Nordic Journal of Computing

THE COMPETITIVE RATIO FOR ON-LINE DUAL BIN PACKING WITH RESTRICTED INPUT SEQUENCES *†

Joan Boyar Lene M. Favrholdt Kim S. Larsen Morten N. Nielsen

Abstract. We consider the On-Line Dual Bin Packing problem where we have a fixed number n of bins of equal size and a sequence of items. The goal is to maximize the number of items that are packed in the bins by an on-line algorithm. An investigation of First-Fit and an algorithm called Log shows that, in the special case where all sequences can be completely packed by an optimal off-line algorithm, First-Fit has a constant competitive ratio, but Log does not. In contrast, if there is no restriction on the input sequences, Log is exponentially better than First-Fit. This is the first separation of this sort with a difference of more than a constant factor. We also design randomized and deterministic algorithms for which the competitive ratio is constant on sequences which the optimal off-line algorithm can pack using at most αn bins, if α is constant and known to the algorithm in advance.

1. Introduction

First in this introduction, we present the On-Line Dual Bin Packing problem. Then we discuss the relevant performance measures and give an overview of our results.

1.1 The Problem

Bin Packing is one of the most classical problems in combinatorial optimization and in theoretical computer science. In the *Classical Bin Packing problem* we are given an unlimited number of bins and a set of items, each with a positive size, where the goal is to minimize the number of bins used to pack all the items. (For surveys on Classical Bin Packing, see [9, 12].) In the *Dual Bin Packing problem*, we are given a fixed number n of bins and a set of items, each with a positive size, where the goal is to maximize the

*A preliminary version of this paper appeared as: Yossi Azar, Joan Boyar, Lene M. Favrholdt, Kim S. Larsen, Morten N. Nielsen. "Fair versus Unrestricted Bin Packing". *Proceedings of the Seventh Scandinavian Workshop on Algorithm Theory*, Lecture Notes in Computer Science, vol. 1851, pages 200–213, Springer-Verlag, 2000.

† Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark, {joan,lenem,kslarsen,nyhave}@imada.sdu.dk. Supported in part by the Danish Natural Science Research Council (SNF) and in part by the Future and Emerging Technologies programme of the EU under contract number IST-1999-14186 (ALCOM-FT).

Received August 2001. Revised November 2001.

number of items packed. In [8] this problem is reported to have been named Dual Bin Packing in [15]. The Dual Bin Packing problem has been studied in the off-line setting, starting in [10], and its applicability to processor and storage allocation is discussed in [11]. In the on-line version of the problem, *On-Line Dual Bin Packing*, the items arrive one by one and each item must be packed without knowledge of future items.

In a variant of On-Line Dual Bin Packing called Fair Bin Packing [7], an algorithm is only allowed to reject an item if it cannot fit in any bin at the time when it is given. The problem *On-Line Dual Bin Packing* studied in this paper (and in [5]) is the same problem with the fairness restriction removed.

We assume that all items are integer-sized and the bins have size k. All of the results in this paper hold with the weaker assumption that the bins are unit-sized and the smallest item is known to have size at least $\frac{1}{k}$. However, some of the results in [7] do not appear to hold with this assumption, so we use the stronger assumption for consistency.

1.2 The Performance Measures

The standard measure for the quality of on-line algorithms is the *competitive ratio*. For the On-Line Dual Bin Packing problem, the competitive ratio of an algorithm $\mathbb{A}$ is the worst case ratio, over all possible input sequences, of the number of items packed by $\mathbb{A}$ to the number of items packed by an optimal off-line algorithm.

For the On-Line Dual Bin Packing problem, as well as for many other on-line problems, the competitive ratio yields very pessimistic results. Restricting the input sequences to those which can be completely packed by an optimal off-line algorithm, we obtain significantly different results. Such sequences are called *accommodating sequences*, since the off-line algorithm can accommodate the whole sequence.

Note that on accommodating sequences, the competitive ratio of On-Line Dual Bin Packing is no worse than the competitive ratio of the fair problem, since the off-line algorithm packs all items and hence is fair. In general, however, the competitive ratio of On-Line Dual Bin Packing is not necessarily better than the competitive ratio of the fair problem since also the off-line algorithm can benefit from not being fair. In fact, in many cases, considering unfair algorithms, i.e., performing admission control on the items, is the more challenging problem; see for example the results for throughput routing in [1, 2, 3]. In particular, with the On-Line Dual Bin Packing problem, the competitive ratio of different algorithms can vary over a large range. This is in contrast to deterministic on-line algorithms for Fair Bin Packing where all competitive ratios are within a constant factor of each other, in the standard case of unrestricted sequences as well as in the special case of accommodating sequences.

