ACADEMIC
PRESS

Journal of Algorithms 44 (2002) 308–320

Journal of
Algorithms

www.academicpress.com

Resource augmentation for online bounded space bin packing [☆]

János Csirik [a] and Gerhard J. Woeginger [b,c,*,1]

[a] *Department of Computer Science, University of Szeged, Aradi vértanúk tere 1, H–6720 Szeged, Hungary*

[b] *Department of Mathematics, University of Twente, PO Box 217, NL-7500 AE Enschede, The Netherlands*

[c] *Institut für Mathematik, Technische Universität Graz, Steyrergasse 30, A-8010 Graz, Austria*

Received 2 January 2001

Abstract

We study online bounded space bin packing in the resource augmentation model of competitive analysis. In this model, the online bounded space packing algorithm has to pack a list L of items in $(0, 1]$ into a small number of bins of size $b \geqslant 1$. Its performance is measured by comparing the produced packing against the optimal offline packing of the list L into bins of size 1.

We present a complete solution to this problem: For every bin size $b \geqslant 1$, we design online bounded space bin packing algorithms whose worst case ratio in this model comes arbitrarily close to a certain bound $\rho(b)$. Moreover, we prove that no online bounded space algorithm can perform better than $\rho(b)$ in the worst case.

Keywords: Online algorithm; Competitive analysis; Resource augmentation; Approximation algorithm; Asymptotic worst case ratio; Bin packing

[☆] A preliminary version of this paper appeared in the Proceedings of the 27th International Colloquium on Automata, Languages and Programming, Lecture Notes in Comput. Sci., Vol. 1853, Springer-Verlag, 2000, 296–304.

[*] Corresponding author.

E-mail addresses: csirik@inf.u-szeged.hu (J. Csirik), g.j.woeginger@math.utwente.nl (G.J. Woeginger).

[1] Supported by the START program Y43-MAT of the Austrian Ministry of Science.

1. Introduction

Resource augmentation (or *extra-resource analysis*) is a technique for analyzing online algorithms that apparently was part of online algorithms from the very beginning. It can be traced back to two early papers by Graham [1,2] from 1966 and 1969, and its also been used in 1985 in the seminal paper of Sleator and Tarjan [3]. Officially, it was introduced to the field and identified as a powerful tool in 1995 by Kalyanasundaram and Pruhs [4]. It is a relaxed notion of competitive analysis in which the online algorithm is given better resources than the optimal offline algorithm to which it is compared. This is, e.g., the case, if the machines of the online algorithm run at slightly higher speed than those of the offline algorithm, or if the online algorithm has more machines than the offline algorithm, or if the production deadlines of the online algorithm are less stringent than those of the offline algorithm. The main idea behind the resource augmentation technique is to give the online algorithm a fairer chance in competing against the omniscient and all-powerful offline algorithm from classical competitive analysis. During the last few years the resource augmentation technique has become a very popular tool, and it has been applied to many problems in scheduling (cf., e.g., Phillips et al. [5] and Edmonds [6]), in paging (Albers et al. [7]), and in combinatorial optimization (Kalyanasundaram and Pruhs [8]). In this paper, we will study the online bounded space bin packing problem in this resource augmentation model.

In the classical bin packing problem, a list $L = \langle a_1, a_2, \ldots\rangle$ of items $a_i \in [0,1]$ has to be packed into the minimum number of unit-size bins. The *offline* optimum $\mathrm{OPT}_1(L)$ is the minimum number of unit-size bins into which the items in L can be fit. A bin packing algorithm is called *online* if it packs all items a_i solely on the basis of the sizes of the items a_j, $1 \leqslant j \leqslant i$, and without any information on subsequent items. A bin packing algorithm uses *k-bounded space* if for each item a_i, the choice of bins to pack it into is restricted to a set of k or fewer *active* bins. Each bin becomes active when it receives its first item, but once it is declared inactive (or *closed*), it can never become active again. An online *bounded space* bin packing algorithm is an online algorithm that uses k-bounded space for some fixed value $k \geqslant 1$. The bounded space restriction models situations in which bins are exported once they are packed (e.g., in packing trucks at a loading dock that has positions for only k trucks, or in communication channels with buffers of limited size in which information moves in large fixed-size blocks).

