Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Information and Computation 204 (2006) 795–815

Information and Computation

www.elsevier.com/locate/ic

Bin packing problems with rejection penalties and their dual problems

György Dósa [a,*], Yong He [b,†]

[a]*Department of Mathematics, University of Veszprem, Hungary*
[b]*Department of Mathematics, Zhejiang University, Hangzhou 310027, PR China*

Received 22 October 2004; revised 16 January 2006
Available online 18 April 2006

Abstract

In this paper we consider the following problems: we are given a set of n items $\{u_1, \ldots, u_n\}$ and a number of unit-capacity bins. Each item u_i has a size $w_i \in (0, 1]$ and a penalty $p_i \geqslant 0$. An item can be either rejected, in which case we pay its penalty, or put into one bin under the constraint that the total size of the items in the bin is no greater than 1. No item can be spread into more than one bin. The objective is to minimize the total number of used bins plus the total penalty paid for the rejected items. We call the problem *bin packing with rejection penalties*, and denote it as BPR. For the on-line BPR problem, we present an algorithm with an absolute competitive ratio of 2.618 while the lower bound is 2.343, and an algorithm with an asymptotic competitive ratio arbitrarily close to 1.75 while the lower bound is 1.540. For the off-line BPR problem, we present an algorithm with an absolute worst-case ratio of 2 while the lower bound is 1.5, and an algorithm with an asymptotic worst-case ratio of 1.5. We also study a closely related bin covering version of the problem. In this case p_i means some amount of profit. If an item is rejected, we get its profit, or it can be put into a bin in such a way that the total size of the items in the bin is no smaller than 1. The objective is to maximize the number of covered bins plus the total profit of all rejected items. We call this problem *bin covering with rejection* (BCR). For the on-line BCR problem, we show that no algorithm can have absolute competitive ratio greater than 0, and present an algorithm with asymptotic competitive ratio 1/2, which is the best possible. For the off-line BCR problem, we also present an algorithm with an absolute worst-case ratio of 1/2 which matches the lower bound.

* Corresponding author.
E-mail addresses: dosagy@almos.vein.hu (G. Dósa), mathhey@zju.edu.cn (Y. He).
† Deceased.

0890-5401/$ - see front matter
doi:10.1016/j.ic.2006.02.003

AMS Subject Classification: 68Q25; 05B40; 90C27

Keywords: Bin packing; Bin covering; Approximation algorithm; On-line; Off-line

1. Introduction

In this paper, we consider a variant of the classical bin packing problem which has the special feature that items can be rejected at a certain cost. We are given a set of n items $\{u_1, \ldots, u_n\}$ and a number of unit-capacity bins. Each item u_i has a size $w_i \in (0, 1]$ and a penalty $p_i \geqslant 0$. The cost of purchasing one bin is 1. An item can be either rejected, in which case we pay its penalty, or put into one bin under the constraint that the total size of the items in the bin, called the *content* of the bin, is no greater than 1. No item can be spread into more than one bin. The objective is to minimize the sum of the cost for purchasing bins and the penalties of all rejected items. We call this problem *bin packing with rejection penalties* or BPR for short.

This problem may have applications in the real world. Let us first consider a company's intranet. The intranet enables the company's employees to access both the company's own data files and external files via the Internet [4]. If employees access an external file often, the administrator of the intranet might copy it to a file server on the intranet to decrease the communication cost. When a server reaches its capacity, a new server must be purchased. We can model this situation as follows. We have a list of external files considered for copying, each with a size and penalty, where the penalty is the estimated communication cost if it is not duplicated on the intranet. The goal is to minimize the total cost, i.e., the sum of the cost for purchasing file servers and the total penalties for files that are not duplicated. Clearly this problem can be reduced to BPR.

As another example, consider a shipping company that owns a fleet of trucks. The company can pack a shipment of goods in its own trucks, provided that the total size of the goods packed in each truck is no greater than the truck's capacity. Or the company can assign the goods to a forwarding agent. Suppose all of the company's trucks have the same capacity and the same cost. The total cost for the shipment is the sum of the cost of its trucks and the cost for sending the goods via the forwarding agent. The goal is to send the shipment in the cheapest way, to minimize the total cost. This problem is essentially BPR.

We also consider the dual problem of BPR in this paper, which can be described as follows: we are given a set of n items $\{u_1, \ldots, u_n\}$, each item u_i is characterized by its size $w_i \in (0, 1]$ and its profit $p_i \in [0, 1)$, and a number of unit-capacity bins. An item can be either rejected, in which case we obtain its profit, or assigned to one bin. If one bin has a content of at least 1, then we say that it is *covered* and provides profit 1. No item can be spread into different bins. The objective is to maximize the total profit, i.e., the sum of the number of covered bins and the profits of all rejected items. This problem can also be viewed as a variant of the classical bin covering problem, hence we also refer it to *bin covering with rejection profits* or BCR for short.

The problem BCR may model the following application. An industry company holds several monopolies. It can choose to break them up into smaller companies, each of which must be large enough to be viable, or sell them to obtain profit. Suppose that the profit of each viable company is the same. The objective is to maximize the total profit, i.e., the number of viable companies and the profit of those monopolies which are sold.

For the problem BPR, if all $p_i > 1$, $i = 1, \ldots, n$, then no item can be rejected in an optimal solution. Hence the standard bin packing problem is a special case of BPR. Similarly, in BCR, if all $p_i = 0$, then again no item can be rejected in an optimal solution. Hence the bin covering problem is a special case of BCR. It follows that both BPR and BCR are strongly NP-hard [12].

Bin packing is a classical combinatorial optimization problem that has been extensively studied for more than three decades. The readers may refer to survey papers [5,6] and papers cited therein. Meanwhile, the bin covering problem is also well-studied since it was first proposed by Assman et al. [1]. The first APTAS for bin packing was given by [11], and the first APTAS for bin covering in [7]. The readers may refer to recent papers [7,16], survey paper [6], and papers cited therein. But to the authors' best knowledge, both BPR and BCR are unexplored.

In the *on-line* versions of BPR and BCR, items arrive one by one, and the algorithm must decide either reject an item or pack it into a bin before any information about the next item is revealed. If we are allowed to make decisions with full information of the set of items, the version is called *off-line*.

The quality of an approximation algorithm is usually measured by its worst-case ratio (for the off-line version) or competitive ratio (for the on-line version), respectively. Let I be an instance of the problem. then let $A(I)$ denote the objective value produced by an approximation algorithm A, and let $OPT(I)$ denote the optimal objective value in the off-line version. Then for the problem BPR, the *absolute worst-case ratio* (or *absolute competitive ratio*) of A, denoted by R_A, is defined by

$$R_A = \sup_I \left\{ \frac{A(I)}{OPT(I)} \right\}.$$

The *asymptotic worst-case ratio* (or *asymptotic competitive ratio*) of A, denoted by R_A^∞, is defined by

$$R_A^\infty = \lim \sup_{n\to\infty} \max \left\{ \frac{A(I)}{OPT(I)} \mid OPT(I) = n \right\}.$$

For the problem BCR, the absolute worst-case ratio (absolute competitive ratio) and asymptotic worst-case ratio (asymptotic competitive ratio) of A, are, respectively, defined by

$$R_A = \inf_I \left\{ \frac{A(I)}{OPT(I)} \right\}, \quad R_A^\infty = \lim \inf_{n\to\infty} \min \left\{ \frac{A(I)}{OPT(I)} \mid OPT(I) = n \right\}.$$

In this paper, an off-line (on-line) minimization/maximization problem has a *lower /upper bound* ρ with respect to absolute or asymptotic worst-case (competitive) ratio if no off-line (on-line) algorithm has a absolute or asymptotic worst-case (competitive) ratio smaller/greater than ρ, respectively. An off-line (on-line) algorithm is called *best possible*if its worst-case (competitive) ratio matches the corresponding lower/upper bound of the minimization/maximization problem.

Table 1
Summary of the results for BPR and BCR

	BPR		BCR	
	Upper bound	Lower bound	Lower bound	Upper bound
On-line, abs.	2.618 (*RFF*1)	2.343	0	0
On-line, asymp.	1.75+ϵ (*RFF*2)	1.540 [24]	0.5 (*MDNF*)	0.5 [8]
Off-line, abs.	2 (*RFF*3)	1.5 [12]	0.5 (*MDNFD*)	0.5 [1]
Off-line, asymp.	1.5 (*RFF*4)	open	0.5 (*MDNF*)	open

Table 2
Summary of the known best results for bin packing and covering problems

	Bin packing problem		Bin covering problem	
	Upper bound	Lower bound	Lower bound	Upper bound
On-line, abs.	1.75 [26]	1.5 [25]	0.5 (*DNF*) [1]	open
Oon-line, asymp.	1.589 [21]	1.540 [24]	0.5 (*DNF*) [1]	0.5 [8]
Off-line, abs.	1.5 [23]	1.5 [20]	0.5 (*DNF*) [1]	0.5 [1]
Off-line, asymp.	*FPTAS* [18]	1	*FPTAS* [16]	1

In this paper, we study the problems BPR and BCR. Both on-line and off-line versions are considered. The results are listed in Table 1. Algorithms *RFF*1–*RFF*4 and *MDNFD* run in time $O(n \log n)$, and algorithm *MDNF* runs in time $O(n)$. For comparison purposes, we also list the known best results of bin packing and covering problems in Table 2. As we know, even for the classical bin packing and covering problems, it took a long time to obtain most of the results in Table 2, and it is still open how to close the existing gaps. The BPR and BCR problems are more complicated, and harder to approximate since one more parameter is introduced for every item. To devise the algorithms presented in this paper, we will introduce several strategies for trade-off between packing cost/profit and rejection penalty/profit. Furthermore, we will employ the harmonic technique with consideration of penalty parameter. In the analysis of the algorithms, we will develop methods to estimate the optimal value. Especially we will apply a simple linear programming technique instead of case by case analysis to prove the asymptotic competitive ratio of algorithm *RFF*2 (see Theorem 7).

