ON OPEN-ENDS BIN PACKING PROBLEM

Siu-chung Lau[•], *Gilbert H. Young*[+], *W. K. Kan*[•] *and Yu-Liang Wu*[•]

Abstract

Classical bin packing has been studied extensively in the literature. Open-ends bin packing is a variant of the classical bin packing. Open-ends bin packing allows pieces to be partially beyond a bin, while the classical bin packing requires all pieces to be completely inside a bin. We investigate the open-ends bin packing problem for both the off-line and on-line versions and give algorithms to solve the problem for parametric cases.

1 Introduction

The classical one-dimensional bin packing problem has been studied extensively in the past two decades. Suppose we are given a list $L = (a_1, a_2, \ldots, a_n)$ of items, each with a size $s(a_i) \in (0,1)$ with $i = 1, \ldots, n$ and the goal is to find a packing of these items into a minimum number of unit-capacity bins. This problem is a basic problem in Theoretical Computer Science and Combinatorial Optimization. It has many potential real-world applications in paged computer systems, in packet routing in communication networks, in assigning advertisments to station breaks on television, in cutting-stock problems etc. Since the problem of finding an optimal packing is NP-hard [3], many research on finding near-optimal approximation algorithms have been proposed. In the past twenty-five years, many interesting results were obtained in this area [2].

In the one-dimensional covering problem, the goal is to pack a list L of items into a maximum number of bins of size one such that the sum of the sizes in any one of the bins is at least one. This means that we have to fill as many bins as possible. The problem was investigated for the first time by Assmann et al. [1] who showed that the problem is NP-hard. The open-ends bin packing problem may be considered as a kind of inverse or dual version of the one-dimensional covering problem.

The open-end bin packing problem which only allows one end open for a bin was first considered by Young [6]. Some experimental results are shown. This problem arises in the fare payment system in the subway stations in Hong Kong [6]. Leung et al. [5] showed that the open-end bin packing problem is strongly NP-hard and that all on-line algorithms must have an asymptotic worst-case ratio at least two which can be easily attained by a simple on-line algorithm.

[•] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong. Email: {schlau,wkkan,ylw}@cse.cuhk.edu.hk, Tel.: 852-2609-8440, Fax: 852-2603-5032
[+] Department of Computing, The Hong Kong Polytechnic University, Hong Kong.

JCMCC 44 (2003), pp. 3-9

Open-end bin packing in two and three dimensions are considered in [8,9] which only allows one end opens for bins and some experimental results and algorithmic analysis were also given. The number of possible cases for the open ends in high dimension of a bin were considered in [7].

A bin packing (covering) problem is called on-line if the items are not known in advance but arrive one by one and must be assigned to bins as soon as they arrive. Once an item is packed, it is not allowed to be moved again. Therefore an on-line bin packing (covering) algorithm always packs all items solely on the basis of the sizes $s(a_j)$ of the items a_j, $1 \le j \le n$, and without knowing any information on the subsequent items. The decisions of the on-line algorithm are irrevocable.

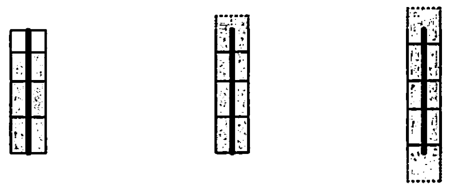
For a given list L of items and an approximation algorithm A, let $A(L)$ denote the number of bins used by algorithm A and $OPT(L)$ denote the number of bins in the optimal packing (off-line). Then the asymptotic worst-case performance ratio, denoted by R_A^∞, of algorithm A is defined to be the ratio between $A(L)$ and $OPT(L)$ as $OPT(L)$ tends to infinity. If in addition one restricts lists to those for which all items have sizes at most α, one can analogously define the parametric worst-case performance ratio $R_A^\infty(\alpha)$. Note that $R_A^\infty[1] = R_A^\infty$.