We investigate the behavior of online bounded space bin packing algorithms that pack the list L into bins of size $b \geqslant 1$. This larger bin size b is the augmented resource of the online algorithm; the offline algorithm has to work with bins of size 1. For an online algorithm A and a bin size b, we denote by $A_b(L)$ the number of bins of size b that algorithm A uses in packing the items in L; this number $A_b(L)$ includes both the active and the inactive bins. The *worst case performance* of algorithm A for bin size b, denoted by $R_b(A)$, is defined as

$$R_b(A) = \lim_{\textsc{Opt}_1(L)\to\infty} \sup_L A_b(L)/\textsc{Opt}_1(L). \tag{1}$$

A small worst case performance means a good quality of the online algorithm. We stress the fact that the worst case performance compares an online algorithm against a completely unrestricted offline algorithm without any reference to bounded space. Another possible performance ratio (that will not be considered in this paper) compares an online bounded space bin packing algorithm against a *weak* offline algorithm that also obeys the bounded space restriction. Such a weak offline algorithm would produce a packing of the items $a_1, a_2, \ldots$ into a sequence of bins that results according to the k-bounded space rule. To the best of our knowledge, this second type of performance ratio has not been investigated so far in the literature, not even for classical online bounded space bin packing without any resource augmentation. We will not consider this concept in the rest of this paper, and we leave it for future research.

Online bin packing is a classical problem in optimization and theoretical computer science. A well-known result by Lee and Lee [9] on classical online bounded space bin packing (without resource augmentation) yields an online algorithm A with $R_1(A) \approx 1.69103$. We refer the reader to Csirik and Woeginger [10] for more information and for an up-to-date survey of this area.

Our results and organization of the paper. In this paper we present a complete analysis of online bounded space bin packing in the resource augmentation model: For every bin size $b \geqslant 1$, we determine the best possible worst case performance $\rho(b)$ over all online bounded space bin packing algorithms. The precise values $\rho(b)$ are defined in Section 2. In Section 3 we state several auxiliary results. In Section 4 we discuss technical properties of the function $\rho(b)$. In Section 5 we design and analyze an online algorithm whose worst case performance comes arbitrarily close to $\rho(b)$. Finally, in Section 6 we prove that no online algorithm can beat the bound $\rho(b)$.

2. Statement of the main result

Throughout the paper, $L = \langle a_1, a_2, \ldots, a_n \rangle$ is a list of items in $(0, 1]$, and $b \geqslant 1$ is the bin size for the online algorithm. We associate with b an infinite sequence $T(b) = \langle t_1, t_2, \ldots \rangle$ of positive integers as follows:

$$t_1 = \lfloor 1 + b \rfloor \quad \text{and} \quad r_1 = \frac{1}{b} - \frac{1}{t_1}, \tag{2}$$

and for $i = 1, 2, \ldots,$

$$t_{i+1} = \left\lfloor 1 + \frac{1}{r_i} \right\rfloor \quad \text{and} \quad r_{i+1} = r_i - \frac{1}{t_{i+1}}. \tag{3}$$

An equivalent way for defining this sequence $T(b)$ is the following: Suppose that we want to fill a bucket of size $1/b$ greedily with reciprocal values of positive integers. First, we pack the largest possible reciprocal value that fits into the bucket, but without filling it completely. Then we add the largest reciprocal value that fits without filling the rest capacity completely, and then this process is repeated over and over again. In this 'bucket' interpretation, the value r_i represents the rest capacity after the reciprocal value of the positive integer t_i has been put into the bucket. Note that the smallest integer whose reciprocal would fit into a space of $r \leqslant 1$ is $\lceil 1/r \rceil$. If $1/r$ happens to be an integer, we must not fill the bucket completely, and hence we have to pack the reciprocal of $\lceil 1/r \rceil + 1$ instead. The reader may want to verify that the recursive definitions in (2) and (3) exactly agree with these interpretations. Altogether, this discussion demonstrates that

$$\frac{1}{b} = \sum_{i=1}^{\infty} \frac{1}{t_i} = \frac{1}{t_1} + \frac{1}{t_2} + \frac{1}{t_3} + \frac{1}{t_4} + \cdots. \tag{4}$$

Finally, we define

$$\rho(b) = \sum_{i=1}^{\infty} \frac{1}{t_i - 1}. \tag{5}$$

In Section 3 we will prove that the infinite sum in the right-hand side of (5) converges for every value of b. The following lemma provides the reader with some intuition on the (somewhat irregular and somewhat messy) behavior of the function $\rho(b)$; see also the picture in Fig. 1 for an illustration. The lemma will be proved in Section 4.