The notion of accommodating sequences can be extended to α-*sequences*. A sequence of items is an α-sequence, if an optimal off-line algorithm can pack all items of the sequence using at most αn bins. In this paper, we concentrate on $\alpha \geq 1$.

1.3 The Results

We give an overview of the results obtained. The separation result depends on the results obtained for Log and First-Fit on unrestricted as well as accommodating sequences.

1.3.1 Unrestricted Sequences

In Section 3.2, we describe a "classify and randomly select" [2] algorithm, Log, with competitive ratio $\Theta(\frac{1}{\log k})$ and show that this is optimal. If $n \geq c \lceil \log_2 k \rceil$, for some constant $c > 1$, a derandomized version of the algorithm can be used. In contrast, it is shown in [7] that the competitive ratio of any deterministic on-line algorithm for Fair Bin Packing is $\Theta(\frac{1}{k})$.

1.3.2 Accommodating Sequences

If the sequences are all accommodating sequences, there exist algorithms for the On-Line Dual Bin Packing problem with constant competitive ratios. First-Fit, the algorithm which packs each item in the lowest indexed bin into which it fits and only rejects items which do not fit in any bin, is studied in [7]. It is shown there that, for the Fair Bin Packing problem, First-Fit has a competitive ratio of at least $\frac{5}{8}$ on accommodating sequences. Since the competitive ratio on accommodating sequences is no worse when the fairness restriction is removed, this result is also valid for On-Line Dual Bin Packing. In [4] (of which a preliminary version appeared in [5]) it is shown that the bound is asymptotically tight, i.e., there exist values of n for which First-Fit's competitive ratio on accommodating sequences is arbitrarily close to $\frac{5}{8}$.

The competitive ratio of the algorithm Log is $\Theta(\frac{1}{\log k})$ on accommodating sequences, as on unrestricted sequences.

1.3.3 A Non-Constant Separation

From the discussion above it follows that for unrestricted sequences, the algorithm Log has the better competitive ratio, and in the special case of accommodating sequences, First-Fit has the better competitive ratio. This is the first case of two algorithms, where one has the better competitive ratio on unrestricted sequences and the other the better competitive ratio on accommodating sequences, and the difference is more than a constant factor. In [7], a similar result was shown about First-Fit and Worst-Fit, but only a constant separation could be obtained.

1.3.4 Knowing the Minimum Resources Necessary

We design randomized and deterministic on-line algorithms for which the competitive ratio is a constant on α-sequences, if α is a constant and known to the on-line algorithm in advance. In contrast, we observe that for First-Fit the competitive ratio on accommodating sequences drops down to $\Theta(\frac{1}{k})$ for $\alpha \geq 1 + c$, for any constant $c > 0$.

2. The Performance Measures

This section contains formal definitions of the performance measures discussed in the introduction.

For completeness, we define the competitive ratio and α-sequences. The competitive ratio as a function of α is also called the *accommodating function.* Note that On-Line Dual Bin Packing is a maximization problem, and all ratios are less than or equal to one.

Let $\mathbb{A}(I)$ denote the number of items algorithm $\mathbb{A}$ accepts when given the sequence I and let $\mathrm{OPT}(I)$ denote the number of items an optimal off-line algorithm, OPT, accepts.

Definition 1. *An on-line algorithm, $\mathbb{A}$, is* c-competitive *if*

$$\mathbb{A}(I) \geq c \cdot OPT(I), \text{ for all input sequences } I.$$

The competitive ratio $CR = \sup\{c \mid \mathbb{A} \text{ is } c\text{-competitive}\}$.

Sometimes in the definition of the competitive ratio, an additive term is allowed, so the requirement is weakened to $\mathbb{A}(I) \geq c \cdot \mathrm{OPT}(I) - b$, where b is a fixed constant independent of I [6]. In that situation, our definition would then be referred to as the *strict* competitive ratio. However, we will not need the additive term in this paper.

