Proceedings of the 8^{th} International Conference on Applied Informatics
Eger, Hungary, January 27–30, 2010. Vol. 2. pp. 191–197.

On a Multidimensional Semi-on-line Bin Packing Problem*

János Balogh[a], József Békési[a], Gábor Galambos[a], Gerhard Reinelt[b]

[a]Department of Informatics' Applications, Gyula Juhász Faculty of Education, University of Szeged

[b]Institute of Computer Science, University of Heidelberg

Abstract

In the paper we present the first lower bounds for the asymptotic competitive ratio of multidimensional semi-on-line bin packing. We give the lower bound 1.3871... for the case of multidimensional on-line bin packing problem where repacking is allowed with the arriving of a new element, but the number of such repackable elements are bounded by a finite constant per a new arriving element. The result is valid in every d-dimension. Our results improve the lower bound of 4/3 by Coppersmith and Raghavan given in [4] for the on-line hypercube packing problem.

Keywords: semi-on-line bin packing problems, multidimensional bin packing, cube packing, worst case analysis

MSC: 68Q25, 68W25, 68W27

1. Introduction

In the classical one–dimensional bin-packing problem we are given a list $L = \{x_1, x_2, \ldots, x_n\}$ of n elements with size s_i, where $0 < s_i \leq 1$. We need to pack the items into a minimum number of unit-capacity bins so that the sum of the sizes in each bin does not exceed 1. It is well-known that finding an optimal packing is *NP-hard* [8]. Consequently, large number of research have been published which look for polynomial time algorithms with an acceptable approximative behavior. The algorithms have been classified into different classes: The *on-line bin packing algorithms* put items into bins as they appear without knowing anything

*The research was supported by the Hungarian National Research Fund (projects T 048377 and T 046822) an by the MÖB-DAAD Hungarian-German Researcher Exchange Program (project No. 21).

about the subsequent elements (neither the sizes nor the number of the elements). *Off-line algorithms* can use more information: most of them examine the entire list before they apply their strategy to pack the items. The so called *semi-on-line algorithms* [3] are between the on-line and off-line ones. For such algorithms at least one of the following operations is allowed: repacking of some items [5, 6, 7, 11], lookahead of the next several elements [9], or some kind of preordering. If the lookahead is valid for certain sublists – without any constraint on the number of elements – then we speak about *batched algorithm.* In case of *dynamic bin packing algorithms* in each step not only the insertion of the arrived elements is allowed, but in any step one element can be deleted as well. We speak about *fully dynamic bin packing algorithm* if repacking is also allowed.

The efficiency of different algorithms is generally measured by two different methods: the investigation of the worst-case behavior, or – assuming some probability distribution of the elements – a probability analysis. In this paper we will concentrate on the asymptotic competitive ratio which can be defined as follows: denote $A(L)$ the number of bins used by the algorithm A while packing the elements of a list L, and let L^* the number of bins in an optimal packing. If

$$R_A(k) := \max\left\{\frac{A(L)}{k} | L^* = k\right\} \tag{1.1}$$

denotes the maximum ratio of $A(L)/L^*$ for any list L with $L^* = k$, then the *asymptotic performance ratio (APR)* R_A of the algorithm A is defined as

$$R_A := \limsup_{k\to\infty} R_A(k). \tag{1.2}$$

The best known lower bound for the APR of any on-line bin packing algorithm A is 1.54014 (given by Van Vliet [14]), while for the current best algorithm has an APR of 1.58889 (Seiden, [13]).

Semi-on-line bin packing algorithms have been studied by Gambosi et al. [7]. While having packed the elements they have not restricted the number of repackable elements by a finite constant. They gave an $O(n)$ time algorithm with an APR of 1.5 and an $O(n \log n)$ time algorithm with an APR of $\frac{4}{3}$. The latter algorithm was improved by Ivkovič and Lloyd [12] who constructed an algorithm with APR of $\frac{5}{4}$.

The first lower bound for this problem is proved by Ivkovič and Lloyd. This bound is $\frac{4}{3}$. In their lower bound construction the repacking of a constant number of items is allowed after the arrival of each a new item. The same $\frac{4}{3}$ lower bound has been proved for fully dynamic bin packing with restricted repacking. Although the $\frac{4}{3}$ lower bound was constructed for fully dynamic bin packing algorithms, the model can be easily applied for those of semi-on-line bin-packing algorithms where repacking is allowed.

Batched bin packing algorithms were studied by Gutin et al. [10]. They investigated deeply only the case when we have only two batches, and they gave a lower bound 1.3871 ... for the batched algorithms for this case.