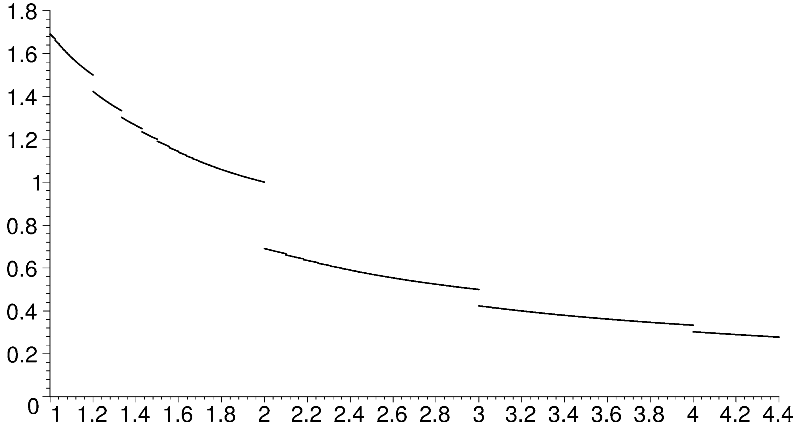

Fig. 1. The graph of the function $\rho(b)$.

Lemma 2.1. *The function $\rho(b) : [1,\infty) \to \mathbb{R}$ has the following properties*:

(i) $\rho(1) \approx 1.69103$ *and* $\rho(2) \approx 0.69103$.
(ii) $1/m \leqslant \rho(m) \leqslant 1/(m-1)$ *for integers* $m \geqslant 2$.
(iii) $\rho(b)$ *is strictly decreasing on* $[1,\infty)$.
(iv) *As b tends to* 2 *from below,* $\rho(b)$ *tends to* 1. *As b tends to infinity,* $\rho(b)$ *tends to* 0.
(v) *At every irrational value of $b > 1$, the function $\rho(b)$ is continuous.*
(vi) *At every rational value of $b > 1$, the function $\rho(b)$ is not continuous.*

The following theorem summarizes the main result of this paper. Its proof is split into the proof of the upper bound in Theorem 5.4 in Section 5, and into the proof of the lower bound in Theorem 6.1 in Section 6.

Theorem 2.2 (Main result of the paper). *For every bin size $b \geqslant 1$, there exist online bounded space bin packing algorithms with worst case performance arbitrarily close to $\rho(b)$. For every bin size $b \geqslant 1$, the bound $\rho(b)$ cannot be beaten by an online bounded space bin packing algorithm.*

Note that by setting $b = 1$ in Theorem 2.2 we get a worst case performance of $\rho(1) \approx 1.69103$. Hence, this special case reproves the well-known result of Lee and Lee [9] on classical online bounded space bin packing. The best known lower bound for general (not bounded space) online bin packing is 1.5401; see Van Vliet [11].

3. Some useful facts

In this section we collect several facts on the sequence $T(b)$ that will be used in the later sections. First, we observe that for every $b \geqslant 1$ the corresponding sequence $T(b) = \langle t_1, t_2, \ldots \rangle$ is growing rapidly: By the equations in (3), we have $r_{i-1} \leqslant 1/(t_i - 1)$ and $1/t_{i+1} < r_i = r_{i-1} - 1/t_i$. Consequently, $1/t_{i+1} < 1/(t_i - 1) - 1/t_i$. Rewriting this yields the inequality $t_{i+1} > t_i(t_i - 1)$, which in turn is equivalent to

$$t_{i+1} - 1 \geqslant t_i(t_i - 1) \quad \text{for all } i \geqslant 1. \tag{6}$$

Next, consider some fixed index $j \geqslant 1$. A straightforward inductive argument based on (6) yields that $t_{j+k} - 1 \geqslant (t_j - 1)^{k+1}$ holds for all $k \geqslant 0$. From this we get that

$$\sum_{i=j}^{\infty} \frac{1}{t_i - 1} = \sum_{k=0}^{\infty} \frac{1}{t_{j+k} - 1} \leqslant \sum_{k=0}^{\infty} (t_j - 1)^{-k-1} = \frac{1}{t_j - 2}. \tag{7}$$

For $j = 1$ this inequality demonstrates that the infinite series in Eq. (5) indeed converges, and that the function $\rho(b)$ is well-defined.

The following result will be used in the proof of Lemma 5.3.

Lemma 3.1. *Let $z \geqslant 1$ be an integer. Then the sequence $T(b)$ fulfills the inequality*

$$\frac{t_z + 1}{t_z} \cdot \sum_{i=z}^{\infty} \frac{1}{t_i} \leqslant \sum_{i=z}^{\infty} \frac{1}{t_i - 1}. \tag{8}$$

Proof. By (2) and (3), the sum in the left-hand side of (8) is at most $1/t_z + 1/(t_{z+1} - 1)$. On the other hand, the sum in right-hand side of (8) is at least $1/(t_z - 1) + 1/(t_{z+1} - 1)$. These two bounds together with $t_{z+1} \geqslant t_z(t_z - 1) + 1$ from (6) imply the claimed inequality. □