A related problem is the parallel machine scheduling with rejection, which can be described as follows: we have m parallel machines and n independent jobs. Each job is characterized by its processing time and its penalty. A job can be either rejected, in which case we pay its penalty, or scheduled on one of the machines, in which case it contributes its processing time to the completion time of that machine. The objectives are to minimize the sum of the penalties of all rejected jobs and one of the scheduling criteria such as the makespan, weighted completion times, or lateness/tardiness of all accepted jobs. Refs. [2,13,15,20] considered the first objective, Refs. [9,10] considered the second objective, and Ref. [22] considered the last objective.

This paper is organized as follows. Section 2 considers the problem BPR. Section 3 studies the problem BCR. Some final remarks are presented in Section 4. A preliminary version of this paper appeared in a conference proceedings [14].

In the remainder of this paper, let $W(S) = \sum_{u_i \in S} w_i$ and $P(S) = \sum_{u_i \in S} p_i$ for an item set S. Define $M_1 = \{u_i | p_i/w_i > 1\}$ and $M_2 = \{u_i | p_i/w_i \leqslant 1\}$. Define by $\varepsilon > 0$ a sufficiently small number, and by N

a sufficiently large positive integer, whose exact values are immaterial in later proofs. For simplicity, in the on-line setting we use the notation $u_i \in M_1$ to mean $p_i/w_i \leqslant 1$, and we use the notation $u_i \in M_2$ to mean $p_i/w_i \leqslant 1$, even though not all items have arrived.

2. The problem BPR

2.1. Preliminaries

We use I' to denote an instance of the bin packing problem, $OPT'(I')$ to denote the minimum number of bins used in an optimal solution for I' with respect to the bin packing problem. Because we will use the classic algorithms First Fit (FF) and First Fit Decreasing (FFD), thus we first explain them briefly. FF always packs the next item into the first bin, where it fits. If the next item does not fit into any bin, it is packed into a new bin. In case of algorithm FFD first the items are ordered into the non-increasing order of their sizes, and then we execute FF.

Lemma 1. *For any instance I' of the bin packing problem with $|I'| = n'$ and $w_i \leqslant 1/2, i = 1, \ldots, n'$, we have*

$$FF(I') \begin{cases} = 1 & \text{if } \sum_{i=1}^{n'} w_i \leqslant 1, \\ < \frac{3}{2}\sum_{i=1}^{n'} w_i + \frac{1}{2} & \text{if } \sum_{i=1}^{n'} w_i > 1, \end{cases}$$

where $FF(I')$ denotes the number of bins used by First Fit (FF) algorithm.

Proof. The result for the case $\sum_{i=1}^{n'} w_i \leqslant 1$ is trivial. Hence we assume that $\sum_{i=1}^{n'} w_i > 1$ and $FF(I') = k \geqslant 1$. Johnson et al. [17] proved that $FF(I') < \frac{3}{2}\sum_{i=1}^{n'} w_i + 2$. We show here that the additive constant can be tightened to 1/2.

Let the total size of items in the ith bin be s_i, $i = 1, \ldots, k$. If $s_i \geqslant 2/3$ holds for every $i = 1, \ldots, k$, then we obtain that $k \leqslant \frac{3}{2}\sum_{i=1}^{k} s_i = \frac{3}{2}\sum_{i=1}^{n'} w_i$, and we are done. Hence we suppose that there is at least one index i such that $s_i < 2/3$. Let $j = \min\{i : s_i < 2/3\}$. If $j < k$, let u_l be any item in the j'th bin by FF algorithm, where $j' > j$. Then $w_l > 1/3$. It states that all items in the $j + 1$th, ..., kth bins have a size of at least 1/3. Note that every bin except the last one must contain at least two items since every item has size at most 1/2. Hence each of the $j + 1$th, $\cdots$, $k - 1$th bins has items of total size at least 2/3.

Since $s_j + s_k > 1$, we obtain that $\sum_{i=1}^{n'} w_i = \sum_{i=1}^{k} s_i > \frac{2}{3}(k-2) + 1 = \frac{2}{3}k - \frac{1}{3}$, i.e., $k < \frac{3}{2}\sum_{i=1}^{n'} w_i + \frac{1}{2}$. If $j = k$, then the last two bins have total content at least 1 and all remaining bins have content at least 2/3. The result can be obtained similarly. □

The next lemma can be proved in a similar way:

Lemma 2 ([3,5]). *For any instance I' of the classical bin packing problem with $|I'| = n'$, we have*

$$FF(I') \begin{cases} = 1 & \text{if } \sum_{i=1}^{n'} w_i \leqslant 1/2, \\ < 2\sum_{i=1}^{n'} w_i & \text{if } \sum_{i=1}^{n'} w_i > 1/2. \end{cases}$$

Corollary 3. $\sum_{i=1}^{n'} w_i \geqslant (OPT'(I') - 1)/2$.

Theorem 4. *For BPR, $L = W(M_1) + P(M_2)$ is a lower bound of the optimal value, and $L \geqslant (OPT(I) - 1)/2$ for any instance I.*

Proof. First we prove that $OPT(I) \geqslant L$. To see it, we assume that in an optimal solution there are k used bins. Let S_1 and S_2 denote sets consisting of rejected items and accepted items in the optimal solution, respectively. We obtain

$$OPT(I) = k + \sum_{u_j \in S_1} p_j \geqslant \sum_{u_j \in S_2} w_j + \sum_{u_j \in S_1} p_j \geqslant \sum_{u_j \in S_2 \cup S_1} \min\{w_j, p_j\} = \sum_{u_j \in M_1} w_j + \sum_{u_j \in M_2} p_j = L.$$

Consider a feasible solution that accepts all items in M_1, packs them optimally, and rejects all items in M_2. Hence we have $OPT(I) \leqslant OPT'(M_1) + P(M_2)$. Combining this inequality with Corollary 3, we obtain $W(M_1) \geqslant (OPT'(M_1) - 1)/2 \geqslant (OPT(I) - P(M_2) - 1)/2$. It follows that

$$L = W(M_1) + P(M_2) \geqslant \frac{OPT(I) - P(M_2) - 1}{2} + P(M_2) \geqslant \frac{OPT(I) - 1}{2}. \quad \square$$

2.2. On-line algorithms

First, we present a simple 2.618-competitive algorithm: All of the initial sequence of items are rejected until they accumulate a penalty of not more than 0.618; thereafter, reject all items such that $p_k \leqslant w_k$ and apply First Fit to pack the other items. Formally:

Algorithm *RFF*1:

1. $k = 1, P = 0$. Let $\phi = \frac{\sqrt{5}-1}{2} \approx 0.618$.
2. If no new item arrives, stop. Else go to 3.
3. If $P + p_k < \phi$, then reject u_k, set $P = P + p_k$, $k = k + 1$ and go to 2; Otherwise go to 4.
4. If $u_k \in M_1$, then pack it by *FF* algorithm; Otherwise reject it. Set $k = k + 1$. If no new item arrives, stop. Else go back to 4.

Theorem 5. $R_{RFF1} = \frac{1+\phi}{\phi} = 2 + \phi = \frac{\sqrt{5}+3}{2} \approx 2.618$.

Proof. We first show that $\frac{RFF1(I)}{OPT(I)} \leqslant \frac{1+\phi}{\phi}$ for any instance I. If the total penalty of all items is less than ϕ, then *RFF*1 rejects all items and thus yields an optimal solution. Let I' be the set consisting of all items packed or rejected by Step 4. We then have that $RFF1(I \setminus I') < \phi$. Define $M_1' = \{u_i \in M_1 | u_i \in I'\}$, and $M_2' = \{u_i \in M_2 | u_i \in I'\}$. We distinguish three cases to obtain the result.