Furthermore, one could have chosen to focus on the inverse ratio to obtain numbers larger than one. However, we made our choice for consistency with similar decisions in the area of approximation algorithms where ratios for maximization problems are smaller than one and the inverse is referred to as the approximation factor [14].

Definition 2. *Let $\alpha > 0$. A sequence of items is called an* α-sequence *if an optimal off-line algorithm can pack the whole sequence using at most αn bins. Let $0 < c \leq 1$. An on-line algorithm $\mathbb{A}$ is* c-competitive on α-sequences *if $\mathbb{A}(I) \geq c \cdot OPT(I)$, for any α-sequence I.*

The accommodating function $\mathcal{A}$ *is defined as*

$$\mathcal{A}(\alpha) = \sup\{c \mid \mathbb{A} \textit{ is } c\textit{-competitive on } \alpha\textit{-sequences}\}.$$

DEFINITION 3. *1-sequences are also called* accommodating sequences. *The* competitive ratio on accommodating sequences *is*

$$AR = \sup\{c \mid \mathbb{A} \textit{ is } c\textit{-competitive on accommodating sequences}\}.$$

3. A Non-Constant Separation

We discuss First-Fit, Log, and general hardness results separately.

3.1 First-Fit

It is easy to see that the competitive ratio of First-Fit for On-Line Dual Bin Packing is $\frac{1}{k}$. For the upper bound, consider the sequence consisting of n items of size k followed by $n \cdot k$ items of size 1. For the lower bound, note that if First-Fit rejects anything, it accepts at least n items, and no algorithm can accept more than $n \cdot k$ items. From that it follows that First-Fit's accommodating function drops down to $\frac{1}{k}$ for $\alpha \geq 2$. Moreover, it is $\Theta(\frac{1}{k})$ for $\alpha \geq 1 + c$, for any constant $c > 0$, by using $(\alpha - 1)n \cdot k$ (instead of $n \cdot k$) items of size 1.

3.2 Algorithm Log

Algorithm Log uses the standard "classify and randomly select" technique [2]. First, we describe a derandomized version of the algorithm, Dlog, using a number of bins which is at least the number of classes. The proof of the lower bound for Dlog is similar to a proof in [2]. In the description and analysis of the algorithm Dlog, we assume that $n \geq c \lceil \log_2 k \rceil$, for some constant $c > 1$.

Dlog divides the n bins into $\lceil \log_2 k \rceil$ groups $G_1, G_2, \ldots, G_{\lceil \log_2 k \rceil}$. Let $p = \lfloor \frac{n}{\lceil \log_2 k \rceil} \rfloor$ and let $s = n - p \cdot \lceil \log_2 k \rceil$. Groups $G_1, G_2, \ldots, G_s$ consist of $p + 1$ bins and the rest of the groups consist of p bins. Let $S_1 = \{x \mid \frac{k}{2} \leq x \leq k\}$, and $S_i = \{x \mid \frac{k}{2^i} \leq x < \frac{k}{2^{i-1}}\}$, for $2 \leq i \leq \lceil \log_2 k \rceil$. When Dlog receives an item o of size $s_o \in S_i$, it decides which group G_j of bins to pack it in by calculating $j = \max\{j \leq i \mid \text{there is a bin in } G_j \text{ that has room for } o\}$. If j exists, o is packed in G_j according to the First-Fit packing rule. If not, the item o is rejected.

THEOREM 1. *For every $\alpha \geq 1$, if $n \geq c \lceil \log_2 k \rceil$, for some constant $c > 1$, the competitive ratio of Dlog on α-sequences is $\Theta(\frac{1}{\log k})$.*

PROOF. Consider first the lower bound. For $i \in \{1, 2, \ldots, \lceil \log_2 k \rceil\}$, let $n_i(I)$ denote the number of items of size $s \in S_i$ accepted by OPT when given the sequence I of items. Since group G_i is reserved for items of size $\frac{k}{2^{i-1}}$ or smaller, the bins in group G_i will receive at least $\min\{2^{i-1}p, n_i(I)\}$ items. OPT can accept at most $2^i n$ items with sizes in S_i, i.e., $n_i(I) \leq 2^i n$. Thus, $2^{i-1}p > 2^{i-1}(\frac{n}{\lceil \log_2 k \rceil} - 1) \geq n_i(I)(\frac{1}{2\lceil \log_2 k \rceil} - \frac{1}{2n})$. Given the same sequence, Dlog packs at least $n_i(I)(\frac{1}{2\lceil \log_2 k \rceil} - \frac{1}{2n})$ items in G_i, for $i \in \{1, 2, \ldots, \lceil \log_2 k \rceil\}$. So, for any I,

