Fair versus Unrestricted Bin Packing

Yossi Azar[1*], Joan Boyar[2**], Lene M. Favrholdt[2], Kim S. Larsen[2**], and Morten N. Nielsen[2]

[1] Department of Computer Science, Tel-Aviv University, azar@math.tau.ac.il
[2] Department of Mathematics and Computer Science, University of Southern Denmark, Odense, {joan,lenem,kslarsen,nyhave}@imada.sdu.dk.

Abstract. We consider the Unrestricted Bin Packing problem where we have 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. We investigate the power of performing admission control on the items, i.e., rejecting items while there is enough space to pack them, versus behaving fairly, i.e., rejecting an item only when there is not enough space to pack it. We show that by performing admission control on the items, we get better performance for various measures compared with the performance achieved on the fair version of the problem. Our main result shows that we can pack 2/3 of the items for sequences in which the optimal can pack all the items.

1 Introduction

1.1 General

In this paper, we are investigating the competitive ratio for a bin packing problem. However, in addition to considering unrestricted request sequences, we also consider some restricted sequences which we refer to as accommodating sequences. Informally, these are sequences where an optimal algorithm can satisfy all requests. Clearly, the competitive ratio on accommodating sequences[1] is no worse than the competitive ratio on unrestricted sequences for any given problem and sometimes can be much better. For problems where the competitive ratio is a bad measure, it may be useful to compare algorithms by their competitive ratio on accommodating sequences. Specifically, it was shown in [4, 5] that there are (benefit) problems where the competitive ratio tends to zero while the competitive ratio on accommodating sequences is a constant, i.e., independent of the parameters of the problem. Moreover, when we are trying to distinguish between two algorithms, the competitive ratio on accommodating sequences may prefer one algorithm while the competitive ratio measure (on all sequences) prefers the other [5].

* Supported in part by the Israel Science Foundation, and by a USA-Israel BSF grant.
** Supported in part by the Danish Natural Science Research Council (SNF).

[1] In earlier papers [4–6], this competitive ratio on accommodating sequences was called the accommodating ratio. The change is made here for consistency with common practice in the field.

In the *Bin Packing problem* we are given some bins and the goal is to pack a set of items into these bins. We concentrate on the benefit variant of the problem, where there are n bins and the objective is to maximize the total number of items in these bins. This problem has been studied in the off-line setting, starting in [8], and its applicability to processor and storage allocation is discussed in [9]. (For surveys on bin packing, see [10, 7].)

In the on-line version of the problem the items arrive in some sequence and the assignment of an item should be done before the next item arrives. We assume that the items are integer-sized and the bins all have size k. One can discuss the Fair Bin Packing problem[2] where it is required that the packing be *fair*, that is, an item can only be rejected if it cannot fit in any bin at the time when it is given. Note that the optimal algorithm is also required to be fair. It is shown in [5] that for this problem, Worst-Fit has a strictly better competitive ratio than First-Fit, while First-Fit has a strictly better competitive ratio than Worst-Fit on accommodating sequences. In this case, the competitive ratio on accommodating sequences seems the more appropriate measure, since it is constant while the competitive ratio (on all sequences) is close to zero, for large values of k, basically due to some sequences which seem very contrived. This demonstrated the usefulness of the more general accommodating function [6] which comprises the competitive ratio as well as the competitive ratio on accommodating sequences (it is a function of the restriction on the request sequences).

Here, we consider what happens when the fairness restriction is removed. Thus, for the on-line problem *Unrestricted Bin Packing (UBP)*, there are again n bins, all of size k, the items are integer-sized, and the goal is to maximize the total number of items placed in the bins, but there is no fairness restriction.