Case 1. $OPT(I) < 1$. Thus, in an optimal solution all items must be rejected, and $OPT(I) = \sum_{i=1}^{n} p_i < 1$. If $M_1' = \emptyset$, *RFF*1 yields the optimal solution. Hence we assume $M_1' \neq \emptyset$. By the definition of M_1 and $\sum_{i=1}^{n} p_i < 1$, we obtain that $W(M_1') < P(M_1') < 1$, and *RFF*1 uses exactly one bin. Hence

$$RFF1(I) = 1 + RFF1(I \setminus I') + P\left(M_2'\right).$$

On the other hand,

$$OPT(I) \geqslant \max\left\{\phi, RFF1(I \setminus I') + P\left(M_2'\right)\right\}$$

holds trivially, thus we obtain

$$\frac{RFF1(I)}{OPT(I)} \leqslant \frac{1 + RFF1(I \setminus I') + P\left(M_2'\right)}{\max\left\{\phi, RFF1(I \setminus I') + P\left(M_2'\right)\right\}} < \frac{1+\phi}{\phi}.$$

Case 2. $OPT(I) \geqslant 1$ and $W(M_1') \geqslant 0.5$. By Lemma 2 and Theorem 4, we obtain

$$RFF1(I') \leqslant 2W(M_1') + P\left(M_2'\right) \leqslant 2\left(W(M_1') + P\left(M_2'\right)\right) \leqslant 2OPT(I').$$

Therefore, we have

$$RFF1(I) = RFF1(I \setminus I') + RFF1(I') < \phi + 2OPT(I') \leqslant \phi + 2OPT(I) \leqslant (2+\phi)\, OPT(I).$$

Case 3. $OPT(I) \geqslant 1$ and $0 < W(M_1') < 0.5$. Thus $RFF1$ uses only one bin, and

$$RFF1(I) = RFF1(I \setminus I') + 1 + P\left(M_2'\right) < 1 + \phi + P\left(M_2'\right).$$

t is trivial that $OPT(I) \geqslant \max\left\{1, P\left(M_2'\right)\right\}$. We thus obtain

$$\frac{RFF1(I)}{OPT(I)} < \frac{1+\phi+P\left(M_2'\right)}{\max\left\{1, P\left(M_2'\right)\right\}} \leqslant 2+\phi.$$

The following item sequence shows that the absolute competitive ratio of algorithm $RFF1$ cannot be smaller than $2+\phi$. Let $w_1 = \varepsilon, p_1 = \phi - \varepsilon, w_2 = \varepsilon, p_2 = 2\varepsilon, w_3 = 1 - 2\varepsilon, p_3 = 1 - 3\varepsilon$. Then $RFF1$ rejects the first and third items and accepts the second one, while all items are packed into one bin in an optimal solution. Therefore, $RFF1(I)/OPT(I) = (2+\phi-4\varepsilon)/1 \to 2+\phi$ as $\varepsilon \to 0$. □

Next we consider the lower bound of the problem BPR in terms of the absolute competitive ratio. Consider the following system of equations:

$$\frac{2+\beta+x}{1+x} = \frac{1+\beta}{\beta} = \beta + 2x + 1, \tag{1}$$

where β and x are real variables. This system leads to the following equations

$$\beta\,(1+\beta)\,(1+2\beta) = 1 + 3\beta, \quad x = \frac{\beta}{1+2\beta}. \tag{2}$$

Since function $f(\beta) = \beta\,(1+\beta)\,(1+2\beta)$ is convex for $\beta \geqslant 0$, and $f(0) = 0$ while the value of $1+3\beta$ for $\beta = 0$ is 1, follows that (2) can have at most one positive solution. Because $f'(\beta) \to \infty$ as $\beta \to \infty$, follows that the equation has a unique positive solution, which can be computed for example by MAPLE. Define $\theta = (-1/3)\arctan((2/9)\sqrt{237}) + \pi/3$, and $\beta_0 = \sqrt{21}e^{i\theta}$. Then the exact solution of (2) is $\beta = \frac{1}{6}\beta_0 + \frac{7}{2\beta_0} - \frac{1}{2} \approx 0.7446$, and $x \approx 0.2991$. In the next theorem we will construct such instances I, for which the ratio $A(I)/OPT(I)$ is bounded by below by some term of (1). Thus, we obtain the best lower bound, if the three terms are equal. $\leftarrow$

Theorem 6. *No deterministic on-line algorithm can have an absolute competitive ratio of less than* $\frac{1+\beta}{\beta} \approx 2.343$.

Proof. We show the result by adversary method. Recall that N denotes a sufficiently big number, and ε is chosen in such a way that even $N\varepsilon$ is sufficiently small. Let I_1 consist of N copies of $(\frac{\varepsilon}{2N}, \frac{\beta}{N})$, I_2 is just one item $\left(\frac{x}{N} - 2\varepsilon, \frac{x}{N} - 2\varepsilon\right)$, I_3 consists of $N-1$ copies of $\left(\frac{x}{N} - \varepsilon, \frac{x}{N} - \varepsilon\right)$, I_4 is also one item (x,x), and let $I' = I_1 \cup I_2 \cup I_3 \cup I_4$, it consists of $2N+1$ items. Then all sequences I which are used to obtain the desired result, when we remove the appropriately chosen last 0, 1 or 2 items, are all prefixes of sequence I'. The adversary watches the algorithm A until the time t when the first item which is packed. If the first accepted item is some item of I', then the next items of I' do not come, but we append 0, 1 or 2 well-chosen items depending on t, so that the objective function has value at least $(1+\beta)/\beta$. Define $y = \frac{x}{N} - 2\varepsilon$.

If u_k, $1 \leqslant k \leqslant N$, is the first accepted item by an on-line algorithm A, then no more item comes. We have $OPT(I) = \frac{k\beta}{N}$, and $A(I) = \frac{(k-1)\beta}{N} + 1$. Thus

$$\frac{A(I)}{OPT(I)} = \frac{(k-1)\beta + N}{k\beta} \geqslant \frac{(N-1)\beta + N}{N\beta} \to \frac{1+\beta}{\beta} \text{ as } N \to \infty.$$

If u_{N+1} is the first accepted item by A, two copies of $(\frac{1}{2} - \frac{\varepsilon}{4}, 1)$ come. Then A may accept items $u_{N+1}, u_{N+2}, u_{N+3}$ and pack them into two bins (the total size of these three items is larger than 1), while in an optimal solution only u_{N+1} is rejected and all remaining items are packed into one bin. As $\sum_{i=1}^{N} p_i = \beta$, we obtain that $A(I) \geqslant 2+\beta$, and $OPT(I) = 1+y$. We thus obtain

$$\frac{A(I)}{OPT(I)} \geqslant \frac{2+\beta}{1+y} = \frac{2+\beta}{1+x/N-2\varepsilon} > \frac{1+\beta}{\beta} \text{ as } N \to \infty.$$

If $u_{N+k+1}, 1 \leqslant k \leqslant N-1$, is the first accepted item by A, two copies of $(\frac{1}{2} - \frac{x}{2N} + \frac{3}{4}\varepsilon, 1)$ come. Then similarly as above, A has to pack items $u_{N+k+1}, u_{N+k+2}, u_{N+k+3}$ into two bins, while in an optimal solution items $u_1, u_2, \ldots, u_{N+1}$ and u_{N+k+2}, u_{N+k+3} are packed into one bin, and all remaining items are rejected. Therefore, we have $A(I) \geqslant 2+\beta+ky+(k-1)\varepsilon$, and $OPT(I) = 1+k(y+\varepsilon)$. It follows that

$$\frac{A(I)}{OPT(I)} \geqslant \frac{2+\beta+ky+(k-1)\varepsilon}{1+k(y+\varepsilon)} \to \frac{2+\beta+ky}{1+ky} \geqslant \frac{2+\beta+x}{1+x} \text{ as } \varepsilon \to 0.$$

If u_{2N+1} is the first accepted item by A, two copies of $(\frac{1}{2} - \frac{x}{2} + \frac{1}{2}(N+\frac{1}{2})\varepsilon, 1)$ come. Then as before, A has to pack items $u_{2N+1}, u_{2N+2}, u_{2N+3}$ into two bins. Noting that the sum of penalties of the first $2N$ items is $\beta + y + (N-1)(y+\varepsilon) = \beta + x - (N+1)\varepsilon$, we have $A(I) \geqslant 2+\beta+x-(N+1)\varepsilon$. At the same time, only u_{2N+1} is rejected and all remaining items are packed into one bin in an optimal solution. Hence $OPT(I) = 1+x$. It follows that

$$\frac{A(I)}{OPT(I)} \geqslant \frac{2+\beta+x-(N+1)\,\varepsilon}{1+x} \rightarrow \frac{2+\beta+x}{1+x} = \frac{1+\beta}{\beta} \text{ as } \varepsilon \rightarrow 0.$$

Hence we assume that algorithm A rejects the first $2N+1$ items. Then the last item is $(\varepsilon, 1)$. No matter whether this item is rejected, we have $A(I) = 1 + \sum_{i=1}^{2N+1} p_i = 1 + \beta + 2x - (N+1)\,\varepsilon$ while $OPT(I) = 1$. Thus, we obtain

$$\frac{A(I)}{OPT(I)} = 1+\beta+2x-(N+1)\,\varepsilon \rightarrow 1+\beta+2x = \frac{1+\beta}{\beta} \text{ as } \varepsilon \rightarrow 0.$$

Therefore, the proof is completed. □

Now we turn to study the asymptotic competitive ratios of on-line algorithms. To show the asymptotic competitive ratio of $RFF1$, we consider the following sequence with $2N+1$ items: $w_1 = \varepsilon, p_1 = \phi - \varepsilon$, $w_2 = \cdots = w_{2N+1} = \varepsilon + 1/2, p_2 = \cdots = p_{2N+1} = 2\varepsilon + 0.5$. Then $RFF1$ rejects only the first item while the optimal solution only accepts the first two items. Therefore $\frac{RFF1(I)}{OPT(I)} = \frac{2N+\phi-\varepsilon}{N+4N\varepsilon-2\varepsilon+1/2} \rightarrow 2$ as $N\varepsilon \rightarrow 0$, and $N \rightarrow \infty$. It follows that the asymptotic competitive ratio of $RFF1$ cannot be smaller than 2. In the following, we present another on-line algorithm $RFF2$ with an asymptotic competitive ratio of arbitrarily close to 7/4.