In this paper we give the first lower bound for that of the variant of the multidimensional on-line bin packing problem when in each step the repacking of constant number of elements is allowed. We focus on the multidimensional case of this problem, but the same construction can be used for deriving the same lower bound for the similar version of the fully dynamic bin packing problem

Our bound is valid for the classical on-line hypercube packing problem, in any d-dimension. Furthermore, our lower bound improves the lower bound of 4/3 given by Coppersimth and Raghavan [4]. For the cases $d \geq 4$ it was the best known lower bound for the problem.

2. Preliminaries: Constructing the lower bounds by a linear and a nonlinear optimization problem

The following theorem is proved in [1]:

Theorem 2.1. *[1] Let $k \geq 1$ and $c \geq 1$ be arbitrary integers and $x_1, x_2, \ldots, x_k$, $(\frac{1}{2} \leq x_1 < x_2 < \cdots < x_k < 1)$ be fixed real numbers. Let $y_i = 1 - x_i$, $(i = 1, \ldots, k)$ and $y_{k+1} = 0$. Then the solution of the following linear programming problem is a lower bound for the APR of an arbitrary semi-on-line bin packing algorithm with c-repacking:*

$$\min \ b, \tag{2.1}$$

subject to

$$b \geq 1 + y_i + 2y_i \left(\sum_{j=1}^{i-1} z_j \left(\frac{1}{y_j} - 1 \right) - \sum_{j=i}^{k} z_j \right), \ (i = 1, \ldots, k), \tag{2.2}$$

$$b \geq 1 + \sum_{j=1}^{k} 2z_j \left(\frac{1}{y_j} - 1 \right), \tag{2.3}$$

$$z_i \geq 0, \ (i = 1, \ldots, k), \tag{2.4}$$

$$\sum_{j=1}^{k} z_j < \frac{1}{2}. \tag{2.5}$$

Note that the expression c-repacking means that a given number c of elements can be repacked in each step.

Although we omit the proof of this theorem, we introduce a list-construction as the basic idea behind the proof: consider a series of lists $L_1, ..., L_k$, where L_j $(j = 1, \ldots, k)$ contains $\left\lceil \frac{n}{2y_j} \right\rceil$ items of size $x_j + \varepsilon_j$, where $\varepsilon_j := \frac{\varepsilon}{\left\lceil \frac{n}{2y_j} \right\rceil}$, and $\varepsilon < \min_{j=1,\ldots,k} \{y_j - y_{j+1}\}$ is an arbitrary positive number. L_0 is defined as a list of M items of size a, where $a < \frac{\varepsilon_k}{\left\lceil \frac{n}{y_k} \right\rceil c}$ and $M := \left\lfloor \frac{\frac{n}{2} - \varepsilon}{a} \right\rfloor$. It can be seen, that $size(L_j)$, i.e. the sum of the elements in L_j is $\frac{n}{2} + \varepsilon$, while $size(L_0)$ is $\frac{n}{2} - \varepsilon$.

For an arbitrary list L_j, $0 \leq j \leq k$, we denote the sum of the sizes of all elements in L_j by $size(L_j)$. It is easy to prove that $size(L_0) \leq \frac{n}{2} - \varepsilon$. From the definition of the lists L_j, it follows that — while packing their elements — each element is placed into separate bin, and the sum of the free space in the bins is at least $\frac{n}{2} - \varepsilon$.

The key idea of the construction is the following: if a is so small, then considering the list-concatenations $L_0L_1, L_0L_2, \ldots, L_0L_k$, the total size of the repackable small elements during the packing of the second list L_j (in $\left\lceil \frac{n}{2y_j} \right\rceil$ steps) is less than ε_j $(j = 1.\ldots, k)$.

This way we "almost switch off" the role of the repacking. In the sequel the $level(B)$ denotes the cumulative size of the items have been packed into B. It is directly follows from the size of any "big" element, that such a big element can be packed only in a bin, which level is at most $y_j - \varepsilon_j$. If $z_i n$ denotes the cumulative size of the items that are packed in y_i-type bins, – we call a bin B y_i-type bin if $size(B) \in (y_{i+1}, y_i]$ – then because of the above reasoning a bin containing a big element had level of at most y_j after the packing of L_0.

We can now estimate (1.2) for L_0 and for L_0L_j, $j = 1, \ldots, k$. Namely, inequality (2.2) of Theorem 2.1 comes from the estimation of (1.2) for the list concatenations $L_0L_j, \forall j$, while inequality (2.3) comes from the estimation of (1.2) for the case list L_0 (i.e. for the case when there is no list appearing after L_0).