$$\frac{\text{Dlog}(I)}{\text{OPT}(I)} > \frac{\sum\limits_{i \in \{1,2,\ldots,\lceil \log_2 k \rceil\}} n_i(I)(\frac{1}{2\lceil \log_2 k \rceil} - \frac{1}{2n})}{\sum\limits_{i \in \{1,2,\ldots,\lceil \log_2 k \rceil\}} n_i(I)} = \frac{1}{2\lceil \log_2 k \rceil} - \frac{1}{2n},$$

so $CR_{\text{Dlog}} > \frac{1}{2\lceil \log_2 k \rceil} - \frac{1}{2n}$.

For the upper bound, consider the accommodating sequence I with n items of size k. Then,

$$\frac{\text{Dlog}(I)}{\text{OPT}(I)} = \frac{\lceil \frac{n}{\lceil \log_2 k \rceil} \rceil}{n} < \frac{1}{\lceil \log_2 k \rceil} + \frac{1}{n},$$

so $AR_{\text{Dlog}} < \frac{1}{\lceil \log_2 k \rceil} + \frac{1}{n}$. $\Box$

If there is no constant $c > 1$ such that $n \geq c \lceil \log_2 k \rceil$, a randomized version of the algorithm Dlog can be used. Instead of dividing the bins into the groups $G_1, G_2, \ldots, G_{\lceil \log_2 k \rceil}$, we uniformly at random choose an index i among the $\lceil \log_2 k \rceil$ possibilities. All of the bins are assigned to group G_i and the other groups are empty. Items which would have been assigned to groups other than G_i are rejected. We call this randomized algorithm Log.

COROLLARY 1. *On accommodating sequences, the competitive ratio of Log is* $\Theta(\frac{1}{\log k})$.

3.3 A Hardness Result

In this section, we consider an arbitrary on-line algorithm $\mathbb{A}$ for On-Line Dual Bin Packing and prove general bounds on how well it can do. The proof of this general upper bound for the competitive ratio is analogous to the proof of the corresponding lemma in [1].

Clearly, if the input sequences are all accommodating, the algorithm Log does not have the best possible competitive ratio, but on unrestricted sequences, its competitive ratio is quite close to optimal, as shown by the following theorem.

THEOREM 2. *Any deterministic or randomized on-line algorithm for On-Line Dual Bin Packing has a competitive ratio of less than* $\frac{4}{\lfloor \log_2 k \rfloor}$.

PROOF. The items are given in phases numbered $0, 1, \ldots, r$, $r \le \lfloor \log_2 k \rfloor$. In Phase i, $n2^i$ items of size $2^{\lfloor \log_2 k \rfloor - i}$ are given. Clearly, any optimal off-line algorithm will accept all $n2^r$ items in Phase r.

Let x_i be the expected number of items that the on-line algorithm accepts in Phase i, $0 \le i \le r$, and $x_i = 0$, $r < i \le \lfloor \log_2 k \rfloor$. By linearity of expectations, the expected total number of items accepted by the on-line algorithm is $\sum_{i=0}^{\lfloor \log_2 k \rfloor} x_i$ and the expected total size of the items accepted is $\sum_{i=0}^{\lfloor \log_2 k \rfloor} 2^{\lfloor \log_2 k \rfloor - i} x_i > \sum_{i=0}^{\lfloor \log_2 k \rfloor} \frac{k}{2} 2^{-i} x_i$. Since there are only nk units of capacity overall, we get: $\sum_{i=0}^{\lfloor \log_2 k \rfloor} \frac{k}{2} 2^{-i} x_i \le nk$, or $\sum_{i=0}^{\lfloor \log_2 k \rfloor} 2^{-i} x_i \le 2n$.