Let m be a large positive integer. We construct the next sets: $S_1 = \{j \,|\, m+1 \leqslant j \leqslant 2m\}$, and $S_2 = \{(j,k) \,|\, j \in S_1, m+1 \leqslant k \leqslant j-1\}$. We partition set M_1 into several subsets as follows:

$$\begin{aligned}
&M_{11} = \{u_i \in M_1 | w_i \leqslant \tfrac{1}{2}\},\\
&M_{jk} = \left\{u_i \in M_1 | \tfrac{m}{j} < w_i \leqslant \tfrac{m}{j-1}, \tfrac{k-1}{m} < \tfrac{p_i}{w_i} \leqslant \tfrac{k}{m}\right\}, (j,k) \in S_2\\
&M_{jj} = \{u_i \in M_1 | \tfrac{m}{j} < w_i \leqslant \tfrac{m}{j-1}, \tfrac{j-1}{m} < \tfrac{p_i}{w_i}\}, j \in S_1.
\end{aligned}$$

Then the algorithm can be described as follows:

Algorithm $RFF2$:

1. If the incoming item is in M_{11}, pack it by FF algorithm.
2. If the incoming item is in M_{jk}, $(j,k) \in S_2$, or M_2, reject it.
3. If the incoming item is in $M_{jj}, j \in S_1$, then pack it each into a bin, and this bin will not be used to pack any other item.

Theorem 7. *For any given positive integer* $m \geqslant 2, R^{\infty}_{RFF2} \leqslant \frac{7m-3}{4m-2}$, *hence there exists an on-line algorithm with an asymptotic competitive ratio of arbitrarily close to* $7/4 = 1.75$.

Proof. Since all items in M_{11} have sizes no greater than 1/2, and are packed by FF algorithm, we have

$$RFF2(M_{11}) = FF(M_{11}) \leqslant \max\left\{\frac{3}{2}W(M_{11}) + \frac{1}{2}, 1\right\} \leqslant \frac{3}{2}W(M_{11}) + 1$$

by Lemma 1. By the definition of M_{jk}, $(j,k) \in S_2$, we have $P(M_{jk}) \leqslant \frac{k}{m} W(M_{jk})$. Since every item in M_{jj} does not share a bin with any other item in *RFF*2 algorithm, the number of bins used for the items in M_{jj} is

$$|M_{jj}| \leqslant \sum_{u_i \in M_{jj}} \frac{j}{m} w_i = \frac{j}{m} W(M_{jj}),$$

where the inequality holds because $\frac{m}{j} < w_i$, i.e., $\frac{j}{m} w_i > 1$, for any item $u_i \in M_{jj}$. Therefore, we have

$$RFF2(I) \leqslant \frac{3}{2} W(M_{11}) + 1 + \sum_{(j,k) \in S_2} \frac{k}{m} W(M_{jk}) + \sum_{j \in S_1} \frac{j}{m} W(M_{jj}) + P(M_2). \tag{3}$$

By Theorem 4, we have

$$OPT(I) \geqslant W(M_{11}) + \sum_{(j,k) \in S_2} W(M_{jk}) + \sum_{j \in S_1} W(M_{jj}) + P(M_2). \tag{4}$$

Next we are going to obtain another lower bound of *OPT*(*I*). Let

$$I' = I \setminus (M_{11} \cup M_2) = \bigcup_{j=m+1,\ldots,2m, k=m+1,\ldots,j} M_{jk}.$$

It is obvious that $OPT(I') \leqslant OPT(I)$. Let us consider an optimal solution for instance I'. For every item $u_i \in M_{jk} \subseteq I'$, $(j,k) \in S_2$, since $w_i > 1/2$, this item cannot share a bin with any other item in I', if the item is accepted in the optimal solution. In this case, the item's contribution to objective value is $1 \geqslant \frac{j-1}{m} w_i$. If this item is rejected in the optimal solution, the contribution to objective value is $p_i \geqslant \frac{k-1}{m} w_i$. Hence we conclude that the contribution of item u_i to the optimal value is at least $\min\left\{\frac{j-1}{m}, \frac{k-1}{m}\right\} w_i = \frac{k-1}{m} w_i$. Similarly, for any $u_i \in M_{jj}, j \in S_1$, its contribution to the optimal value of I' is at least $\frac{j-1}{m} w_i$. Therefore, we obtain

$$OPT(I) \geqslant OPT(I') \geqslant \sum_{(j,k) \in S_2} \frac{k-1}{m} W(M_{jk}) + \sum_{j \in S_1} \frac{j-1}{m} W(M_{jj}). \tag{5}$$

To obtain the asymptotic competitive ratio, we need to show that there exist constants a, b such that $\frac{RFF2(I)}{OPT(I)} \leqslant a + \frac{b}{OPT(I)}$ for any instance I. We are interested in the maximum possible value of $\frac{RFF2(I)}{OPT(I)}$. Dividing the inequalities (3)–(5) by *OPT*(*I*), we construct a linear program as follows:

$$\begin{aligned} \max \quad & z = \frac{3}{2} x_{11} + \sum_{(j,k) \in S_2} \frac{k}{m} x_{jk} + \sum_{j \in S_1} \frac{j}{m} x_{jj} + y \\ \text{s.t.} \quad & x_{11} + \sum_{(j,k) \in S_2} x_{jk} + \sum_{j \in S_1} x_{jj} + y \leqslant 1, \\ & \sum_{(j,k) \in S_2} \frac{k-1}{m} x_{jk} + \sum_{j \in S_1} \frac{j-1}{m} x_{jj} \leqslant 1, \\ & x_{11} \geqslant 0, x_{jk} \geqslant 0, \ m+1 \leqslant k \leqslant j \leqslant 2m, \quad y \geqslant 0, \end{aligned} \tag{6}$$

where $x_{11} = \frac{W(M_{11})}{OPT(I)}, x_{jk} = \frac{W(M_{jk})}{OPT(I)}, m+1 \leqslant k \leqslant j \leqslant 2m, y = \frac{P(M_2)}{OPT(I)}$, and $z = \frac{RFF2(I)}{OPT(I)}$ are variables. Note that we omit the additive factor $1/OPT(I)$ from the objective function of the linear program, since we are considering the asymptotic competitive ratio. It is clear that the asymptotic competitive ratio is no greater than the optimal value of (6). We state that (6) has a unique optimal solution as follows: $x_{11} = \frac{m-1}{2m-1}, x_{2m,2m} = \frac{m}{2m-1}$, and the values of all remaining variables equal to 0. The optimal value is $\frac{3}{2}\frac{m-1}{2m-1} + 2\frac{m}{2m-1} = \frac{7m-3}{4m-2}$ which can be arbitrarily close to 7/4 if m is chosen to be large enough. To obtain the optimum solution one need to have some intuition about the LP, but it is easy to verify that this is really the unique optimum solution: The optimal basis is $\left[a_{11}, a_{2m,2m}\right] = \begin{bmatrix} 1 & 1 \\ 0 & \frac{2m-1}{m} \end{bmatrix}$, the pricing vector is $\pi' = c'_B B^{-1} = \left[3/2, \frac{m}{2(2m-1)}\right]$. Thus, $\pi' a_{jk} - c_{jk} = \left[3/2, \frac{m}{2(2m-1)}\right]\left[1, \frac{k-1}{m}\right]' - k/m = 3/2 + \frac{k-1}{2(2m-1)} - k/m$, and $\pi' a_{jj} - c_{jj} = \left[3/2, \frac{m}{2(2m-1)}\right]\left[1, \frac{j-1}{m}\right]' - j/m = 3/2 + \frac{j-1}{2(2m-1)} - j/m$, and these values are nonnegative for any possible values of k and m. Thus, by complementary slackness the solution is optimal, and the proof is completed. □

For the classical bin packing problem, no on-line algorithm can have an asymptotic competitive ratio of less than 1.540 [24], we conclude that no on-line algorithm can have an asymptotic competitive ratio of less than 1.540 for the problem BPR, too. It is open whether a larger lower bound of the problem BPR in terms of asymptotic competitive ratio exists.

2.3. Off-line algorithms

Algorithm *RFF*3:

1. For $\sum_{i=1}^n w_i \leqslant 1$, if $\sum_{i=1}^n p_i > 1$, then all items are accepted and packed into one bin, otherwise reject all jobs.
2. For $\sum_{i=1}^n w_i > 1$, if $u_i \in M_1, i = 1, 2, \ldots, n$, then pack it by *FF* algorithm, otherwise reject it.

Theorem 8. $R_{RFF3} = 2$.