4. Some properties of the function $\rho(b)$

This section is devoted to the proof of Lemma 2.1. Since by (6) the underlying series converges fast, the values $\rho(1)$ and $\rho(2)$ in statement (i) of Lemma 2.1 are easy to approximate by a computer program. For statement (ii), consider an integer $m \geqslant 2$. Since the sequence $T(m)$ starts with $t_1 = m + 1$, the definition of $\rho(b)$ in (5) immediately yields $\rho(m) \geqslant 1/m$. Moreover, by setting $j = 1$ in inequality (7) we get that

$$\rho(m) = \sum_{i=1}^{\infty} \frac{1}{t_i - 1} \leqslant \frac{1}{t_1 - 2} = \frac{1}{m - 1} \quad \text{for all integers } m \geqslant 2. \tag{9}$$

This completes the proof of statement (ii). We turn to statement (iii). Let $1 \leqslant a < b$, and let $T(a) = \langle t_i \rangle$ and $T(b) = \langle t_i' \rangle$ denote the two infinite sequences associated with a and b. Define $j \geqslant 1$ to be the smallest index with $t_j \neq t_j'$. Since $a < b$, this implies $t_j \leqslant t_j' - 1$. Then

$$\rho(a) - \rho(b) = \sum_{i=j}^{\infty} \frac{1}{t_i - 1} - \sum_{i=j}^{\infty} \frac{1}{t_i' - 1} > \frac{1}{t_j - 1} - \frac{1}{t_j' - 2} \geqslant 0, \tag{10}$$

where we used (7) to derive the first inequality and $t_j \leqslant t_j' - 1$ in the second inequality. Hence, $a < b$ indeed implies $\rho(a) > \rho(b)$.

Next, we turn to statement (iv). Let $m \geqslant 2$ be an integer and consider the value $b_m = 2m/(m + 2)$. It can be verified that the series $T(b_m)$ starts with the term $t_1 = 2$, which is followed by the all the terms of the sequence $T(m)$. Consequently, $\rho(b_m) = 1 + \rho(m)$ holds and from (9) we get that $1 + 1/m \leqslant \rho(b_m) \leqslant 1 + 1/(m - 1)$. As m goes to ∞, b_m tends to 2 from below, and $\rho(b_m)$ tends to 1 from above. Since $\rho(b)$ is a decreasing function by statement (iii), we have thus proved the first part of statement (iv). The second part of statement (iv) follows by combining statements (ii) and (iii).

We turn to statement (v). Let $b \geqslant 1$ be an arbitrary irrational number, and let $\varepsilon > 0$ be an arbitrary real number. Consider the sequence $T(b) = \langle t_i \rangle$, and let j be the smallest index with $1/(t_j - 2) < \varepsilon$. Since b is irrational and by the definition of $T(b)$,

$$\frac{1}{b} < \left(\sum_{i=1}^{j-1} \frac{1}{t_i} \right) + \frac{1}{t_j - 1}. \tag{11}$$

(For rational values b, besides the stated inequality also equality may hold true.) Our goal is to show that for every c sufficiently close to b, $\rho(c)$ is at most ε away from $\rho(b)$. We will deal separately with the two cases $c > b$ and $c < b$. First consider an arbitrary $c > b$ such that $1/c > \sum_{i=1}^{j-1} 1/t_j$. Then

$$\left|\rho(b) - \rho(c)\right| \leqslant \sum_{i=1}^{\infty} \frac{1}{t_i - 1} - \sum_{i=1}^{j-1} \frac{1}{t_i - 1} = \sum_{i=j}^{\infty} \frac{1}{t_i - 1} \leqslant \frac{1}{t_j - 2} < \varepsilon, \tag{12}$$

where we used inequality (7). Next consider an arbitrary $c < b$ with $1/b < 1/c < 1/(t_j - 1) + \sum_{i=1}^{j-1} 1/t_j$. By (11), such values of c indeed exist. Then the sequence $T(c)$ starts with the j terms $t_1, \ldots, t_{j-1}, t_j$, and we have

$$\left|\rho(c) - \rho(b)\right| \leqslant \sum_{i=j+1}^{\infty} \frac{1}{t_i - 1} \leqslant \frac{1}{t_{j+1} - 2} < \varepsilon. \tag{13}$$

The inequalities in (12) and (13) demonstrate that $\rho(b)$ is continuous at b, exactly as we desired. This completes the proof of statement (v).