We now show that r can be chosen such that $\sum_{i=0}^{r} x_i < \frac{4 \cdot n2^r}{\lfloor \log_2 k \rfloor}$, meaning that OPT will pack more than $\frac{1}{4}\lfloor \log_2 k \rfloor$ times as many items as the on-line algorithm. Defining $S_j = 2^{-j} \sum_{i=0}^{j} x_i$, this statement can be reformulated as $\exists r \in \{0, 1, \ldots, \lfloor \log_2 k \rfloor\} : S_r < \frac{4n}{\lfloor \log_2 k \rfloor}$, which is proven by the following inequality. $\displaystyle\sum_{j=0}^{\lfloor \log_2 k \rfloor} S_j \;=\; \sum_{0 \le i \le j \le \lfloor \log_2 k \rfloor} 2^{-j} x_i \;<\; \sum_{i=0}^{\lfloor \log_2 k \rfloor} 2 \cdot 2^{-i} x_i \;\le\; 4n.$ □

4. Knowing the Minimum Resources Necessary

Suppose that, for each sequence I of items, the on-line algorithm knows, beforehand, a good upper bound αn on the number of bins needed to pack the items in I. Then even for $\alpha > 1$, there exist simple algorithms (both deterministic and randomized) with a constant (that is, independent of k and n) accommodating function when evaluated at this α, as long as α is a constant. Note that for $\alpha < 1$, the existence of such algorithms follows trivially from the constant competitive ratio of First-Fit on accommodating sequences [7].

4.1 A Randomized Algorithm

In this section we describe an algorithm using the standard "classify and randomly select" technique [2]. One way of exploiting the knowledge of an upper bound on αn is to use αn "virtual" bins. At the beginning, the randomized algorithm $\mathbb{R}$ chooses uniformly at random which n of the αn virtual bins are going to correspond to the "real" n bins. Call the set of these n virtual bins B_A and the rest of the αn virtual bins B_R. An algorithm $\mathbb{A}$ with a "good" competitive ratio on accommodating sequences $AR_{\mathbb{A}}$ is used to decide where the actual items would be packed in the αn virtual bins.

When $\mathbb{A}$ packs an item in a bin in B_A, the algorithm $\mathbb{R}$ accepts the item and places it in the corresponding real bin. All other items are rejected.

The expected fraction of the items which $\mathbb{R}$ accepts is at least $\frac{AR_{\mathbb{A}}}{\alpha}$, since on average $\frac{|B_A|}{|B_A|+|B_R|} = \frac{n}{\alpha n} = \frac{1}{\alpha}$ of the items accepted by $\mathbb{A}$ will be packed in B_A. Using Unfair-First-Fit which was shown to have an asymptotically competitive ratio of $\frac{2}{3}$ an accommodating sequences in [4], this gives $\mathcal{A}(\alpha) \geq \frac{2}{3\alpha}$ (asymptotically), which is constant when α is.

Another way of using virtual bins is to use an algorithm that is known to be able to pack any accommodating sequence of items in βn bins for some constant β. In this case, $\alpha\beta n$ virtual bins are used. The algorithm with the best known value of β is Harmonic++ [16]. In [16], it is shown that when n goes to infinity, β goes to a value that is at most 1.58889. Furthermore, it is proven in [17] that no on-line algorithm can have a β smaller than 1.54014. Thus, using this approach, $\mathcal{A}(\alpha) \leq \frac{1}{1.54014\alpha} \approx \frac{0.649}{\alpha}$, which is a little lower (worse) than for the method described above using Unfair-First-Fit to pack items in αn virtual bins.

Amos Fiat [13] has noted that the technique described above can be used more generally, for many maximization problems, to give good values for the accommodating function when $\alpha \geq 1$ is small. If an algorithm $\mathbb{A}$ with competitive ratio on accommodating sequences $AR_{\mathbb{A}}$ is used with a quantity αn of the virtual resource, and a quantity n of these virtual resources are randomly chosen and used on the real resources, then the algorithm will achieve an accommodating function of $\mathcal{A}(\alpha) \geq \frac{AR_{\mathbb{A}}}{\alpha}$.

4.2 A Deterministic Algorithm

It is also possible for a deterministic algorithm to have an accommodating function such that the function value of the accommodating function is constant (that is, independent of k and n) when evaluated at a constant α as long as $n \geq 5$. The following algorithm $\mathbb{D}$ has this property.