Proof. For the case $\sum_{i=1}^n w_i \leqslant 1$, it is clear that *RFF*3 yields an optimal solution. For the case $\sum_{i=1}^n w_i > 1$, we have $FF(M_1) \leqslant 2\sum_{i \in M_1} w_i$ by Lemma 2. Combining this result with Theorem 4, we have

$$RFF3(I) = FF(M_1) + P(M_2) \leqslant 2\sum\nolimits_{u_i \in M_1} w_i + P(M_2)$$
$$\leqslant 2(\sum\nolimits_{u_i \in M_1} w_i + P(M_2)) \leqslant 2OPT(I).$$

To see that the ratio cannot be smaller than 2, we consider the following instance: We have a total of $2N$ items; each item has the same size $\varepsilon + 1/2$ and the same penalty $2\varepsilon + 1/2$. It is clear that $RFF3(I)/OPT(I) = 2N/(N + 4N\varepsilon) \to 2$ as $\varepsilon \to 0$. □

From the instance given in the proof of Theorem 8, it is clear that $R^\infty_{RFF3} = 2$. Furthermore, the absolute and asymptotic worst-case ratios of *RFF*3 cannot become smaller if we replace *FF* algorithm by *First Fit Decreasing* (*FFD*) algorithm in Step 2. Note that there does not exist a polynomial algorithm with an absolute worst-case ratio of smaller than 3/2 for the classical bin packing problem, unless P=NP [20], this statement still holds for the problem *BPR*.

In the remainder of this section, we present an off-line algorithm with an asymptotic worst-case ratio of 3/2.

First we split M_1 into four subsets as follows:

$$M_{1h} = \left\{u_i \in M_1 : w_i > \frac{2}{3}\right\}, \quad M_{1l} = \left\{u_i \in M_1 : \frac{1}{2} < w_i \leqslant \frac{2}{3}\right\},$$
$$M_{1m} = \left\{u_i \in M_1 : \frac{1}{3} < w_i \leqslant \frac{1}{2}\right\}, \quad M_{1s} = \left\{u_i \in M_1 : w_i \leqslant \frac{1}{3}\right\},$$

and the items in $M_{1h}, M_{1l}, M_{1m}, M_{1s}$ are called as *huge, large, medium* and *small* items respectively. We propose a procedure for pre-processing as follows, which puts one large item and one medium item pairwise into one bin as much as possible. Define by $\max(I)$ the maximum number k such that k large items and k medium items can be packed pairwise into k bins for instance I. Then the next proposition holds.

Proposition 9. *Suppose that $u_{l1}, u_{l2}, \ldots, u_{lk}$ are k large items, and $u_{m1}, u_{m2}, \ldots, u_{mk}$ are k medium items such that u_{li} and u_{mi} can be packed into a common bin, for every $i = 1, \ldots, k$, and $k = \max(I)$. Then for any $i = 1, \cdots, k$, we have* $\max(I \setminus \{u_{li}, u_{mi}\}) = \max(I) - 1$.

Lemma 10. *Let $u_{l\alpha}$ be the largest large item for which there exists a medium item u_{mj} such that $w_{l\alpha} + w_{mj} \leqslant 1$, and let $u_{m\beta}$ be an arbitrary medium item satisfying $w_{l\alpha} + w_{m\beta} \leqslant 1$. Then* $\max(I \setminus \{u_{l\alpha}, u_{m\beta}\}) = \max(I) - 1$.

Proof. We construct a bipartite graph $G = (N, E) = (N_1 \cup N_2, E)$ to prove the result. Let $L(I)$ and $M(I)$ be the sets consisting of large, and medium items of I, respectively, then the bipartite graph can be obtained as follows: The node set $N = L(I) \cup M(I)$ satisfies $N_1 = L(I)$ and $N_2 = M(I)$, i.e., each large and medium item corresponds to a node of G, and there is an edge between $u_{li} \in L(I)$ and $u_{mj} \in M(I)$ if and only if $w_{li} + w_{mj} \leqslant 1$, i.e., these two items can fit commonly into a bin.

It is straightforward to know that $\max(I)$ is equal to the number of edges of a maximum matching (here the maximum matching means that its cardinality is maximum among all matchings). We only need to prove that there is a maximum matching which contains edge $(u_{l\alpha}, u_{m\beta})$. To see it, let $MM = \{(u_{l1}, u_{m1}), (u_{l2}, u_{m2}), \ldots, (u_{lk}, u_{mk})\}$ be a maximum matching, where $k = \max(I)$. Let $L = \{u_{l1}, u_{l2}, \ldots, u_{lk}\}$ and $M = \{u_{m1}, u_{m2}, \ldots, u_{mk}\}$. We consider four cases as follows:

Case 1. $u_{l\alpha} \notin L$ and $u_{m\beta} \notin M$. Then MM could be increased by adding edge $(u_{l\alpha}, u_{m\beta})$, a contradiction. Thus this case cannot occur.

Case 2. $u_{l\alpha} \notin L$ and $u_{m\beta} \in M$. Then $u_{m\beta} = u_{mi}$ for some $i \in \{1, \ldots, k\}$. Thus, simply replacing the edge (u_{li}, u_{mi}) by $(u_{l\alpha}, u_{mi}) = (u_{l\alpha}, u_{m\beta})$, we obtain another maximum matching, which contains edge $(u_{l\alpha}, u_{m\beta})$.

Case 3. $u_{l\alpha} \in L$ and $u_{m\beta} \notin M$. The statement can be shown in the same way as Case 2.

Case 4. $u_{l\alpha} \in L$ and $u_{m\beta} \in M$. If edge $(u_{l\alpha}, u_{m\beta}) \in MM$, then we are done. Thus, we assume that $(u_{l\alpha}, u_{m\beta}) \notin MM$. Let $u_{l\alpha} = u_{li}$ and $u_{m\beta} = u_{mj}$ for some $1 \leqslant i, j \leqslant k, i \neq j$. From the definition of $u_{l\alpha}$, it follows that $w_{l\alpha} \geqslant w_{lj}$. Thus, $w_{lj} + w_{mi} \leqslant w_{l\alpha} + w_{mi} = w_{li} + w_{mi} \leqslant 1$ holds, i.e., nodes u_{lj} and u_{mi} are adjacent in the graph. By replacing edges $(u_{l\alpha}, u_{mi})$ and $(u_{lj}, u_{m\beta})$ in MM by edges $(u_{l\alpha}, u_{m\beta})$ and (u_{lj}, u_{mi}), we obtain a new maximum matching with the desired property.

Therefore, we conclude that there is a maximum matching which contains edge $(u_{l\alpha}, u_{m\beta})$, and the lemma follows directly from Proposition 9. □

Applying Lemma 10, we can describe the pre-process procedure, which puts large and medium items commonly into bins as much as possible in a greedy way:

Procedure *Greedy*:

1. Let $M'_{1l} = M_{1l}$ and $M'_{1m} = M_{1m}$. Sort items in M'_{1l} in non-increasing order of their sizes.
2. If M'_{1l} or M'_{1m} is empty then go to 5.
3. Let u_i be the first item in M'_{1l}.
4. Let u_j be the first item in M'_{1m} satisfying $w_i + w_j \leqslant 1$. If it does not exist, then delete item u_i from set M'_{1l} and go to 2; Otherwise pack items u_i and u_j into a new bin, delete them from set M'_{1l} and M'_{1m}, respectively. Go to 2.
5. For all unpacked medium items, pack them pairwise into a bin. If the number of unpacked medium items is odd, then pack the last unpacked medium item alone into a bin.
6. Each unpacked large item with penalty at least 1 packed into its own bin.
7. Pack each huge item into its own bin.

Remark 11. Procedure *Greedy* does not decide whether and how to pack small items in M_{1s}, some large items with penalty less than 1, and items in M_2.

Corollary 12. *Let greedy*(I) *be the number of bins in which one large and one medium items are packed by Step 4 of procedure Greedy. Then greedy*$(I) = \max(I)$.

Proof. This assertion follows directly from Lemma 10 by induction on the number of items. □

Call the bins used in Steps 4-7 as B_1-, B_2-, B_3-, and B_4-bins, respectively. In the remainder of this subsection, we call a bin *open* if it is allowed to pack more items, otherwise it is *closed*. Now we can describe the algorithm.

Algorithm *RFF*4:

1. Apply procedure *Greedy*. Sort items in M_{1s} in non-increasing order of their sizes. Let all B_1-, B_2-, and B_4-bins be closed.
2. Arbitrarily choose an open B_3-bin if it exists, else go to 5.
3. Pack the first small item in M_{1s} into this open bin, if it fits into this bin, and delete it from M_{1s}, go to 3.
4. If the first small item in M_{1s} does not fit into this open bin, close this bin, and go to 2.
5. If there does not exist open B_3-bin, (and M_{1s} is not empty yet), and there is at least one unpacked large item in M_1, then pack the first unpacked large item into a new bin. Call this bin as an open B_5-bin. Go to 3.
6. If there are no unpacked small items, and there are unpacked large items, then reject them. Define by R_6 the set consisting of all rejected large items. Go to 8.
7. If all large items are packed, then pack all remaining small items in M_{1s} into new bins by algorithm *FF*. Let these bins denoted as B_7-bins. Go to 8.
8. Reject all items in M_2.