Finally, we turn to statement (vi). Let $b \geqslant 1$ be an arbitrary rational number. We consider the additive representation of $1/b$ as a finite sum of Egyptian fractions obtained by the greedy algorithm (cf., e.g., Niven and Zuckerman [12] or pages 271–277 of Wagon [13]). An *Egyptian fraction* simply is the reciprocal value of a positive integer. Every positive rational number can be represented as the sum of a finite number of Egyptian fractions. One way of getting such a finite representation of $1/b$ is by a greedy algorithm: Repeatedly subtract the largest possible Egyptian fraction until you reach zero. It is known that this greedy algorithm terminates after a finite number, say j, of steps. Comparing the outcome of this procedure to (2)–(4), we see that

$$\frac{1}{b} = \left(\sum_{i=1}^{j-1} \frac{1}{t_i} \right) + \frac{1}{t_j - 1}, \tag{14}$$

where $t_1, \ldots, t_{j-1}, t_j$ are just the first j terms of the sequence $T(b)$. Since $b > 1$, we have $t_j \geqslant 3$. Now consider an arbitrary real value $c < b$ just slightly smaller than b that fulfills

$$\frac{1}{c} < \frac{1}{b} + \frac{1}{(t_j - 1)(t_j - 2)} = \left(\sum_{i=1}^{j-1} \frac{1}{t_i}\right) + \frac{1}{t_j - 2}. \tag{15}$$

By the choice of c, the sequence $T(c)$ starts with the j terms $t_1, \ldots, t_{j-1}, t_j - 1$ that are followed by the terms $t'_{j+1}, t'_{j+2}, \ldots$. Then

$$\rho(c) = \sum_{i=1}^{j-1} \frac{1}{t_i - 1} + \frac{1}{t_j - 2} + \sum_{i=j+1}^{\infty} \frac{1}{t'_i - 1} > \sum_{i=1}^{j-1} \frac{1}{t_i - 1} + \frac{1}{t_j - 2} \tag{16}$$

and

$$\begin{aligned} \rho(b) &= \sum_{i=1}^{j-1} \frac{1}{t_i - 1} + \frac{1}{t_j - 1} + \sum_{i=j+1}^{\infty} \frac{1}{t_i - 1} \\ &< \sum_{i=1}^{j-1} \frac{1}{t_i - 1} + \frac{1}{t_j - 1} + \frac{1}{t_{j+1} - 2}. \end{aligned} \tag{17}$$

Here we used (7). By applying $t_{j+2} - 2 \geqslant t_j(t_j - 1) - 1$ from (6), the last two inequalities yield

$$\begin{aligned} \rho(c) - \rho(b) &> \frac{1}{t_j - 2} - \frac{1}{t_j - 1} - \frac{1}{t_{j+1} - 2} \\ &\geqslant \frac{1}{(t_j - 2)(t_j - 1)} - \frac{1}{t_j(t_j - 1) - 1}. \end{aligned} \tag{18}$$

Since $t_j \geqslant 3$, the value of the right-hand side in (18) is strictly bounded away from 0. Hence, the function ρ is not continuous in b, and this completes the proof of statement (vi).

5. Proof of the upper bound

In this section, we prove the upper bound stated in Theorem 2.2. As usual, let $b \geqslant 1$ denote the bin size, and let $T(b) = \langle t_1, t_2, \ldots \rangle$ be the integer sequence associated with b. Let $\ell \geqslant 3$ be an integer. We introduce t_ℓ intervals $\mathcal{I}_j$ with $j = 1, \ldots, t_\ell$ that form a partition of the interval $(0, b]$. For $1 \leqslant j \leqslant t_\ell - 1$, we define the interval $\mathcal{I}_j = \left(\frac{b}{j+1}, \frac{b}{j}\right]$. Moreover, we define the last interval $\mathcal{I}_{t_\ell} = (0, b/t_\ell]$.

Our online algorithm keeps one active bin $\mathcal{B}_j$ for every interval $\mathcal{I}_j$ ($j = 1, \ldots, t_\ell$). Hence, it uses t_ℓ-bounded space. All items from the interval $\mathcal{I}_j \cap (0, 1]$

are packed into the corresponding active bin $\mathcal{B}_j$. If a newly arrived item does not fit into $\mathcal{B}_j$, this bin is closed, and a new corresponding bin for interval $\mathcal{I}_j$ is opened. In other words, the items from interval $\mathcal{I}_j \cap (0, 1]$ are packed into the active bins $\mathcal{B}_j$ according to the NEXT-FIT algorithm. This completes the description of the online algorithm.

To analyze this online algorithm, we define the following *weight function* $w : (0, 1] \to \mathbb{R}$. For items x in $\mathcal{I}_j$ with $1 \leqslant j \leqslant t_\ell - 1$, we define $w(x) = 1/j$. For items x in the last interval $\mathcal{I}_{t_\ell}$, we define $w(x) = (xt_\ell)/(bt_\ell - b)$. The weight of a packed bin equals the sum of the weights of the items contained in this bin. The weight $w(L)$ of an item list L equals the sum of the weights of the items in L.