In [2] it was shown that the solution of (3) subject to (4)-(7) is a lower bound not only for the c-repacking semi-on-line bin packing problem, but the c-repacking fully dynamic bin packing problem and for the 2-batched bin packing problem as well (obtaining the same lower bound).

3. Multidimensional lower bound

The next statement shows that the optimum of (2.1)–(2.5) is valid in any dimension.

Lemma 3.1. *Let c be an arbitrary positive integer and $0 < y_k < \ldots < y_1 \leq \frac{1}{2}$ real numbers. The optimum of (2.1)–(2.5) is a lower bound for all c-repacking SOL, c-repacking FDP, and 2-BBP problems in any d-dimension ($d \geq 1$, integer).*

Proof. We generalize the 1-dimensional construction to a d-dimensional *cube packing* problem.

Firstly, we will choose the values ε and ε_j $(j = 1, \ldots, k)$ as above. For a the inequalities $a < \frac{\varepsilon_k}{\left\lceil \frac{n}{y_k} \right\rceil c}$ and $a \leq \left(\frac{\varepsilon}{\left\lceil \frac{n}{2y_k} \right\rceil d\cdot(2^d-1)}\right)^d$ must hold. Since we will construct d-dimensional cubes with matching volumes, so the lengths of the sides of the cubes in the lists $L_0, L_1, \ldots, L_k$ are considered as $\sqrt[d]{a}$ and $\sqrt[d]{x_1 + \varepsilon_1}, \ldots, \sqrt[d]{x_k + \varepsilon_k}$, respectively. Here $x_i = 1 - y_i$, $i = 1 \ldots, k$. The variables z_j $(j = 1, \ldots, k)$ correspond to the occupied volumes in the d-dimensional unit cube after having packed the elements of L_0. Following the train of thought has been used in proof of the one-dimensional case in [1], we show that the d-dimensional modification of the construction leads to the same problem defined in (2.1)–(2.5).

It is easy to see that the lower bounds given in [1] for $A(L_0)$ and $A(L_0L_i)$, $i \in \{1,\dots,k\}$, remain valid for the d-dimensional construction as well. (Notice that for case one needs to correspond the cumulative volume of the contained elements to the level of a bin.) So, we get

$$A\left(L_0\right) \geq \sum_{j=1}^{k}\frac{z_j n}{y_j} + Ma - n\sum_{j=1}^{k} z_j = Ma + n\sum_{j=1}^{k} z_j\left(\frac{1}{y_j} - 1\right), \tag{3.1}$$

and

$$A\left(L_0L_i\right) \geq \frac{n}{2y_i} + n\sum_{j=1}^{i-1}\frac{z_j}{y_j} + Ma - n\sum_{j=1}^{k} z_j, \ \ (i = 1,\dots,k). \tag{3.2}$$

Now, we will show that those upper bounds what we proved for OPT(L_0) and OPT(L_0L_i) in the one-dimensional case are also true for the d-dimensional construction.

First, we will estimate OPT(L_0). We remind the reader that the list L_0 contains very small elements with equal sizes. (We call them as *small-elements.*) To produce a feasible packing for these type of elements it is enough to use a simple level oriented next-fit type strategy (we denote it by LNF): we pack the small elements level by level into bins packing at each level so many elements as it possible. If there is no place in the actual bin for new element we declare it full and open a new, empty bin. In the full bins the packed elements form a d-dimensional cube with side $1 - \sqrt[d]{a}$. So, the wasted space in each bin – except at most the last one – is $d \cdot \sqrt[d]{a}$, and so

$$\mathrm{OPT}\left(L_0\right) \leq \frac{Ma}{1-a'} + 1 \leq \frac{\frac{n}{2} - \varepsilon}{1-a'} + 1 \leq \frac{\frac{n}{2}}{1-a'} + 1. \tag{3.3}$$

Inequalities (3.1) and (3.3) yield the condition (2.3) here, as well.

To get an upper bound for OPT(L_0L_i) we will create a feasible packing. First we pack the elements of L_i, and thereafter the elements of L_0. In both cases we apply the next-fit strategy, and we call this algorithm as *Double Next-Fit* (DNF).

Step 1. Pack the elements of L_i into bins, each in one. (We call these bins as *big-bins.*)

Step 2. Using the LNF rule fill up the empty spaces in big-bins, bin by bin (and level by level) with the elements of L_0. If the algorithm run out from the big-bins, open a new, empty bin, and packs the remained small elements by the LNF rule.

We remark that in the last step while the algorithm packs small-elements in the actual bin it puts them not only on the top of the big-element but it fill up the empty spaces at its sides as well. Let us denote a bin by B_i if it contains element from L_i. It is easy to check then after *Step 1* a B_i bin has empty volume $y_i - \varepsilon_i$, while the empty volume after the *Step 2* is $a" = d \cdot (2^d - 1)\sqrt[d]{a}$.

