range is $s_n \le s_{pi} < s_{ni-1}$, for i = m.

Next, we define $z^H = z^H(L)$ and $z^* = z^*(L)$ and use these notation for the rest of the paper. Then, we derive the following lemma to proof Theorem 1:

Lemma 3 $\lim_{z^* \to \infty} \frac{z^H}{z^*} \le W^M$, *where W^M is the maximum occupied space of optimal bins.*

Proof:

$$\sum_{b=1}^{z^H} W(J_b) = \sum_{b=1}^{z^*} W(J^*_b) = W(L) \quad (1)$$

$$z^H - A = \sum_{b=1}^{z^*} W(J^*_b) \quad \le W^M z^* \quad (2)$$

$$z^H \le W^M z^* + A \quad (3)$$

Where A is a positive number. We need to show that W^M is equal to 11/9. □

3.3 The Integer Program for the Maximum Space Occupied by the Optimal Bins

In this section, we adapt an integer program for the Open Bin Packing Problem by Ongkunaruk (2005) to show how to determine the maximum value of $W(J^*_b)$ for the BPP. The objective is to determine the maximum occupied space of the optimal packing so that the total size of the items in a bin is not more than one. For $1/(m+1) < s_n \le 1/m$, we can formulate the problem as follows:

Problem 1: $\max \sum_i w(i)x_i$

$$s.t. \sum_i LS_i(m)x_i \le 1. \quad (4)$$

$$x_i \ge 0, \ \text{integer } \forall i \in \{n_j, p_{j+1}\}\ \forall j = \{1, \cdots, m-1\}. \quad (5)$$

Where x_i is the number of item type i in J^* (See Ex.1), $LS_i(m) \in LS(m)$, where LS(m) is a set of the smallest size of item types, which is a function of s_n, and s_n is a function of m. Furthermore, $W(i)$ is the occupied space function of an item type i. Constraint 4 requires that the total content of items in J^* must be ≤ 1. Constraint 5 requires that the number of items of type i in J^* must be a positive integer. The objective is to determine the maximum occupied space in J^*. We need to modify the left hand side of Constraint 4 since the value of $LS_i(m)$ depends on $s_n \in (1/(m+1), 1/m]$. Let $LS_i(m) = a_i + b_i s_n$. Then, consider the following two cases:

1. If $b \ge 0$, then we use the lower bound of s_n since

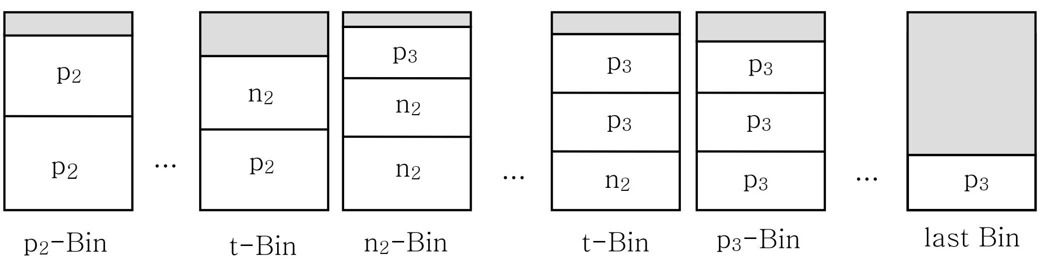


Figure 1. An example of the FFD packing that contains a p_2 bin, a transition bin, a n_2 bin, a p_3 bin and the last bin, respectively.

$$\sum_i (a_i + b_i(\frac{1}{m+1} + \frac{1}{2\delta})x_i) \le \sum_i (a_i + b_i s_n x_i) \le 1 \quad (6)$$

Where δ is the least common factor of the denominators of LS(m), for $s_n = 1/(m+1)$.

2. If b < 0, then we use the upper bound of s_n since

$$\sum_i (a_i + b_i(\frac{1}{m})x_i) \le \sum_i (a_i + b_i s_n x_i) \le 1 \quad (7)$$

If the lower bound of the size is not an exact value, we add the small value by using the least common factor of the denominators of LS(m). From these two cases, we can formulate an integer program by using $s_n = 1/m$ and $s_n = 1/(m+1) + 1/2\delta$. Then, we select the maximum objective function or $W(J^*_b)$ between two integer programs as shown in Example 1.