$\mathbb{D}$ divides the possible item sizes into $\lceil \log_2 k \rceil$ intervals, $S_1, S_2, \ldots, S_{\lceil \log_2 k \rceil}$, defined by $S_1 = \{x \mid \frac{k}{2} \leq x \leq k\}$, and $S_i = \{x \mid \frac{k}{2^i} \leq x < \frac{k}{2^{i-1}}\}$, for $2 \leq i \leq \lceil \log_2 k \rceil$. Thus, for any two items with sizes s_a and s_b belonging to the same size interval, $s_a \geq \frac{1}{2} s_b$.

For each i, $1 \leq i \leq \lceil \log_2 k \rceil$, $\mathbb{D}$ does the following. It accepts the first item with size $s \in S_i$. After that it accepts every $\frac{\alpha}{\beta}$th item with size $s \in S_i$, for a given constant β, and rejects all other items with sizes in S_i. The accepted items are packed according to the First-Fit packing rule and the constant β will be chosen as described below, so that $\mathbb{D}$ has no problem doing so. Since $\mathbb{D}$ accepts every $\frac{\alpha}{\beta}$th item in each size interval, $\mathcal{A}(\alpha) \geq \frac{\beta}{\alpha}$.

Let O be the set of all the items given, let O_F be the set of items consisting of the first item in each size interval and let $O' = O \setminus O_F$. Let A be the set of items accepted by $\mathbb{D}$ and let $A' = A \setminus O_F$. For any set S of items, let $V(S)$, denote the sum of the sizes of the items in S.

LEMMA 1. *Let m be the number of bins containing at least c items in a First-Fit packing. If $c \geq 1$ and $m \geq c+1$, then the total size V of the items in these m bins is more than $\frac{c}{c+1}mk$.*

PROOF. Let C denote the set of bins containing at least c items, and, for any bin b, let $V(b)$ denote the total size of the items in b.

Suppose, for the sake of contradiction, that $V \leq \frac{c}{c+1}mk$. Then there is a bin $b \in C$ such that $V(b) = \frac{c}{c+1}k - \varepsilon$, $\varepsilon \geq 0$. The size of any item placed in a bin to the right of b must be greater than $\frac{1}{c+1}k + \varepsilon$, since otherwise it would fit in b. Therefore any bin $b' \in C$ to the right of b has $V(b') > \frac{c}{c+1}k + c\varepsilon \geq \frac{c}{c+1}k$. This means that there is only one bin $b \in C$ with $V(b) \leq \frac{c}{c+1}k$, and if b is not the rightmost nonempty bin in C, then $V > (m-2)\frac{c}{c+1}k + (\frac{c}{c+1}k - \varepsilon) + (\frac{c}{c+1}k + c\varepsilon) \geq m\frac{c}{c+1}k$. Thus, b must be the rightmost nonempty bin in C.

One of the items in b must have size at most $\frac{1}{c+1}k - \frac{\varepsilon}{c}$. Since this item was not placed in one of the $m-1$ bins to the left of b, these must all be filled to more than $\frac{c}{c+1}k + \frac{\varepsilon}{c}$. Thus, $V > (m-1)(\frac{c}{c+1}k + \frac{\varepsilon}{c}) + (\frac{c}{c+1}k - \varepsilon) = m\frac{c}{c+1}k + (m-1)\frac{\varepsilon}{c} - \varepsilon \geq m\frac{c}{c+1}k + c\frac{\varepsilon}{c} - \varepsilon = m\frac{c}{c+1}k$, which is a contradiction. □

It follows from Lemma 1 that the total size of the items in any First-Fit packing using n bins is more than $\frac{nk}{2}$. Thus, if β is chosen such that $V(A) \leq \frac{nk}{2}$, $\mathbb{D}$ will be able to pack all the accepted items.

To determine an appropriate value for β, first notice that $V(O') \leq V(O) \leq \alpha nk$, since all the items can fit in αn bins, and $V(O') > \frac{1}{2}\frac{\alpha}{\beta}V(A')$, since for every item $o \in A'$, $\frac{\alpha}{\beta} - 1$ items, each of size $s \geq \frac{1}{2}\text{size}(o)$, have been rejected. Combining these inequalities gives $\frac{1}{2}\frac{\alpha}{\beta}V(A') < \alpha nk$, and solving for $V(A')$ yields $V(A') < 2\beta nk$.