Let $V_s(B_i)$ denote the cumulative volume of small-elements in B_i after *Step 2*. Then

$$V_s(B_i) \geq \left\lceil \frac{n}{2y_i} \right\rceil (y_i - \varepsilon_i - a") \geq \frac{n}{2} - \left\lceil \frac{n}{2y_i} \right\rceil \varepsilon_i - \left\lceil \frac{n}{2y_i} \right\rceil a" \geq Ma - 2\varepsilon.$$

If $N(B_s)$ denotes the number of those bins which contain only small-elements then it follows that $N(B_s) \leq 1$. So, we can give an estimation for the minimum number of bins:

$$\mathrm{OPT}\,(L_0 L_i) \leq \left\lceil \frac{n}{2y_i} \right\rceil + N(B_s) \leq \frac{n}{2y_i} + N(B_s) + 1 \leq \frac{n}{2y_i} + 2 \tag{3.4}$$

If we combine (3.2) and (3.4) then we get the inequalities (2.2), and this yields the desired result. □

We note that the statement of Theorem 2 is valid for the c-repacking fully dynamic bin packing problem and for the 2-batched bin packing problem as well.

4. Summary

The paper gave bounds for multidimensional semi-on-line bin packing problems. The lower bound construction is given by hypercubes, and so valid for the classical d-dimensional on-line hypercube packing problem, even if repacking is not allowed. The bound improves the lower bound of 4/3 ([4]) which was the best known lower bound for cases $d \geq 4$.

References

[1] Balogh, J., Békési, J., Galambos, G., Reinelt, G., Lower Bound for Bin Packing Problem with Restricted Repacking, *SIAM Journal on Computing* Vol. 38 (2008), 398–410.

[2] Balogh, J., Békési, J., Galambos, G., Markót, M.Cs., Improved lower bounds for semi-on-line bin packing problems, *Computing* Vol. 84 (2009), 139–148.

[3] Coffman, E.G., Galambos, G., Martello, S., Vigo, D., Bin Packing Approximation Algorithms: Combinatorial Analysis, In: *Handbook of Combinatorial Optimization* (Eds. Du, D.-Z. and Pardalos, P.M.), 151–208, Kluwer Academic Publishers (1999).

[4] Coppersmith, D., Raghavan, P., Multidimsnional online bin packing algorithms and orst case analysis, *Operations Research Letters*, Vol. 8 (1989), 17–20.

[5] Galambos, G., A new heuristic for the classical bin packing problem, Technical Report 82, Institute fuer Mathematik, Augsburg (1985).

[6] Galambos, G., Woeginger, G.J., Repacking helps in bounded space online bin packing, *Computing* Vol. 49 (1993), 329–338.

[7] Gambosi, G., Postiglione, A., Talamo, M., Algorithms for the Relaxed Online Bin-Packing Model, *SIAM Journal on Computing* Vol. 30 (2000), 1532–1551.

[8] Garey, M.R., Johnson, D.S., Computers and Intractability (A Guide to the theory of NP-Completeness), W.H. Freeman and Company, San Francisco (1979).

[9] Grove, E.F., Online bin packing with lookahead, *SODA* (1995), 430–436.

[10] Gutin, G., Jensen, T., Yeo, A., Batched bin packing, *Discrete Optimization* Vol. 2 (2005), 71–82.

[11] Ivkovič, Z., Lloyd, E.L., A fundamental restriction on fully dynamic maintenance of bin packing, *Information Processing Letters* Vol. 59 (1996), 229–232.

[12] Ivkovič, Z., Lloyd, E.L., Fully Dynamic Algorithms for Bin Packing: Being (Mostly) Myopic Helps, *SIAM Journal on Computing* Vol. 28 (1998), 574–611.

[13] Seiden, S.S., On the online bin packing problem, *Journal of the ACM* Vol. 49 (2002), 640–671.

[14] van Vliet, A., An improved lower bound for online bin packing algorithms, *Information Processing Letters* Vol. 43 (1992), 277–284.

János Balogh, József Békési, Gábor Galambos
Department of Informatics' Applications
Gyula Juhász Faculty of Education
University of Szeged
6725. Szeged, Boldogasszony sgt. 6.
Hungary
e-mail: {balogh, bekesi, galambos}@jgytf.u-szeged.hu

Gerhard Reinelt
Institute of Computer Science
University of Heidelberg
Im Neuenheimer Feld 368
D-69120 Heidelberg
Germany
e-mail: gerhard.reinelt@informatik.uni-heidelberg.de