We note that on accommodating sequences, the competitive ratio of UBP is no worse than the competitive ratio of the fair problem, since the optimal algorithm serves all the requests and hence is fair. In general, however, the competitive ratio of UBP is not necessarily better than the competitive ratio of the fair problem since the optimal algorithms may be different. In fact, in many cases, considering unfair algorithms, i.e., performing admission control on the requests, is the more challenging problem; see for example the results for throughput routing in [1–3]. In particular, with the Unrestricted Bin Packing problem, it is easier to differentiate between algorithms since both their competitive ratio and their competitive ratio on accommodating sequences can vary over a large range. This is in contrast to on-line algorithms for Fair Bin Packing where all of them must have both within a constant factor of each other.

1.2 Accommodating Sequences and the Accommodating Function

For completeness, we define the competitive ratio and the accommodating function for Unrestricted Bin Packing. Note that Unrestricted Bin Packing is a maximization problem, and all ratios are less than or equal to 1.

[2] In [6] where some of the results from [5] were first presented in a preliminary form, this problem was called Unit Price Bin Packing.

Let $\mathbb{A}(I)$ denote the number of items algorithm $\mathbb{A}$ accepts when given request sequence I and let $\mathrm{OPT}(I)$ denote the number an optimal off-line algorithm, OPT, accepts. An on-line algorithm, $\mathbb{A}$, is *c-competitive* if there exists a constant b, such that $\mathbb{A}(I) \geq c \cdot \mathrm{OPT}(I) - b$ for all sequences I. The *competitive ratio* $CR = \sup\{c \mid \mathbb{A} \text{ is } c\text{-competitive}\}$.

Next, we introduce the restricted request sequences. We say that I is an *α-sequence*, if I could be packed in αn bins. We investigate the competitive ratio on such restricted sequences. To be precise, an on-line algorithm $\mathbb{A}$ is *c-competitive on α-sequences* if $c \leq 1$ and there exists a constant b, such that for every α-sequence I, $\mathbb{A}(I) \geq c \cdot OPT(I) - b$. The *accommodating function $\mathcal{A}$* is defined as $\mathcal{A}(\alpha) = \sup\{c \mid \mathbb{A} \text{ is } c\text{-competitive on } \alpha\text{-sequences}\}$.

Thus, the accommodating function for an algorithm is the competitive ratio of that algorithm on α-sequences as a function of α. We refer to 1-squences as *accommodating sequences*, since the optimal algorithm can accommodate all requests in such a sequence. We use AR to denote the competitive ratio on accommodating sequences.

1.3 Results

We prove results on the Unrestricted Bin Packing problem for the usual competitive ratio, the competitive ratio on accommodating sequences and the accommodating function. We start with the competitive ratio.

For the usual competitive ratio we prove the following:

- The algorithm Log (Section 2.2) has a competitive ratio of $\Theta(\frac{1}{\log k})$.
- No on-line algorithm can have a competitive ratio which is better than $O(\frac{1}{\log k})$, even when considering randomized algorithms.
- We observe that the competitive ratios of First-Fit and Worst-Fit are $\frac{1}{k}$.

These results should be compared with the competitive ratio of any on-line algorithm for the fair problem: they are all $\Theta(\frac{1}{k})$ [5].

For the competitive ratio on accommodating sequences we prove:

- The competitive ratio of Log on accommodating sequences is $\Theta(\frac{1}{\log k})$.
- We conclude from [5] that the competitive ratio of First-Fit on accommodating sequences is between $\frac{5}{8}$ and $\frac{7}{11}$, since the fairness restriction on OPT is irrelevant when all of the items can be packed.
- We design an unrestricted algorithm, Unfair-First-Fit, whose competitive ratio on accommodating sequences is $\frac{2}{3}$, which is strictly higher than the competitive ratio of First-Fit on accommodating sequences.
- The competitive ratio of any on-line algorithm on accommodating sequences is no better than $\frac{6}{7}$, even when considering randomized algorithms.

Thus, according to the usual competitive ratio, Log is the better algorithm, and according to the competitive ratio on accommodating sequences, First-Fit is the better algorithm (the same is true for Log and Unfair-First-Fit).