Example 1: An integer program for m = 3 or $1/4 < s_n \le 1/3$. The occupied space and sizes of items are in Table 1. First, if $s_n = 1/3$, then LS(3) = {(1-1/3)/2+1/2ε, 1/3+1/4ε, 1/3} = {5/12, 3/8, 1/3}, where ε = 6. The integer program is as follows:

$$\max \quad \frac{1}{2}x_{p_2} + \frac{1}{3}x_{n_2}\,\frac{1}{3}x_{p_3}$$

$$s.t. \quad \frac{5}{12}x_{p_2} + \frac{3}{8}x_{n_2}\,\frac{1}{3}x_{p_3} \le 1$$

$$\text{or} \quad 10x_{p_2} + 9x_{n_2}\,8x_{p_3} \le 24 \quad (8)$$

$$x_{p_2}, x_{n_2}, x_{p_3} \ge 0, \text{ integer} \quad (9)$$

Next, if $s_n = 1/4+1/2\delta$, where δ = 2*3*4 = 24. Then, LS(3) = {(1-13/48)/2+1/2ε, 1/3+1/4ε, 13/48}, where ε =24. Thus, LS(3) = {37/96, 11/32, 13/48}, then Constraint 8 changes to

$$\frac{37}{96}x_{p_2} + \frac{11}{32}x_{n_2}\,\frac{13}{48}x_{p_3} \le 1 \quad (10)$$

After solving two problems, the optimal solution is $J^* = (p_2, n_2, p_3)$ and $W(J^*_b) = 7/6$. The integer programs for m = 4 and m = 5 are shown in Appendix.

Table 1. Summary of item sizes and weight for $1/4 < s_n \le 1/3$.

Items Types	Range of Item Sizes	Occupied Space
p_2	$((1-s_n)/2, 1/2]$	1/2
n_2	$(1/3, (1-s_n)/2)$	1/3
p_3	$(s_n, 1/3]$	1/3

3.4 The Proof of Theorem 1

In this section, we use Lemma 3 for the proof of $1/6 \le s_n \le 1/3$. We partition the range into three ranges. In each range, we derive the maximum occupied space as follows:

3.4.1 If $1/4 < s_n \le 1/3$, then $z^H(L) \le 7/6\, z^*(L)$

In this case, we have that $|J^*b| \le 3$. Combine this with Lemma 2, we have that $|J^*b| = 3$. Since $s_G+2s_n > 1/2+2*1/4 = 1$. Thus, G item is not in J^*. Table 1 shows item sizes and weight for $1/4 < s_n \le 1/3$.

According to the FFD rule (see Figure 1), two items of item type p_2, two items of item type n_2 with one item of type p_3, and three items of type p_3; except the transition bins with the total weight ≤ 3. Each of Other FFD bins has total occupied space equal to one. From the integer program, we have $W^M \le 7/6$. Thus, we have

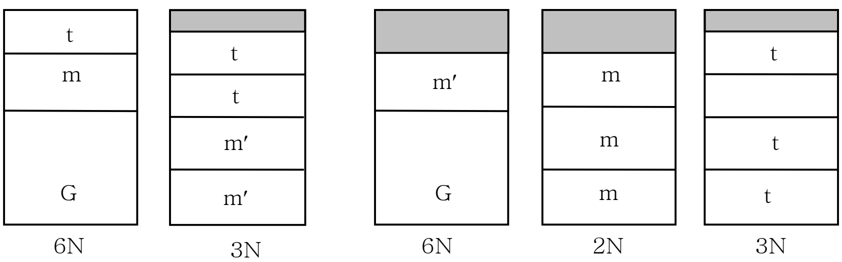
$$z^H - 3 \le \frac{7}{6}z^*$$

$$z^H \le \frac{7}{6}z^* + 3$$

$$\lim_{z^*(L)\to\infty} \frac{z^H(L)}{z^*(L)} \le \frac{7}{6}.$$

3.4.2 If $1/5 < s_n \le 1/4$, then $z^H(L) \le 11/9\, z^*(L)$

According to the FFD rule, an item of type G will be packed with one or two smaller items, two items of item type p_2, two items of item type n_2 with one item of type p_3 or smaller, three items of type p_3, three items of item type n_3 with one item of type p_4 and four items of



Optimal Packing FFD Packing

Figure 2. An example of the worst case example of FFD packing using 11N bins and optimal packing using 9N bins, where $s_{m'} > s_m$

type p_4; except the transition bins with the total weight ≤ 5. Each of the remaining FFD bins has weight ≥1. According to the size and weight in Table 2, we solve the integer program, and get the result as follows:

1. If there is no G item in a bin, then $W^M \leq 7/6$ and J* = ($2p_3$, $2p_4$) or (2m, 2t).
2. If there is a G item in a bin, then $W^M \leq 4/3$ and J* = (G, p_3, p_4) or (G, m, t).