Before obtaining the asymptotic worst-case ratio of algorithm $RFF4$, we first estimate the content of used bins. It can be listed as follows:

(i) Every B_1-bin contains a large and a medium item, its content is more than 5/6.
(ii) Every B_2-bin except the last one contains two medium items, its content is more than 2/3.
(iii) Every B_4-bin contains a huge item, its content is more than 2/3.
(iv) Every closed B_3-, and B_5-bin contains one large and at least one small items. We show as follows that its content is more than 3/4. In fact, if one small item in one such bin has size at least 1/4, then the content of this bin is more than $1/2 + 1/4 = 3/4$. If all small items in this bin are less than 1/4, then let x be the small item which first does not fit into this bin, and causes the bin closed. Since small items are sorted in non-increasing order of their sizes, we know that $x < 1/4$, and again the content of the bin is more than 3/4.
(v) Every open B_3-, and B_5-bin contains one large and possibly some small items, its content is more than 1/2. Furthermore, there is at most one open B_5-bin. If Step 5 is executed at least once, then every B_3-bin is closed. Otherwise, if there is at least one opened B_3-bin, then no B_5-bin exists.

Theorem 13. $R^{\infty}_{RFF4} \leqslant \frac{3}{2}$.

Proof. It is trivial that the algorithm packs or rejects all items. We consider all possible situations which may occur. First suppose that there is at least one B_7-bin. Because the size of a small item is at most $\frac{1}{3}$, then we conclude that

(I) No open B_3- or B_5-bin exists. Hence the content of every used bin except B_7-bins and the last B_2-bin is at least $\frac{2}{3}$;

(II) No R_6-item exists.

From (I), we observe that the number of used bins except B_7-bins and the last B_2-bin is no more than $\frac{3}{2}$ times of the total sizes of all items packed in these bins. Note that Lemma 1 provides an upper bound of the number of B_7-bins. Hence we conclude that the number of all used bins by $RFF4$ is at most $2 + \frac{3}{2}\sum_{u_i \in M_1} w_i$. Therefore, we obtain

$$\begin{aligned} RFF4(I) &\leqslant 2 + \frac{3}{2}\sum_{u_i \in M_1} w_i + P(M_2) \\ &\leqslant 2 + \frac{3}{2}(\sum_{u_i \in M_1} w_i + P(M_2)) \leqslant 2 + \frac{3}{2}OPT(I), \end{aligned}$$

which is the desired asymptotic worst-case ratio. Thus, we suppose that no B_7-bin exists in the following.

Define by a, b, c, d, e, f the number of B_1-, B_2-, closed B_3-, open B_3-, B_4-, and closed B_5-bins respectively. Let $|R_6| = g$ and $P(M_2) = h$. Recall that there is at most one open B_5-bin. Clearly we have

$$\begin{aligned} RFF4(I) &\leqslant a + b + c + d + e + f + 1 + \sum_{u_i \in R_6} p_i + h \\ &\leqslant a + b + c + d + e + f + g + h + 1 \doteq \omega + 2, \end{aligned}$$

where $\omega = a+b+c+d+e+f+g+h-1$. Hence it suffices to show that

$$OPT(I) \geqslant \frac{2}{3}\omega$$

to obtain the desired asymptotical worst-case ratio. Four cases are considered as follows.

Case 1. Step 5 is never executed, and $d+g \leqslant a+2h$.

To obtain a tight lower bound of $OPT(I)$, we need a better estimate of the total size of all items in M_1 by analyzing the content of every used bin by $RFF4$. Recall that no B_5-bin exists now, thus $f=0$. By above (i)–(v), and the fact that every item of R_6 has a size at least 1/2, we have

$$\begin{aligned} OPT(I) &\geqslant \sum_{u_i \in M_1} w_i + P(M_2) \geqslant \frac{5}{6}a + \frac{2}{3}(b-1) + \frac{3}{4}c + \frac{1}{2}d + \frac{2}{3}e + \frac{1}{2}g + h \\ &\geqslant \frac{2}{3}(a+b+c+d+e+g+h-1) + \frac{1}{6}(a+2h-d-g) \geqslant \frac{2}{3}\omega. \end{aligned}$$

Case 2. Step 5 is executed at least once, and $2a+c+f+4h \geqslant 2g$.

Recall that at this moment every B_3-bin is closed and thus $d=0$. Similar to Case 1, we have

$$\begin{aligned} OPT(I) &\geqslant \sum_{u_i \in M_1} w_i + P(M_2) \geqslant \frac{5}{6}a + \frac{2}{3}(b-1) + \frac{3}{4}c + \frac{2}{3}e + \frac{3}{4}f + \frac{1}{2}g + h \\ &\geqslant \frac{2}{3}(a+b+c+f+g+h-1) + \frac{1}{12}(2a+c+f-2g+4h) \geqslant \frac{2}{3}\omega. \end{aligned}$$

For the remaining two cases, we obtain the desired result by contradiction. Suppose that there exists a counterexample which violates $RFF4(I) \leqslant \frac{3}{2}OPT(I)+2$, hence a *minimal counterexample I* with fewest items should exist. We first present two useful propositions. Their proofs are trivial and omitted here.

Proposition 14. *For any instance I, if item u_i is rejected in an optimal solution, then we have $OPT(I \setminus \{u_i\}) = OPT(I) - p_i$.*

Proposition 15. *For any instance I, if item $u_i \in R_6$, then we have $RFF4(I \setminus \{u_i\}) = RFF4(I) - p_i$.*

Therefore, we assume that we are dealing with the minimal counterexample I. Suppose $u_i \in R_6$. If it is also rejected in an optimal solution, then by Propositions 14–15, we obtain

$$\frac{RFF4(I \setminus \{u_i\}) - 2}{OPT(I \setminus \{u_i\})} = \frac{RFF4(I) - p_i - 2}{OPT(I) - p_i} > \frac{RFF4(I) - 2}{OPT(I)} > \frac{3}{2},$$

a contradiction. Therefore, we assume in the following that every item $u_i \in R_6$ is accepted in the optimal solution.

Case 3. Step 5 is never executed, and $d+g > a+2h$.

Consider the instance modified from I by deleting all small items, and denoted by I'. It is trivial that $OPT(I') \leqslant OPT(I)$. We next give a lower bound of $OPT(I')$ by the same method for proving Inequality (5) in the proof of Theorem 7. Each large item is either packed individually into its own bin or rejected. We have seen that every item $u_i \in R_6$ must be packed into different bins, their contribution to $OPT(I')$ is g. The large items being packed into B_3-bins must be also accepted in the optimal solution since their penalty are at least 1, hence their contribution is $c + d$. Noting that the penalty is no smaller than the size for each item in M_1, we conclude that the contribution of the large items in B_1-bins is at least $\frac{a}{2}$ no matter whether the items are accepted or rejected. Thus the total contribution of all large items to $OPT(I')$ is at least $\frac{a}{2} + c + d + g$. From Corollary 12 there are at most a bins, each of which can be shared by a large item and a medium item. Hence there are at least $2b - 1$ medium items which do not share bins with large items in the optimal solution. Their contribution is at least $\frac{2b-1}{3}$. It is also clear that for the huge items, their total contribution is at least $\frac{2}{3}e$ no matter whether they are packed individually into different bins, or rejected. Recall that $f = 0$. Thus applying $d + g > a + 2h$, we have

$$\begin{aligned} OPT(I) \geqslant OPT(I') &\geqslant \frac{1}{2}a + \frac{2b-1}{3} + c + d + \frac{2}{3}e + g \\ &\geqslant \frac{2}{3}(a + b + c + d + e + g + h - 1) + \frac{1}{6}(2c + 2d + 2g - a - 4h) \\ &\geqslant \frac{2}{3}(a + b + c + d + e + g + h - 1) + \frac{1}{6}(2c + a) \geqslant \frac{2}{3}\omega. \end{aligned}$$

Case 4. Step 5 is executed at least once, and $2a + c + f + 4h < 2g$.
Then $d = 0$. Similar to Case 3, we obtain

$$\begin{aligned} OPT(I) \geqslant OPT(I') &\geqslant \frac{1}{2}a + \frac{2b-1}{3} + c + \frac{2}{3}e + \frac{1}{2}f + g \\ &\geqslant \frac{2}{3}(a + b + d + e + f + g + h - 1) + \frac{1}{6}(-a - f + 2c + 2g - 4h) \geqslant \frac{2}{3}\omega. \end{aligned}$$

In summary, we have shown that $RFF4(I) \leqslant 2 + \frac{3}{2}OPT(I)$ for all possible cases, hence we obtain Theorem 13. $\square$

Whether there is an asymptotic PTAS for BPR is open.

3. The problem BCR

First we explain briefly the classic algorithms Dual Next Fit (DNF) and Dual Next Fit Decreasing (DNF). DNF always packs the next item into the first bin, which is not fully covered. In case of algorithm DNFD the items are first ordered into the non-increasing order of their sizes, and then we execute DNF.

3.1. On-line algorithms

Theorem 16. *No on-line algorithm can have an absolute competitive ratio of greater than* 0.