Furthermore, $V(O_F) \leq \displaystyle\sum_{i=0}^{\lceil \log_2 k \rceil - 1} \frac{k}{2^i} < \sum_{i=0}^{\infty} \frac{k}{2^i} = 2k.$

We now have that $V(A) = V(A') + V(O_F) < 2\beta nk + 2k$. To obtain $2\beta nk + 2k \leq \frac{nk}{2}$, n must be at least 5, for any $\beta > 0$. For $n \geq 5$, $\beta = \frac{1}{20}$ assures that $V(A) \leq \frac{nk}{2}$. If we accept that n must be at least 10, then $\beta = \frac{3}{20}$ can be used. Thus, if $n \geq 5$, $\mathcal{A}(\alpha) \geq \frac{1}{20\alpha}$, and if $n \geq 10$, $\mathcal{A}(\alpha) \geq \frac{3}{20\alpha}$.

Acknowledgements

We would like to thank Yossi Azar for discussions which led to this work.

References

[1] Awerbuch, B., Azar, Y., and Plotkin, S. 1993. Throughput-Competitive On-Line Routing. In *Proceedings of the 34th IEEE Symposium on Foundations of Computer Science*, 32–40.

[2] Awerbuch, B., Bartal, Y., Fiat, A., and Rosén, A. 1994. Competitive Non-Preemptive Call Control. In *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms*, 312–320.

[3] Awerbuch, B., Gawlick, R., Leighton, T., and Rabani, Y. 1994. On-line Admission Control and Circuit Routing for High Performance Computation and Communication. In *Proceedings of the 35th IEEE Symposium on Foundations of Computer Science*, 412–423.

[4] Azar, Y., Boyar, J., Epstein, L., Favrholdt, L. M., Larsen, K. S., and Nielsen, M. N.. Fair versus Unrestricted Bin Packing. *Algorithmica* , to appear.

[5] Azar, Y., Boyar, J., Favrholdt, L. M., Larsen, K. S., and Nielsen, M. N. 2000. Fair versus Unrestricted Bin Packing. In *Proceedings of the 7th Scandinavian Workshop on Algorithm Theory*, Volume 1851 of *Lecture Notes in Computer Science*. Springer-Verlag, 200–213.

[6] Borodin, A. and El-Yaniv, R. 1998. *Online Computation and Competitive Analysis*. Cambridge University Press.

[7] Boyar, J., Larsen, K. S., and Nielsen, M. N. 2001. The Accommodating Function: A Generalization of the Competitive Ratio. *SIAM Journal on Computing 31*, 1, 233–258.

[8] Bruno, J. L. and Downey, P. J. 1985. Probabilistic Bounds for Dual Bin-Packing. *Acta Informatica 22*, 333–345.

[9] Coffman Jr., E. G., Garey, M. R., and Johnson, D. S. 1997. Approximation Algorithms for Bin Packing: A Survey. In *Approximation Algorithms for NP-Hard Problems*, Hochbaum, Dorit S., Editor. PWS Publishing Company, chapter 2, 46–93.

[10] Coffman Jr., E. G., Leung, J. Y-T., and Ting, D. W. 1978. Bin Packing: Maximizing the Number of Pieces Packed. *Acta Informatica 9*, 263–271.

[11] Coffman Jr., E. G. and Leung, Joseph Y-T. 1979. Combinatorial Analysis of an Efficient Algorithm for Processor and Storage Allocation. *SIAM Journal on Computing 8*, 202–217.

[12] Csirik, J. and Woeginger, G. 1998. On-Line Packing and Covering Problems. In *Online Algorithms*, Amos Fiat, Gerhard J. Woeginger, Editor. Volume 1442 of *Lecture Notes in Computer Science*. Springer-Verlag, chapter 7, 147–177.

[13] Fiat, A. 1999. Personal communication.

[14] Hochbaum, D. S., Editor. 1997. *Approximation Algorithms for NP-Hard Problems*. PWS Publishing Company.

[15] Leung, J. Y. 1977. *Fast Algorithms for Packing Problems*. PhD thesis, Pennsylvania State University.

[16] Seiden, S. S. 2001. On the Online Bin Packing Problem. In *Proceedings of the 28th International Colloquium on Automata, Languages and Programming*, Volume 2076 of *Lecture Notes in Computer Science*. Springer-Verlag, 237–248.

[17] van Vliet, A. 1992. An Improved Lower Bound for Online Bin Packing Algorithms. *Information Processing Letters 43*, 277–284.