However, when there is a G item in a bin, some of the FFD bins also have total weight greater than one. Hence, we need to adjust the weight of the optimal bin by reducing its weight according to the extra space of the FFD bin. In short,

we subtract the weight of the optimal bin equal to the weight of the FFD bin that is greater than one.

Table 2. Summary of item sizes and weight for $1/5 < s_n \leq 1/4$.

Items Types	Range of Item Sizes	Occupied Space
G	(1/2, 3/5]	3/4
p_2	($(1-s_n)/2$, 1/2]	1/2
n_2	(1/3, $(1-s_n)/2$)	3/8
p_3	($(1-s_n)/3$, 1/3]	1/3
n_3	(1/4, $(1-s_n)/3$]	1/4
p_4	[s_n, 1/4]	1/4

Define a non G item of size ≥ $(1-s_n)/3$ 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 < $(1-s_n)/3$ as an item of type t (that is an item of type n_3 or p_4). In this case, we present the weight adjustment 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 $\in G_{mt}^{m}$ implies an items i of type G 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, $G_{mt}^{t} = \phi$ and $G_{tt}^{mt} = \phi$. Next, consider the weight adjustment. For example, if a G item is packed with an item of type m and t in J^*, then $W(J^*) \leq 4/3$, but packed with m in J, then we can adjust 1/12 to the weight of the optimal bin. Then, consider the following adjustments:

1. For i $\in G_{mt}^{m}$, $W(J^*) \leq 4/3-13/12+1 = 5/4$.
2. For i $\in G_{mt}^{mt}$, $W(J^*) \leq 4/3-4/3+1 = 1$.
3. For i $\in G_{mt}^{tt}$, $W(J^*) \leq 4/3-5/4+1 = 13/12$.
4. For i $\in G_{tt}^{t}$, $W(J^*) \leq 5/4-1+1 = 5/4$.
5. For i $\in G_{tt}^{m}$, $W(J^*) \leq 5/4-13/12+1 = 7/6$.
6. For i $\in G_{tt}^{tt}$, $W(J^*) \leq 5/4-5/4+1 = 1$.

In addition, we found that the worse packing of FFD is the combination of G bin and non G bin as shown in Figure 2. Thus, the maximum weight is the average weight of the proportion of the weight of these bins. Consider the following property:

Property 1: *A t item packed with i $\in G_{tt}^{t}$ in the FFD solution cannot packed with a G bin j $\in G_{tt}^{t}$ in the optimal solution.*

Proof: Suppose there exist such a packing, i.e. let i = G_1 and the t item mentioned is t'_2. Then, $J^*_1 = (G_1, t'_1, t''_1)$, $J^*_2 = (G_2, t'_2, t''_2)$, and t'_2 packed with G_1 only in the FFD packing, i.e. $J_1 = (G_1, t'_2)$ exists. Since t'_2 packed with G_1 only, but packed with G_2 and t''_2 in the optimal solution, $s_{G1} > s_{G2}$. Consider the FFD packing, there must exist an item k of type t packed with G_2 in J must be too large to packed with G_1, i.e. $s_{G1} + s_k > 1$. Since, $s_{G1} < 1-2s_n$, then $s_k > 1- s_{G1} = 2s_n$. If $1/5 < s_n \leq 1/4$, then $s_k > 2/5$ contradicts with the size of t < $(1-s_n)/3$ = 1/4. In addition, if $1/6 < s_n \leq 1/5$, then $s_k > 2/6$ contradicts with the size of t < $(1-s_n)/3$ = 4/15. □

Next, we consider the following possibilities:

1. If i $\in G_{mt}^{m} \neq \varnothing$ and j $\in G_{tt}^{t} = \varnothing$, then an item of type m packed with i in the FFD packing must come from the optimal bin with weight ≤ 7/6, in which at most two of i-bin packed with the bin containing the item of type m. Thus, the average can be calculated as follows:

$$W^M \leq \frac{2}{3}*\frac{5}{4}+\frac{1}{3}*\frac{7}{6}=\frac{11}{9}.$$

2. If i $\in G_{mt}^{m} = \varnothing$ and j $\in G_{tt}^{t} \neq \varnothing$, then by Property 1, an item of type t packed with j in the FFD solution must come from the optimal bin with weight ≤ 7/6. Thus, the average can be calculated as follows:

$$W^M \leq \frac{2}{3}*\frac{5}{4}+\frac{1}{3}*\frac{7}{6}=\frac{11}{9}.$$

3. If i $\in G_{mt}^{m} \neq \varnothing$ and j $\in G_{tt}^{t} \neq \varnothing$, then the average cannot be greater than 11/9. Since there must exist an item k $\in G_{tt}^{m}$ with weight 7/6 packed with i and j and the non-G bins by at most the proportion of 2:2:2:1, respectively. Thus, the average can be calculated as follows:

$$W^M \leq \frac{4}{7}*\frac{5}{4}+\frac{3}{7}*\frac{7}{6}=\frac{17}{14}<\frac{11}{9}.$$

4. If i $\in G_{mt}^{m} = \varnothing$ and j $\in G_{tt}^{t} = \varnothing$, then $W^M \leq 7/6$.

Thus, we have that

$$z^H - 5 \leq \frac{11}{9}z^*.$$

$$z^H \leq \frac{11}{9}z^* + 5.$$

$$\lim_{z^*(L)\to\infty} \frac{z^H(L)}{z^*(L)} \le \frac{11}{9}.$$

3.4.3 If $1/6 < s_n \le 1/5$, then $z^H(L) \le 11/9\, z^*(L)$

According to the FFD rule, an item of type G will be packed with one or two smaller items, two items of item type p_2, two items of item type n_2 with one item of type p_3 or smaller, three items of type p_3, three items of item type n_3 with a smaller item, four items of type p_4, four items of item type n_4 with one item of type p_5 and five items of type p_5; except the transition bins with the total weight ≤ 7. Each of the remaining FFD bins has weight ≥1. According to the size and weight in Table 3, we solve the integer program, and get the result as follows:

1. If there is no G item in a bin, then $W^M \le 71/60$ and $J^* = (p_3, p_4, 3p_5)$ or (m, 4t).
2. If there is a G item in a bin, then $W^M \le 83/60$ and $J^* = (G, p_3, p_4)$ or (G, m, t).

Similar to the previous case, consider the following weight adjustments:

1. For i $\in G^m_{mt}$, $W(J^*) \le$ 83/60-17/15+1 = 5/4.
2. For i $\in G^{mt}_{mt}$, $W(J^*) \le$ 83/60-4/3+1 = 21/20.
3. For i $\in G^{tt}_{mt}$, $W(J^*) \le$ 83/60-5/4+1 = 17/15.
4. For i $\in G^{t}_{tt}$, $W(J^*) \le$ 13/10-21/20+1 = 5/4.
5. For i $\in G^{m}_{tt}$, $W(J^*) \le$ 13/10-17/15+1 = 7/6.
6. For i $\in G^{tt}_{tt}$, $W(J^*) \le$ 13/10-5/4+1 = 21/20.

Table 3. Summary of item sizes and weight for $1/6 < s_n \le 1/5$.

Items Types	Range of Item Sizes	Occupied Space
G	(1/2, 2/3]	4/5
p_2	((1-s_n)/2, 1/2]	1/2
n_2	(1/3, (1-s_n)/2)	2/5
p_3	((1-s_n)/3, 1/3]	1/3
n_3	(1/4, (1-s_n)/3]	4/15
p_4	[(1-s_n)/4, 1/4]	1/4
n_4	(1/5, (1-s_n)/4]	1/5
p_5	[s_n, 1/5]	1/5

Similarly, we consider the following possibilities:

1. If i $\in G^m_{mt} \ne \varnothing$ and j $\in G^t_{tt} = \varnothing$, then an item of type m packed with i in the FFD packing must come from the optimal bin with weight ≤ 71/60, in which at most one of i-bin packed with the bin containing the item of type m. Thus, the average can be calculated as follows:

$$W^M \le \frac{1}{2}*\frac{5}{4}+\frac{1}{2}*\frac{71}{60}=\frac{73}{60}<\frac{11}{9}.$$

2. If i $\in G^m_{mt} = \varnothing$ and j $\in G^t_{tt} \ne \varnothing$, then by Property 1, an item of type t packed with j in the FFD solution must come from the optimal bin with weight ≤ 71/60. Thus, the average can be calculated as follows:

$$W^M \le \frac{1}{2}*\frac{5}{4}+\frac{1}{2}*\frac{71}{60}=\frac{73}{60}<\frac{11}{9}.$$

3. If i $\in G^m_{mt} \ne \varnothing$ and j $\in G^t_{tt} \ne \varnothing$, then the average cannot be greater than 11/9. Since there must exist an item k $\in G^m_{tt}$ with weight 7/6 packed with i and j and the non-G bins by at most the proportion of 2:2:2:1, respectively. Thus, the average can be calculated as follows:

$$W^M \le \frac{4}{7}*\frac{5}{4}+\frac{2}{7}*\frac{7}{6}+\frac{1}{7}*\frac{71}{60}=\frac{73}{60}<\frac{11}{9}.$$

4. If i $\in G^m_{mt} = \varnothing$ and j $\in G^t_{tt} = \varnothing$, then $W^M \le 71/60$.

Thus, we have that

$$z^H - 7 \le \frac{73}{60} z^*.$$

$$z^H \le \frac{73}{60} z^* + 7.$$

$$\lim_{z^*(L)\to\infty} \frac{z^H(L)}{z^*(L)} \le \frac{73}{60}.$$

From all cases, there is no list L that violates Theorem 1. This completes the proof.

4. CONCLUSION

In the past, the asymptotic worst-case ratio of heuristics for the bin packing problem (BPP) has been proved to show the quality of those heuristics. The First Fit Decreasing (FFD) is one of the algorithms that its asymptotic worst-case ratio equals to 11/9. Many researchers prove the asymptotic worst-case ratio by using the weighting function in a lengthy format. In this study, we shorten the proof from two ideas. First, the occupied space in a bin is more than the size of the item. Second, the occupied space in the optimal solution is less than occupied space in the FFD solution. The occupied space is equivalent to the weighting function. The objective is to determine the maximum occupied space of the heuristics by using integer programming with a limited number of variables and constraints. Then, the maximum ratio is derived by matching items case by case as shown in previous section.

ACKNOWLEDGMENT

The author would like to thank Dr. Lap Mui Ann

Chan who is the author advisor for her thoughtful suggestion while the author was studying at Virginia Polytechnic Institute and State University. This research is extended from the author dissertation.

REFERENCES

Baker, B. (1983), A new proof for the first fit decreasing bin-packing algorithm, *Journal of Algorithms*, **6**, 49-70.

Bramel, J. and Simchi-Levi, D. (2001), *The Logic of Logistics*: *Therory*, *Algorithms*, *and Applications for Logistics Management*, Springer, New York.

Coffman, Jr. E., Garey, M., and Johnson, D. (1997), Approximation Algorithms for NP-Hard Problems. PWS Publishing Company, Massachusetts.

Friesen, D. and Langston, M. (1991), Analysis of a compound bin packing algorithm, *SIAM Journal Discrete Mathematics*, **4**, 61-79.

Garey, M. and Johnson D. (1979), *Computers and Intractability*: *A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, New York.

Johnson, D. (1973), *Near-optimal bin packing algorithms*, PhD thesis, MIT, Cambridge, Massachusetts, June.

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

Ongkunaruk, P. (2005), *Asymptotic worst-case analyses for the open bin packing problem*. PhD Dissertation, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, December.

Yue, M. (1991), A simple proof of the inequality FFD (L)≤ 11/9 OPT (L) + 1 ∀ L for the bin-packing algorithm, *ACTA Mathematicae Applicatae Sinica*, **7**, 321-331.

APPENDIX

An integer program for $1/5 < s_n \leq 1/4$ as the occupied space and sizes of items in Table 2.