Proof. Let $w_1 = 1/2, p_1 = \varepsilon$. If an algorithm A packs this item into a bin, then no more item comes. The ratio is $A(I)/OPT(I) = 0/\varepsilon = 0$. If A rejects this item, then the second and last item with $w_2 = 1/2$ and $p_2 = \varepsilon$ comes. We have $A(I)/OPT(I) \leqslant 2\varepsilon/1 \to 0$ as $\varepsilon \to 0$, too. □

Hence in the remainder of this subsection, we consider the asymptotic competitive ratio of on-line algorithms. Since for the bin covering problem, no on-line algorithm can have an asymptotic competitive ratio of greater than 1/2 [8], this bound also holds for BCR. Next we present an on-line algorithm with matching asymptotic competitive ratio of 1/2.

Noting that *Dual Next Fit* (*DNF*) is the best possible on-line algorithm for the classical bin covering problem in terms of asymptotic competitive ratio [1], we next show that a simple modification of *DNF* also works for BCR. Recall that $M_1 = \{u_i | p_i > w_i\}$ and $M_2 = \{u_i | p_i \leqslant w_i\}$.

Algorithm *MDNF*:

1. If the incoming item is in M_2, pack it by *DNF* algorithm.
2. If the incoming item is in M_1, reject it.

The next lemma is useful for proving the main theorem:

Lemma 17. *Let I' be an instance of the classical bin covering with n' items. (1) ([1]) We have $DNF(I') \geqslant \left\lfloor \frac{\sum_{j=1}^{n'} w_j}{2} \right\rfloor$. (2) Furthermore, if $\sum_{j=1}^{n'} w_j \geqslant 2k+1$ for some integer k, then $DNF(I') \geqslant k+1$.*

Proof. Because $w_i \leqslant 1, i = 1, \ldots, n$, the content of every covered bin is less than 2 by *DNF*. Thus $DNF(I') \geqslant \left\lfloor \frac{\sum_{j=1}^{n'} w_j}{2} \right\rfloor$. If $\sum_{j=1}^{n'} w_j \geqslant 2k+1$ for some integer k, the number of covered bins is at least k. Furthermore, $w_i \leqslant 1, i = 1, \ldots, n$ implies that the total content of the first k bins is less than $2k$. Thus there is at least one more covered bin. □

Theorem 18. $R^{\infty}_{MDNF} = 1/2$, *thus MDNF is the best possible on-line algorithm in terms of asymptotic competitive ratio.*

Proof. With an argument analogous to the proof of Theorem 4, we can obtain

$$OPT(I) \leqslant \sum_{j \in M_2} w_j + \sum_{j \in M_1} p_j. \tag{7}$$

On the other hand, From Lemma 17 (1), the number of covered bins is at least $\left\lfloor \frac{\sum_{j \in M_2} w_j}{2} \right\rfloor$ by *MDNF*. Thus, the objective value produced by *MDNF* is $MDNF(I) \geqslant \left\lfloor \frac{\sum_{j \in M_2} w_j}{2} \right\rfloor + \sum_{j \in M_1} p_j$, and asymptotic competitive ratio follows. □

3.2. Off-line algorithms

Since the classical bin covering problem is a special case of BCR, we conclude that no off-line algorithm can have an absolute worst-case ratio of greater than 1/2, unless P=NP ([1]). Next we present a *modified Dual Next Fit Decreasing (MDNFD)* algorithm with an absolute worst-case ratio of 1/2.

Algorithm *MDNFD*:

1. If $\sum_{i=1}^{n} w_i < 1$, reject all items and stop.
2. If $W(M_2) < 1$ and $P(M_1) < 1$, pack all items into one bin and stop.
3. If $W(M_2) < 1$ and $P(M_1) \geqslant 1$, reject all items and stop.
4. Determine an integer $\omega \geqslant 0$ and real number $0 \leqslant \gamma < 2$ such that $W(M_2) = 2\omega + 1 + \gamma$.
5. If $\gamma \leqslant 1$ or $P(M_1) \geqslant 1$, then apply algorithm *MDNF* to all items and stop.
6. If $W(M_2 \cup M_1) \geqslant 2\omega + 3$, then accept all items, pack them by *DNF* algorithm, and stop.
7. Sort the items of M_2 in non-increasing order of their ratios between sizes and profits w_i/p_i such that $\frac{w_1}{p_1} \geqslant \frac{w_2}{p_2} \geqslant \cdots \geqslant \frac{w_{|M_2|}}{p_{|M_2|}}$. Determine

$$\bar{k} = \min\left\{ j : \sum_{i=1}^{j} w_i \geqslant 2\omega + 1, u_i \in M_2, i = 1, \ldots, j \right\}. \tag{8}$$

8. Pack the first $\bar{k}$ items of M_2 by *DNF*, reject the remaining items of M_2 and all items of M_1. Stop.

Theorem 19. $R_{MDNFD} = 1/2$, *thus MDNFD is the best possible in terms of absolute worst-case ratio.*

Proof. It is clear that *MDNFD* yields an optimal solution if the algorithm stops at Step 1. If *MDNFD* stops at Step 2, we have that $\sum_{i=1}^{n} w_i \geqslant 1$ and thus $MDNFD(I) \geqslant 1$. By (7), we have $OPT(I) \leqslant W(M_2) + P(M_1) \leqslant 2$ and thus $MDNFD(I) \geqslant \frac{1}{2}OPT(I)$. If *MDNFD* stops at Step 3, we have $MDNFD(I) \geqslant P(M_1) > \frac{1}{2}(W(M_2) + P(M_1)) \geqslant \frac{1}{2}OPT(I)$.

Hence we are left to consider the cases that the algorithm stops at Steps 5, 6 or 8. It follows that $W(M_2) \geqslant 1$. Hence there exist unique nonnegative integer ω and $0 \leqslant \gamma < 2$ such that $W(M_2) = 2\omega + 1 + \gamma$.

If *MDNFD* stops at Step 5, either $\gamma \leqslant 1$ or $P(M_1) \geqslant 1$ holds. If $\gamma \leqslant 1$, we have $2\omega + 1 \leqslant W(M_2) \leqslant 2\omega + 2$. By Lemma 17 (2) and the algorithm rule, we have

$$MDNFD(I) \geqslant \omega + 1 + P(M_1) \geqslant \frac{1}{2}(W(M_2) + P(M_1)) \geqslant \frac{1}{2}OPT(I).$$

If $\gamma > 1$ and $P(M_1) \geqslant 1$, we have $2\omega + 2 \leqslant W(M_2) < 2\omega + 3$, and thus

$$\begin{aligned} MDNFD(I) &\geqslant \omega + 1 + P(M_1) \geqslant \frac{1}{2}(W(M_2) - 1) + P(M_1) \\ &\geqslant \frac{1}{2}(W(M_2) + P(M_1)) \geqslant \frac{1}{2}OPT(I), \end{aligned}$$

where the third equality is from $P(M_1) \geqslant 1$, and the last one is from (7).

If *MDNFD* stops at Step 6, we have $1 < \gamma < 2$ and $P(M_1) < 1$. We thus obtain $OPT(I) \leqslant W(M_2) + P(M_1) < (2\omega + 3) + 1 = 2(\omega + 2)$. On the other hand, we have $MDNFD(I) \geqslant \omega + 2$ by Lemma 17 (2) and the algorithm rule. Therefore, $MDNFD(I) \geqslant \frac{1}{2}OPT(I)$.

Now we only need to consider the case that *MDNFD* stops at Step 8. We have $\gamma > 1, P(M_1) < 1$, and $W(M_1 \cup M_2) < 2\omega + 3$. $\gamma > 1$ implies $W(M_2) \geqslant 2\omega + 2$. To reach the goal, we need to obtain a better upper bound of the optimal value for this case. For this purpose we consider a linear programming relaxation of the problem.

It is clear that at most n bins can be covered in an optimal solution for the instance consisting of n items. Define

$$y_i = \begin{cases} 1 & \text{if the } i\text{th bin is used,} \\ 0 & \text{otherwise,} \end{cases} \qquad x_{ij} = \begin{cases} 1 & \text{if } u_j \text{ is put in the } i\text{th bin,} \\ 0 & \text{otherwise.} \end{cases}$$

Then BCR can be formulated as a linear program as follows:

$$\begin{aligned} \max \; & \sum_{i=1}^{n} y_i + \sum_{j=1}^{n} p_j \left(1 - \sum_{i=1}^{n} x_{ij}\right) \\ \text{s.t.} \; & \sum_{j=1}^{n} w_j x_{ij} \geqslant y_i, \quad i = 1, \ldots, n \\ & \sum_{i=1}^{n} x_{ij} \leqslant 1, \qquad j = 1, \ldots, n \\ & y_i = 0, 1, \qquad i = 1, \ldots, n \\ & x_{ij} = 0, 1, \qquad i, j = 1, \ldots, n \end{aligned}$$

Replacing $x_{ij} = 0, 1$ by $0 \leqslant x_{ij} \leqslant 1$ for every $i, j = 1, \ldots, n$, we obtain a relaxed problem: any item can be cut and spread into different bins, we call it *the problem BCR with preemption*. Let the optimal value of the instance I for the relaxed problem be $OPT_R(I)$. Then we have $OPT_R(I) \geqslant OPT(I)$ trivially.