Lemma 5.1. *Every bin of size b that has been closed by the online algorithm contains items of total weight at least* 1.

Proof. First assume that the closed bin belongs to an interval $\mathcal{I}_j$ with $1 \leqslant j \leqslant t_\ell - 1$. Then it contains exactly j items, and each of these items has weight $1/j$. Next assume that the closed bin belongs to the interval $\mathcal{I}_{t_\ell}$. Then the bin has been closed, since a new item from $\mathcal{I}_{t_\ell}$ did not fit into it. Hence, the total size of its items is at least $b - b/t_\ell$. Since on the interval $\mathcal{I}_{t_\ell}$ the weight function is linear with slope $t_\ell/(bt_\ell - b)$, the weight of such a bin is at least 1. □

Lemma 5.2. *Let $1 \leqslant z \leqslant \ell - 1$ be an integer. Then for every positive real number $x \leqslant b/t_z$, we have $w(x)/x \leqslant (t_z + 1)/(bt_z)$.*

Proof. First assume that x is from some interval $\mathcal{I}_j$ with $t_z \leqslant j \leqslant t_\ell - 1$. Then $x \geqslant b/(j + 1)$ and $w(x) = 1/j$, and thus $w(x)/x \leqslant (j + 1)/(jb)$ holds. Since the expression $(j + 1)/(jb)$ is decreasing in j and since $j \geqslant t_z$, we get that $w(x)/x \leqslant (t_z + 1)/(bt_z)$ holds, exactly as we desired. Next assume that x is in the interval $\mathcal{I}_{t_\ell}$. Then

$$w(x)/x = t_\ell/(bt_\ell - b) \leqslant (t_z + 1)/(bt_z),$$

where the final inequality follows from $t_\ell \geqslant t_z + 1$. □

Lemma 5.3. *In any packing of the list L into unit-size bins, every unit-size bin receives items of total weight at most*

$$\sum_{i=1}^{\ell} \frac{1}{t_i - 1} + \frac{1}{(t_\ell - 1)^2}. \tag{19}$$

Proof. Consider some fixed unit-size bin $\mathcal{B}$ that contains the items $f_1 \geqslant f_2 \geqslant \cdots \geqslant f_n$ with total size at most 1. We distinguish three cases that depend on the distribution of the item sizes f_i in the intervals $(b/t_j, b/(t_j - 1)]$.

Case 1. For $i = 1, \ldots, \ell$ we have $f_i \in (b/t_i, b/(t_i - 1)]$. We denote by F the sum of the sizes of the remaining items f_i with $i > \ell$. By the definition of the values t_i in (2) and (3), we conclude that

$$F = \sum_{i=\ell+1}^{n} f_i \leqslant 1 - \sum_{i=1}^{\ell} \frac{b}{t_i} = b \cdot r_\ell \leqslant \frac{b}{t_{\ell+1} - 1}. \tag{20}$$

Hence, all items $f_{\ell+1}, \ldots, f_n$ are in the last interval $\mathcal{I}_{t_\ell}$. By the definition of the weight function, the weight of the bin $\mathcal{B}$ then is upper bounded by

$$\begin{aligned} \sum_{i=1}^{\ell} \frac{1}{t_i - 1} + \frac{t_\ell}{bt_\ell - b} F &\leqslant n \sum_{i=1}^{\ell} \frac{1}{t_i - 1} + \frac{t_\ell}{(t_\ell - 1)(t_{\ell+1} - 1)} \\ &\leqslant n \sum_{i=1}^{\ell} \frac{1}{t_i - 1} + \frac{1}{(t_\ell - 1)^2}. \end{aligned}$$

Here we used (20) to derive the first inequality, and (6) to derive the second inequality. This completes the analysis of the first case.

Case 2. There exists an integer z with $1 \leqslant z \leqslant \ell - 1$ such that the following holds: For $i = 1, \ldots, z - 1$ we have $f_i \in (b/t_i, b/(t_i - 1)]$. Moreover, f_z either does not exist (since $n = z - 1$ holds) or if it does exist then $f_z \notin (b/t_z, b/(t_z - 1)]$ holds. We denote by F the sum of the sizes of the remaining items f_i with $i \geqslant z$. Similarly as above, we observe that

$$F = \sum_{i=z}^{n} f_i \leqslant n1 - \sum_{i=1}^{z-1} \frac{b}{t_i} = \sum_{i=z}^{\infty} \frac{b}{t_i}. \tag{21}$$