First, if $s_n = 1/4$, then LS(4) = {1/2+1/2ε, (1-1/4)/2 +1/2ε, 1/3+1/4ε, (1-1/4)/3+1/2ε, 1/4+1/8ε, 1/4} ={25/28, 19/28, 33/96, 13/48, 49/192, 1/4}, where ε = 24. The integer program is as follows:

$$\begin{aligned}
\max\quad & \frac{3}{4}x_G + \frac{1}{2}x_{p_2} + \frac{1}{3}x_{n_2} + \frac{1}{3}x_{p_3} + \frac{1}{4}x_{n_3} + \frac{1}{4}x_{p_4} \\
s.t.\quad & \frac{25}{48}x_G + \frac{19}{48}x_{p_2} + \frac{33}{96}x_{n_2}\ \frac{13}{48}x_{p_3} + \frac{49}{192}x_{n_3} + \frac{1}{4}x_{p_4} \leq 1 \\
\text{or}\quad & 100x_G + 96x_{p_2} + 66x_{n_2} + 52x_{p_3} + 49x_{n_3} + 48x_{p_4} \leq 192 \\
& x_G, x_{p_2}, x_{n_2}, x_{p_3}, x_{n_3}, x_{p_4} \geq 0,\ \text{integer}
\end{aligned}$$

Next, if $s_n = 1/5+1/2\delta = 17/80$, where δ = 40. Then, LS(4) = {1/2 + 1/2ε, (1-17/80)/2 + 1/2ε, 1/3 + 1/4ε, (1-17/80)/3 + 1/2ε, 1/4 + 1/8ε, 17/80} = {81/160, 65/160, 163/480, 66/240, 81/320, 17/80}, where ε = 40. Then, the constraint becomes

$$486x_G + 390x_{p_2} + 326x_{n_2} + 264x_{p_3} + 243x_{n_3} + 204x_{p_4} \leq 960$$

An integer program for $1/6 < s_n \leq 1/5$, as the occupied space and sizes of items in Table 3.

First, if $s_n = 1/5$, then LS(5) = {1/2+1/2ε, (1-1/5)/2 + 1/2ε, 1/3 + 1/4ε, (1-1/5)/3 + 1/2ε, 1/4 + 1/8ε, (1-1/5)/4 + 1/2ε, 1/5 + 1/16ε, 1/5} = {242/480, 194/480, 161/480, 130/480, 241/960, 98/480, 385/1920, 1/5}, where ε = 120. The integer program is as follows:

$$\begin{aligned}
\max\quad & \frac{4}{5}x_G + \frac{1}{2}x_{p_2} + \frac{2}{5}x_{n_2} + \frac{1}{3}x_{p_3} + \frac{4}{15}x_{n_3} + \frac{1}{4}x_{p_4} + \frac{1}{5}x_{n_4} + \frac{1}{5}x_{p_5} \\
s.t.\quad & \frac{242}{480}x_G + \frac{194}{480}x_{p_2} + \frac{161}{480}x_{n_2} + \frac{130}{480}x_{p_3} + \frac{241}{960}x_{n_3} + \frac{98}{480}x_{p_4} \\
& + \frac{385}{1920}x_{n_4} + \frac{1}{5}x_{p_5} \leq 1 \\
\text{or}\quad & 968x_G + 766x_{p_2} + 644x_{n_2} + 520x_{p_3} + 482x_{n_3} + 392x_{p_4} \\
& + 385x_{n_4} + 384x_{p_5} \leq 1920 \\
& x_G, x_{p_2}, x_{n_2}, x_{p_3}, x_{n_3}, x_{p_4}, x_{n_4}, x_{p_5} \geq 0,\ \text{integer}
\end{aligned}$$

Next, if $s_n = 1/6+1/2\delta = 7/40$, where δ = 60. Then, LS(5) = {1/2 + 1/2ε, (1-7/40)/2 + 1/2ε, 1/3 + 1/4ε, (1-7/40)/3 + 1/2ε, 1/4 + 1/8ε, (1-1/5)/4 + 1/2ε, 1/5 + 1/16ε, 7/40}, where ε = 60. Then, the constraint becomes

$$\begin{aligned}
& 488x_G + 404x_{p_2} + 324x_{n_2} + 272x_{p_3} + 242x_{n_3} + 206x_{p_4} \\
& + 193x_{n_4} + 168x_{p_5} \leq 960
\end{aligned}$$