Recall that all items in M_2 are sorted in non-increasing order of their ratios between sizes and profits, w_i/p_i, we define

$$l(k) = \min \left\{ j : \sum_{i=1}^{j} w_i \geqslant k, u_i \in M_2, i = 1, \ldots, j \right\}, \quad k = 1, \ldots, 2\omega + 2,$$

and

$$t(k) = \sum_{i=1}^{l(k)} w_i - k, \quad k = 1, \ldots, 2\omega + 2.$$

It is clear that $0 \leqslant t(k) < w_{l(k)}, k = 1, \ldots, 2\omega + 2$. For any $t(k) > 0, k = 1, \ldots, 2\omega + 2$, let $\alpha_1^k = t(k)/\ w_{l(k)}$ and $\alpha_2^k = 1 - \alpha_1^k$, and cut item $u_{l(k)}$ into items $u_{l(k)1}, u_{l(k)2}$ with $w_{l(k)i} = \alpha_i^k w_{l(k)}$ and $p_{l(k)i} = \alpha_i^k p_{l(k)}, i = 1, 2$. Let

$$M_2' = M_2 \cup \left(\bigcup_{k=1,\ldots,2\omega+2} \left\{ u_{l(k)1}, u_{l(k)2} \right\} \setminus \left\{ u_{l(k)} \right\} \right),$$

and partition M_2' into two subsets M_{21} and M_{22}, where

$$M_{21} = \{u_1, u_2, \ldots, u_{l(2\omega+2)-1}, u_{l(2\omega+2)1}\}$$

and

$$M_{22} = \{u_{l(2\omega+2)2}, u_{l(2\omega+2)+1}, u_{l(2\omega+2)+2}, \ldots, u_{|M_2|}\}.$$

Consider a feasible solution that packs all items in M_{21} by *DNF* algorithm and rejects all items in M_{22} and M_1. Hence we have

$$OPT_R(I) \geqslant 2\omega + 2 + P(M_{22}) + P(M_1). \tag{9}$$

(In fact, it is not hard to show that $OPT_R(I) = 2\omega + 2 + P(M_{22}) + P(M_1)$, but the weaker statement (9) is also enough for us to complete the proof). It is trivial that $P(\{u_{l(2\omega+2)}, u_{l(2\omega+2)+1}, u_{l(2\omega+2)+2}, \ldots, u_{|M_2|}\}) \geqslant P(M_{22})$, because almost the same items are in sets $\{u_{l(2\omega+2)}, u_{l(2\omega+2)+1}, u_{l(2\omega+2)+2}, \ldots, u_{|M_2|}\}$ and M_{22}, except that both $u_{l(2\omega+2)1}$ and $u_{l(2\omega+2)2}$ are in the first set while $u_{l(2\omega+2)1}$ is missing from the second set. Now we are ready to prove the last case of Theorem 19. Since $l(2\omega + 2) \geqslant \bar{k}$, we have

$$\begin{aligned} MDNFD(I) &\geqslant \omega + 1 + P\big(\{u_{l(2\omega+2)}, u_{l(2\omega+2)+1}, \ldots, u_{|M_2|}\}\big) + P(M_1) \\ &\geqslant \omega + 1 + P(M_{22}) + P(M_1) \geqslant \tfrac{1}{2} OPT_R(I) \geqslant \tfrac{1}{2} OPT(I) \end{aligned}$$

applying (9). This bound completes the proof. □

The following instance can show that the asymptotic worst-case ratio is still 1/2. Let I be an instance with $2N + 2$ items, each with size $1 - \varepsilon$ and profit $1 - 2\varepsilon$. Then we know $M_1 = \emptyset$. *MDNFD* stops at Step 5, and we have $MDNFD(I) = N + 1$ while $OPT(I) = (2N + 2)(1 - 2\varepsilon)$. We obtain that $A(I)/OPT(I) \to 1/2$ as $\varepsilon \to 0$.

To give an off-line algorithm with an asymptotic worst-case ratio of greater than 1/2 is open.

4. Final remarks

In this paper, we considered bin packing problems with rejection penalties and their dual problems. We studied their approximation algorithms by distinguishing on-line, off-line, and absolute competitive (worst-case) ratio and asymptotic competitive (worst-case) ratio. All results are listed in Table 1. Note that algorithms *FF*, *FFD*, and *DNFD* for the bin packing and covering problems run in time $O(n \log n)$, and *DNF* runs in time $O(n)$, our algorithms also run in time either $O(n \log n)$ or $O(n)$.

Our study suggests problems for further research. For example, could randomized algorithms perform better, on average, than our deterministic algorithms? Could we achieve better performance bounds in special cases, such as when every item has the same penalty p, or when the size of every item is less than a constant $a < 1$? It also would be interesting to study higher-dimensional bin packing problems with rejection penalties and vector packing problems with rejection penalties.

Acknowledgments

The authors thank three anonymous referees and editor Professor M.C. Loui for their valuable comments.

References

[1] S.B. Assman, D.S. Johnson, D.J. Kleitman, J.Y.T. Leung, On the dual of one-dimensional bin-packingproblem, J. Algorithms 5 (1984) 502–525..
[2] Y. Bartal, S. Leonardi, A. Marchetti-Spaccamela, J. Sgall, L. Stougie, Multiprocessor scheduling with rejection, SIAM J. Discrete Math. 13 (1996) 64–78.
[3] S. Basse, Computer Algorithms: Introduction to Design and Analysis, Addison-Wesley, Reading, MA, 1994.
[4] W. Chen, J. Yang, D. Lu, Y. Pan, Two mathematical models and algorithms of internet communications, Chin. J. Comput. 21 (1999) 51–55 (in Chinese).
[5] E.G. Coffman, M.R. Garey, D.S. Johnson, Approximation algorithms for bin packing: a survey, in: D. Hochbaum (Ed.), Approximation algorithms for NP-hard problems, PWS Publishing, 1997, pp. 46–93.
[6] J. Csirik, G. Woeginger, On-line packing and covering problems, in: A. Fiat, G. Woeginger (Eds.), Online Algorithms, Lecture Notes in Computer Science, vol. 1442, Springer, Berlin, 1998, pp. 147–177.
[7] J. Csirik, D.S. Johnson, C. Kenyon, Better approximation algorithms for bin covering, in: Proc. of SIAM Conference on Discrete Algorithms, Washington, DC, 2001, pp. 557–566.
[8] J. Csirik, V. Totik, On-line algorithms for a dual version of bin packing, Discrete Appl. Math. 21 (1988) 163–167.
[9] D.W. Engels, D.R. Kargar, S.G. Kolliopoulos, S. Sengupta, R.N. Uma, J. Wien, Techniques for scheduling with rejection, J. Algorithms 49 (2003) 175–191.
[10] L. Epstein, J. Noga, G.J. Woeginger, On-line scheduling of unit time jobs with rejection: minimizing the total completion time, Operat. Res. Lett. 30 (2002) 415–420.
[11] W. Fernandez de la Vega, G.S. Lueker, Bin packing can be solved within $1 + \varepsilon$ in linear time, Combinatorica 1 (1981) 349–355.
[12] M.R. Garey, D.S. Johnson, Computer and Intractability: A Guide to the Theory of NP-Completeness, Freeman, New York, 1979.
[13] Y. He, X. Min, On-line uniform machine scheduling with rejection, Computing 65 (2000) 1–12.
[14] Y. He, Gy. Dósa, Bin packing and covering problems with rejection, in: Lecture Notes in Computer Science, vol. 3595, 2005, pp. 885–894.
[15] H. Hoogeveen, M. Skutella, G.J. Woeginger, Preemptive scheduling with rejection, in: Proceeding of ESA'2000, Lecture Notes in Computer Science, vol. 1879, Springer, Berlin, 2000, pp. 268–277.
[16] K. Jansen, R. Solis-Oba, An asymptotic fully polynomial time approximation scheme for bin covering, Theor. Comput. Sci. 306 (2003) 543–551.
[17] D.S. Johnson, A. Demers, J.D. Ullman, M.R. Garey, R.L. Graham, Worst-case performance bounds for simple one-dimensional packing algorithms, SIAM J. Comput. 3 (1974) 299–325.
[18] N. Karmarkar, R.M. Karp, An efficient approximation scheme for the one-dimensional bin packing problem, in: Proc. 23rd Annual Symposium on Foundations of Computer Science, Chicago, 1982, pp. 312–320.
[19] M.B. Richey, Improved bounds for harmonic-based bin packing algorithms, Discrete Appl. Math. 34 (1991) 203–227.
[20] S.S. Seiden, Preemptive multiprocessor scheduling with rejection, Theor. Comput. Sci. 262 (2001) 437–458.
[21] S.S. Seiden, On the online bin packing problem, J. ACM 49 (2002) 640–671.
[22] S. Sengupta, Algorithms and approximation schemes for minimum lateness/tardiness with rejection, in: Proceeding of WADS'2003, Lecture Notes in Computer Science, vol. 2748, Springer, Berlin, 2000, pp. 79–90.
[23] D. Simchi-Levi, New worst-case results for the bin-packing problem, Naval Res. Logist. 41 (1994) 579–585.
[24] A. Van Vliet, An improved lower bound for on-line bin packing algorithms, Inf. Process. Lett. 43 (1992) 277–284.
[25] D. Ye, G. Zhang, On-line scheduling of parallel jobs, in: Lecture Notes in Computer Science, vol. 3104, Berlin, Springer, 2004, pp. 279–290.