By combining inequality (21) with (7) we get that the total size F of all items $f_z, \ldots, f_n$ is at most $b/(t_z - 1)$. Since the largest one of all these items, f_z, is not contained in the interval $(b/t_z, b/(t_z - 1)]$, we conclude that the size of every item $f_z, \ldots, f_n$ is at most b/t_z. Then by Lemma 5.2, their overall weight is at most $F(t_z + 1)/(bt_z)$. The weight of the bin $\mathcal{B}$ is at most

$$\begin{aligned} \sum_{i=1}^{z-1} \frac{1}{t_i - 1} + \frac{F(t_z + 1)}{bt_z} &\leqslant \sum_{i=1}^{z-1} \frac{1}{t_i - 1} + \frac{t_z + 1}{t_z} \sum_{i=z}^{\infty} \frac{1}{t_i} \leqslant n \sum_{i=1}^{\infty} \frac{1}{t_i - 1} \\ &\leqslant \sum_{i=1}^{\ell} \frac{1}{t_i - 1} + \frac{1}{t_{\ell+1} - 2} \\ &\leqslant n \sum_{i=1}^{\ell} \frac{1}{t_i - 1} + \frac{1}{(t_\ell - 1)^2}. \end{aligned}$$

Here we have first applied (21) to bound F from above, then the statement in Lemma 3.1, then the inequality in (7) to bound $\sum_{i=\ell+1}^{\infty} 1/(t_i - 1)$ from above, and in the end the inequality (6) together with $t_\ell \geqslant 2$. This completes the analysis of the second case.

Case 3. This case is essentially the second case with $z = \ell$, which needs special treatment since the statement in Lemma 5.2 does not carry over to $z = \ell$. Assume that for $i = 1, \ldots, \ell - 1$ we have $f_i \in (b/t_i, b/(t_i - 1)]$, and that $f_\ell \notin (b/t_\ell, b/(t_\ell - 1)]$; the subcase where f_ℓ does not exist is trivial. We denote by F the sum of the sizes of the items f_i with $i \geqslant \ell$:

$$F = \sum_{i=\ell}^{n} f_i \leqslant 1 - \sum_{i=1}^{\ell-1} \frac{b}{t_i} = b \cdot r_{\ell-1} \leqslant \frac{b}{t_\ell - 1}. \tag{22}$$

Consequently, all items $f_\ell, \ldots, f_n$ are contained in the last interval $\mathcal{I}_{t_\ell}$. Then the weight of the bin $\mathcal{B}$ is at most

$$\begin{aligned}\sum_{i=1}^{\ell-1} \frac{1}{t_i - 1} + \frac{t_\ell}{bt_\ell - b} F &\leqslant \sum_{i=1}^{\ell-1} \frac{1}{t_i - 1} + \frac{t_\ell}{(t_\ell - 1)^2} \\ &= \sum_{i=1}^{\ell} \frac{1}{t_i - 1} + \frac{1}{(t_\ell - 1)^2}.\end{aligned}$$

Here we used (22) to bound F. This completes the proof. □

Theorem 5.4. *For any bin size $b > 1$ and for any real $\varepsilon > 0$, there exist a sufficiently large k and an online k-bounded space bin packing algorithm A with $R_b(A) \leqslant \rho(b) + \varepsilon$.*

Proof. Choose a sufficiently large integer $\ell \geqslant 3$ such that $1/(t_\ell - 1)^2 \leqslant \varepsilon$ is fulfilled. By the definition of the worst case ratio in (1), we may restrict our attention to item lists L with $\mathrm{OPT}_1(L) \geqslant t_\ell/\varepsilon$.

We derive from Lemma 5.1 that $A_b(L) \leqslant w(L) + t_\ell$, where the term $w(L)$ accounts for the number of closed bins and where the term t_ℓ accounts for the number of open bins in the final packing. We derive from Lemma 5.3 that $w(L) \leqslant (\rho(b) + \varepsilon) \cdot \mathrm{OPT}_1(L)$. Consequently, $A_b(L) \leqslant (\rho(b) + 2\varepsilon) \cdot \mathrm{OPT}_1(L)$. □

6. Proof of the lower bound

In this section, we prove the lower bound stated in Theorem 2.2. To avoid misunderstandings, we repeat the definition of the offline optimum from the introduction: The offline optimum $\mathrm{OPT}_1(L)$ is the minimum number of unit-size bins into which the items in the list L can be fit. Hence, the offline algorithm can pack the items in arbitrary order, and it is not restricted in any way to the order in which the items are received or to bounded space.