In this paper, we consider a variant of the classical bin packing problem, which we call the open-ends bin packing problem. Like the classical bin packing problem, we are also given a list $L = (a_1, a_2, \ldots, a_n)$ of n pieces and each item with a size $s(a_i) \in (0,1]$ and our goal is to pack the pieces into a minimum number of unit-capacity bins. However, unlike the classical bin packing problem, a bin can be filled with pieces packed beyond its bin size as long as at least a fraction of each piece is inside the bin. In other words, there are only two cases for items, one case is that the item is totally inside the bin and the other case is part of the item is inside the bin and part of the item is outside the bin. The latter case can occur at the ends of a bin. Different from the open-end bin packing problem in [5,6] which allows only one end of each bin open, we consider packings that both ends of bins can be opened. Examples of the classical, open-end and open-ends (open-2-end) one-dimensional bin packing problems are shown in Figure 1. Some off-line algorithms are considered first and then we give an on-line algorithm which has the competitive ratio $\frac{(m+4)}{(m+2)}$ for the parametric constraint $s(a_i) \le \frac{1}{m}$.

2 Offline algorithms

The offline open-ends bin packing problem has a close relationship with the offline bin packing problem. We will show these connections with the following Lemmas and Theorems.

For a given list $L = (p_1, p_2, \ldots, p_n)$, let $Opt_s(L)$ denote the optimal solution for the bin packing of L and $Opt(L)$ denote the optimal solution for the open-ends bin packing of L. Throughout the proofs of the following results, we assume that there is no sublist of L whose sizes sum up to one.



Classical Open-end Open-ends

Figure 1. Examples of 1D bin packing

Lemma 1 Let $Opt(L) = m$ and $p_i \le p_{i+1}$, $1 \le i \le n-1$ in L, then the optimal solution for the bin packing of $L_k = (p_1, p_2, \ldots, p_k)$ satisfies $m - 2 \le Opt_s(L_k) \le m$, where $k = n - 2m$.

Proof First we show that $Opt_s(L_k) \le m$. Since $Opt(L) = m$, then for each bin of the optimal open-ends bin packing, pick up the first two largest items from the bin, then the sum of the rest left in the bin is less than 1. Let p^* be the smallest item of the items picked up from the bins. If there is an item which is inside some bin and larger than p^*, then we can exchange p^* with that item until the smallest item outside the bins is equal or larger than the largest item inside the bins. So we have $Opt_s(L_k) \le m$. Next, we show that $Opt_s(L_k) \ge m - 2$. Suppose that $Opt_s(L_k) \le m - 3$. Since each bin of the optimal bin packing of L_k can accept two large items for the open-ends bin packing and an empty bin can accept at least three large items, then the number of bins used for the open-ends bin packing of L is at most

$$Opt_s(L_k) + \frac{2m - 2Opt_s(L_k)}{3}$$

As $Opt_s(L_k) + \frac{2m - 2Opt_s(L_k)}{3} \le m - 1$, it is contradictory to our assumption that $Opt(L) = m$. This ends the proof.

Lemma 2 If there is an α-approximation offline algorithm A_s for the bin packing with time complexity $O(f(n))$, then there is an offline α-approximation algorithm A for the open-ends bin packing with time complexity $O(f(n) \log n)$.

Proof Suppose algorithm A is an α-approximation offline algorithm for the optimal bin packing of a list $L = (p_1, p_2, \ldots, p_n)$ with time complexity $O(f(n))$. We construct an algorithm A^* which produces an α-approximation for the optimal open-ends bin packing of a list with time complexity $O(f(n)\log n)$ as follows.

For a given list L, we first sort the list in an increasing order as $L = (p_1, p_2, \ldots, p_n)$, i.e. $p_i \le p_{i+1}$, for $1 \le i \le n-1$. This will cost $O(n \log n)$ time.

Algorithm A^*

Step 0. Initialize $LB = \left\lceil \frac{1}{3}\sum_{i=1}^{n} p_i \right\rceil$ and $UB = \left\lceil \sum_{i=1}^{n} p_i \right\rceil$.

Step 1. Use A for the list $L_k = (p_1, p_2, \ldots, p_k)$, for $M = \left\lfloor \frac{LB+UB}{2} \right\rfloor$ and $k = n - 2M$.

Step 2. If 1) $A(L_k) < M$ then set $UB = M$ and return Step 1.
2) $A(L_k) > M$ then set $LB = M$ and return Step 1.
3) $A(L_k) = M$ then $A^*(L) = A(L_k)$.