For the accommodating function we prove the following:

– We design randomized and deterministic algorithms for which the accommodating function evaluated at any constant α is a constant, if the algorithm is given the value α.
– In contrast, we observe that First-Fit's (and Unfair-First-Fit's) accommodating function drops down to $\Theta(\frac{1}{k})$ for $\alpha \geq 1+c$, for any constant $c > 0$.

The main technical effort is to prove the competitive ratio of the algorithm Unfair-First-Fit on accommodating sequences. The other results are easier to prove. Algorithm Log uses derandomization of the standard classify and select technique. The proof of the lower bound for Log is similar to the lower bound proof in [2], and the proof of the general upper bound for the competitive ratio is analogous to the proof of the corresponding lemma in [1].

Remark: In this paper, we assume that all items are integer-sized and the bins have size k. All of the results hold with the weaker assumption that the bins are unit-sized and the smallest item has size at least $\frac{1}{k}$. However, some of the results in [5] do not appear to hold with this assumption, so we use the stronger assumption for consistency.

2 The Competitive Ratio

2.1 First-Fit and Worst-Fit

It is easy to see that the competitive ratio of First-Fit or Worst-Fit for Unrestricted 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 (or Worst-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 (and Worst-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.

2.2 Algorithm Log

In the description of the algorithm Log, we assume that $n > c\lceil \log_2 k \rceil$, for some constant $c > 1$. If n is smaller, we can use simple randomization to achieve the same results.

Log 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 Log 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. *The competitive ratio of Log is $\Theta(\frac{1}{\log k})$, even on accommodating sequences.*

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, Log 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,

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

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

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

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

so $AR_{\mathrm{Log}} < \frac{1}{\lceil \log_2 k \rceil} + \frac{1}{n}$. Since all sequences are considered for the competitive ratio, $CR_{\mathrm{Log}} \leq AR_{\mathrm{Log}}$, and the result follows. □

2.3 An Upper Bound on the Competitive Ratio

In this section, we consider an arbitrary on-line algorithm $\mathbb{A}$ for Unrestricted Bin Packing and prove general bounds on how well it can do. First, note that the only possible lower bound on the competitive ratio, even on accommodating sequences, is zero, since for the algorithm which simply rejects everything, the ratio is equal to zero.

Clearly, the algorithm Log does not have the best possible competitive ratio on accommodating sequences, but its competitive ratio is quite close to optimal.

Theorem 2. *Any deterministic or randomized algorithm for Unrestricted Bin Packing has a competitive ratio of less than* $\frac{2}{\log_2 k}$.

Proof. Assume that k is a power of 2. The items are given in phases numbered $0, 1, \ldots, r$, $r \leq \log_2 k$. In phase i, $n2^i$ items of size $k/2^i$ 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, $0 \leq i \leq r$, and $x_i = 0$, $r < i \leq \log_2 k$. By the linearity of expectations, the expected total number of items accepted by the on-line algorithm is $\sum_{i=0}^{\log_2 k} x_i$ and the expected total volume of the items accepted is $\sum_{i=0}^{\log_2 k} k2^{-i}x_i$. Since there are only nk units of capacity overall, we get: $\sum_{i=0}^{\log_2 k} k2^{-i}x_i \leq nk$, or $\sum_{i=0}^{\log_2 k} 2^{-i}x_i \leq n$.

We now show that r can be chosen such that $\sum_{i=0}^{r} x_i < \frac{2 \cdot n 2^r}{\log_2 k}$, meaning that OPT will pack more than $\frac{1}{2}\log_2 k$ 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 $\exists r \in \{0, 1, \ldots, \log_2 k\} : S_r < \frac{2n}{\log_2 k}$, which is proven by the following inequality.

$$\sum_{j=0}^{\log_2 k} S_j = \sum_{0 \le i \le j \le \log_2 k} 2^{-j} x_i < \sum_{i=0}^{\log_2 k} 2 \cdot 2^{-i} x_i \le 2n. \qquad \square$$

3 The Competitive Ratio on Accommodating Sequences

3.1 An Upper Bound

Now we turn to the competitive ratio on accommodating sequences. In [5], it was shown that for $k \geq 7$, any deterministic Fair Bin Packing algorithm has a competitive ratio on accommodating sequences of at most $\frac{6}{7}$. The same result and essentially the same proof hold when the fairness restriction is removed, even for randomized algorithms.