Now consider an arbitrary online k-bounded space algorithm A for bin packing with bin size b. Let $T(b) = \langle t_1, t_2, \ldots \rangle$ be the integer sequence associated with b. Let ℓ be an integer, and let $\varepsilon > 0$ be a small real number such that $\varepsilon \cdot t_{\ell+1} \cdot \ell \leqslant 1$. Furthermore, let $N > k^3 t_{\ell+1}$ be a huge integer. We confront the online algorithm with several phases of 'bad' items, and we show that algorithm A eventually must perform poorly.

Altogether there are ℓ phases. In the jth phase ($j = 1, \ldots, \ell$), exactly N items of size $b/t_{\ell-j+1} + \varepsilon$ arrive. Up to some exceptional cases at the very beginning and the end of such a phase, the best that the bounded space algorithm A can do is to pack these items together in groups of cardinality $t_{\ell-j+1} - 1$ each. This consumes $N/(t_{\ell-j+1} - 1)$ bins. At the beginning of a phase up to k used bins of the previous phase are active, and this may save up to k bins. Summarizing, algorithm A uses at least $N/(t_{\ell-j+1} - 1) - k$ bins for packing the items of phase j. Adding this up over all $j = 1, \ldots, \ell$, we get that

$$A_b(L) \geqslant \sum_{j=1}^{\ell} \left(\frac{N}{t_{\ell-j+1} - 1} - k \right) = N \cdot \sum_{j=1}^{\ell} \frac{1}{t_j - 1} - k\ell. \tag{23}$$

By (4) and by the choice of ε, the ℓ items $b/t_{\ell-j+1} + \varepsilon$ with $1 \leqslant j \leqslant \ell$ together fit into a bin of size 1. Consequently, we have $\textsc{Opt}_1(L) \leqslant N$. By making N sufficiently large, (23) yields that the worst case performance $R_b(A)$ of algorithm A is at least $\sum_{j=1}^{\ell} 1/(t_j - 1)$. Since this statement holds true for every value of ℓ, we may make ℓ arbitrarily large and thus make this bound arbitrarily close to $\rho(b)$.

Theorem 6.1. *For any $b \geqslant 1$ and for any online k-bounded space bin packing algorithm A, we have $R_b(A) \geqslant \rho(b)$.*

Acknowledgments

We thank Clemens Heuberger for several discussions. We thank Bettina Klinz for helping us in generating the picture. We thank the two referees for a very careful reading of the paper, and for pointing out a number of mistakes in an earlier version of this paper.

References

[1] R.L. Graham, Bounds for certain multiprocessing anomalies, Bell Syst. Tech. J. 45 (1966) 1563–1581.

[2] R.L. Graham, Bounds on multiprocessing timing anomalies, SIAM J. Appl. Math. 17 (1969) 263–269.

[3] D.D.K.D.B. Sleator, R.E. Tarjan, Amortized efficiency of list update and paging rules, Comm. ACM 28 (1985) 202–208.

[4] B. Kalyanasundaram, K. Pruhs, Speed is more powerful than clairvoyance, in: Proceedings of the 36th IEEE Symposium on Foundations of Computer Science (FOCS'95), 1995, pp. 214–221.

[5] C.A. Phillips, C. Stein, E. Torng, J. Wein, Optimal time-critical scheduling via resource augmentation, in: Proceedings of the 29th ACM Symposium on Theory of Computing (STOC'97), 1997, pp. 140–149.

[6] J. Edmonds, Scheduling in the dark, in: Proceedings of the 31st Annual ACM Symposium on the Theory of Computing (STOC'99), 1999, pp. 179–188.

[7] S. Albers, S. Arora, S. Khanna, Page replacement for generalized caching problems, in: Proceedings of the 10th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA'99), 1999, pp. 31–40.

[8] B. Kalyanasundaram, K. Pruhs, The online transportation problem, in: Proceedings of the 3rd European Symposium on Algorithms (ESA'95), in: Lecture Notes in Comput. Sci., Vol. 979, Springer-Verlag, Berlin, 1995, pp. 484–493.

[9] C.C. Lee, D.T. Lee, A simple online bin-packing algorithm, J. ACM 32 (1985) 562–572.

[10] J. Csirik, G.J. Woeginger, Online packing and covering problems, in: Online Algorithms: The State of the Art, in: Lecture Notes in Comput. Sci., Vol. 1442, Springer-Verlag, 1998, pp. 147–178.

[11] A. van Vliet, An improved lower bound for on-line bin packing algorithms, Inform. Process. Lett. 43 (1992) 277–284.

[12] I.M. Niven, H.S. Zuckerman, An Introduction to the Theory of Numbers, Wiley, 1960.

[13] S. Wagon, Mathematica in Action, Freeman, 1991.