Since there are at most $O(\log \left\lceil \frac{2}{3}\sum_{i=1}^{n} p_i \right\rceil)$ (i.e., $O(\log n)$ as $0 < p_i \le 1$ for all $1 \le i \le n$) loops in the algorithm A^*, we know that the time complexity of A^* is $O(f(n)\log n)$. Let $k^* = n - 2 \times A^*(L)$. Since $A^*(L) \ge Opt(L)$, $k^* \le n - 2 \times Opt(L)$. From Lemma 1, we know that $Opt_s(L_k^*) \le Opt(L)$. Since $A(L_k^*)$ is at most α times $Opt_s(L_k^*)$, we have that $A^*(L)$ is at most α times $Opt(L)$. This ends the proof.

Combining the above two lemmas and using fully polynomial approximation schema for the classical bin packing problem, e.g., Karmarkar and Karp's algorithm [4], as scheme A, we have the following theorem.

Theorem 1 There is a fully polynomial approximation scheme for the one-dimensional open-ends bin packing problem.

3 Online algorithms

From the above section, we know that the offline open-ends bin packing problem has a fully polynomial approximation scheme. In this section we consider algorithms for the online version of open-ends bin packing problem which must pack the pieces in the order they arrive, and once a piece is packed, it can not be repacked again. Probably the most natural online algorithm for the open-ends bin packing problem is the *Next Fit* (*NF* for short) which was proposed in [5,6]. This algorithm operates by keeping only one open bin and the pieces are packed in the order they arrive. Unlike the case for the one end open bin problem considered in [5,6], we modify the *NF* algorithm for the open-ends bin packing as *Central First Next Fit* (*CFNF* for short) algorithm which packs the first piece at the center of the bin if the bin is empty and then use *NF* for the following pieces at the two sides of the bin. When the open bin is filled at its two ends, the bin will be closed and a new open bin is started. Clearly, The *CFNF* has a linear time implementation. Furthermore, it has an asymptotic worst-case bound of $\frac{m+2}{m}$, if all pieces are bounded by $\frac{1}{m}$.

An algorithm for the open-ends bin packing is called a full packing algorithm if the output has only one or two bins which may have some empty space and the others are all fully packed.

Lemma 3 Let A be a full packing algorithm and M^* be the optimal open-ends bin packing for $L = (p_1, p_2, \ldots, p_n)$ and $p_i \le p_{i+1}$ where $1 \le i < n$, $\alpha = \frac{\sum_{n-2M^*<i\le n} p_i}{M^*}$, then A is an $(1 + \alpha)$-approximation of the optimal packing.

Proof Let $\sum_{n-2M<i\le n} p_i = t$ and $A_f(L)$ be the output of a full packing algorithm. Since the maximum size of items which are totally inside a bin must be smaller than 1, we have $M^* + t = M^* + \sum_{n-2M<i\le n} p_i \ge \sum_{1<i\le n} p_i$. As the level of a bin must reach one before it is closed, $\sum_{1<i\le n} p_i \ge A_f(L)$. Hence,

$\frac{A_f(L)}{M^*} \le \frac{M^* + t}{M^*} = 1 + \alpha$ This ends the proof.

For the open-ends bin packing, and for an empty bin, one can pack the first item any where if part of the item is inside the bin. To make things easy, we will mainly consider the *CFNF*. Since *CFNF* is a full packing algorithm, we have the following theorem.

Theorem 2 If the size of every item is less than $\frac{1}{m}$, then *CFNF* has the approximation ratio $\frac{m+2}{m}$.

Proof Since the item size is bounded by $\frac{1}{m}$, then for the optimal open-ends bin packing, each bin has at most $\frac{2}{m}$ outside the bin, i.e., this is just the case for $\alpha = \frac{2}{m}$. From Lemma 3, the theorem holds.

Assume that all items are less than $\frac{1}{m}$. An p_i item is called a *large* item if its size is greater than $\frac{1}{m+2}$; otherwise it is called a *small* item. With this classification, we can give a better online algorithm which is called *Harmonic Algorithm* as follows.

Harmonic Algorithm (*HA* for short): If *m* is odd, then apply the *Central First Next Fit* algorithm to small items and large items, respectively. If *m* is even, then apply the *CFNF* to small items and pack the first large item for an empty bin beyond the center of the bin and then use *NF* for the rest of the large items. In other words, we have two bin sets, one of which is for packing small items by *CFNF* and another is for packing large items by *CFNF*, if *m* is odd and packing first item for an empty bin beyond the center of the bin and then use *NF* for the rest, if *m* is even.

Theorem 3 The *HA* has the approximation ratio $\frac{m+4}{m+2}$.

Proof For any list $L = (p_1, p_2, \ldots, p_n)$, assume that L contains k large items. Let a_s denote the sum of the small items. For *HA* packing, there are only two types of bins. Let S_1 denote the set of bins containing only large items and let S_2 be the set of bins containing only small items. It is clear that every bin in S_1, except the last one, must accept at least $m + 2$ large items. The number of bins in S_1 is no larger than $\frac{k}{m+2}+1$. Every bin in S_2, except the last one, must be fully packed. So the total number of bins in S_2 is not greater than $a_s + 1$. Therefore the following inequality holds;

$$HA(L) \le \frac{k}{m+2} + a_s + 2$$

Let t be the number of bins used in an optimal open-ends bin packing of L.

i) if $k \ge 2t$, , let $k = 2t + l$. Then there are at least l items which are totally inside the bins of the optimal packing of L. Since the size of a large item is at least $\frac{1}{m+2}$, we have that $a_s + \frac{l}{m+2} \le t$ and then

$$HA(L) \le a_s + \frac{2t+l}{m+2} + 2 \le t + \frac{2t}{m+2} + 2.$$

ii) if $k \le 2t$, let $k = 2t - l$. Since the size of every item in S_2 is smaller than $\frac{1}{m+2}$ and there are at most $2t$ items which are partly outside the bins in optimal packing of L, then we have $a_s - \frac{l}{m+2} \le t$ and then $HA(L) \le a_s + \frac{2t-l}{m+2} + 2 \le t + \frac{2t}{m+2} + 2$.
Both i) and ii) imply the theorem holds.

4 Concluding Remarks

In this article, we proposed the open-ends bin packing problem, which is a new variant of the classical bin packing problem and also a variant of the one end open bin packing problem observed in [6]. Offline and online algorithms are given for the open-ends bin packing problem in this paper.

As for some further research problems, whether the approximation ratio $\frac{m+4}{m+2}$ of *HA* for the online open-ends bin packing problem can be improved is an interesting topic. Moreover, it remains open to develop lower bounds that hold with respect to optimal offline algorithms that must pack a list of items in the order of arrival.

Reference

[1] Assmann S. J., Johnson D. S., Kleitman D. J., Leung J. Y.-T., "On a dual version of the one-dimensional bin packing problem," J. Algorithms 5, 502-525(1984).

[2] Coffman E. G., Garey M. R., Johnson D. S., "Approximation algorithms for bin packing: A survey," Approximation Algorithms for NP-hard Problems (Hochbaum D. Ed.), pp.46-93, Boston:PWS publishing, 1996.

[3] Garey M. R., Johnson D. S., "Computers and intractability: a guide to the theory of NP-completeness," San Francisco: W. H. Freeman, 1979.

[4] Karmarkar N., Karp R. M., "An efficient approximation scheme for the one-dimensional bin packing problem," Proc. 23rd Annual Symposium on Foundations of Computer Science, IEEE Computer Society, Long Beach, CA, 1982, pp.312-320.

[5] Leung J., Dror M. and Young G. H., "A note on an open-end bin packing problem," Journal of Scheduling, to appear.

[6] Young G. H., "On-line algorithms for open-ended bin-packing and a case study," Proc. Second International Symposium on Operations Research and its Applications, ISORS'96, Guilin, China, Dec 96, pp.225-230.

[7] Young G. H. and Lau S., "Foundation of open-end bin packing," unpublished manuscript.

[8] Young G. H., Shen H., Li K., Pan Y. and Zheng S., "On 2-dimensional open-end bin packing," Prof. Third International Symposium on Operations Research and its Applications, ISORA'98, Kunming, China, Aug.98, pp.75-83.

[9] Young G. H. and Li K., "On the 3-dimensional open-end bin packing," Proc. Third International Symposium on Operations Research and its Applications, ISORA'98, Kunming, China, Aug 98, pp.96-107.