Theorem 3. *For $k \geq 7$, any deterministic or randomized Unrestricted Bin Packing algorithm has a competitive ratio of at most $\frac{6}{7}$, even on accommodating sequences.*

Proof. Assume n is even. Consider an arbitrary on-line algorithm $\mathbb{A}$. An adversary can proceed as follows: Give n items of size $\lceil \frac{k}{2} \rceil - 1$, and let q denote the number of bins which contain two items after this. In the case where $E[q] < \frac{2n}{7}$, the adversary gives $\frac{n}{2}$ long requests of size k. The off-line algorithm can pack the first n requests in the first $\frac{n}{2}$ bins and thus accept all $\frac{3n}{2}$ items. On average, the on-line algorithm places two items in $E[q]$ bins and has at most one item in every other bin. The performance ratio is thus at most $\frac{E[n+q]}{n+\frac{n}{2}} = \frac{2n+2q}{3n} < \frac{6}{7}$.

In the case where $E[q] \geq \frac{2n}{7}$, the adversary gives n requests of size $\lfloor \frac{k}{2} \rfloor + 1$. The off-line algorithm can pack the first n items one per bin and thus accept all $2n$ items. The on-line algorithm must reject at least $E[q]$ items on average. The performance ratio is thus at most $\frac{E[2n-q]}{2n} \leq \frac{6}{7}$. $\square$

3.2 Unfair-First-Fit

The Algorithm. In Section 2.2, it was shown that there is an algorithm for Unrestricted Bin Packing which has a better competitive ratio than any algorithm for Fair Bin Packing. It would be difficult to do the same for the competitive ratio on accommodating sequences, since the best upper bound known is $\frac{6}{7}$ for both problems. First-Fit's competitive ratio on accommodating sequences is known to lie between $\frac{5}{8}$ and $\frac{7}{11}$ [6], and no algorithm for Fair Bin Packing is known to have a better competitive ratio on accommodating sequences. The algorithm Unfair-First-Fit (UFF), presented below, is shown to have a competitive ratio on accommodating sequences which is better than that of First-Fit as long as the number of bins is at least 22; the ratio approaches $\frac{2}{3}$ as n increases. What makes Unfair-First-Fit different from First-Fit is that items larger than $\frac{k}{2}$ are rejected if enough items have been accepted already to maintain the desired ratio of $\frac{2}{3}$.

Input: $S = \langle o_1, o_2, \ldots, o_n \rangle$
Output: A, R, and a packing for those items in A
$A := \{o_1\}$; $R := \{\}$; $S := \text{tail}(S)$
while $S \neq \langle\rangle$
    $o := \text{hd}(S); S := \text{tail}(S)$
    if size$(o) > \frac{k}{2}$ **and** $\frac{|A|}{|A|+|R|+1} \geq \frac{2}{3}$
        $R := R \cup \{o\}$
    else if there is space for o in some bin
            place o according to the First-Fit rule
            $A := A \cup \{o\}$
        else
            $R := R \cup \{o\}$

The Competitive Ratio on Accommodating Sequences.

Theorem 4. *For $n \geq 9$, the competitive ratio of Unfair-First-Fit on accommodating sequences is more than $\dfrac{2}{3} - \dfrac{4}{6n+3}$. Thus, for $n \geq 22$, $AR_{UFF} > AR_{FF}$.*

Proof. The term "large" is used for items strictly larger than $\frac{k}{2}$, since they are considered in a special way by the algorithm. Let B denote the set of large items that are alone in a bin in UFF's packing. Let s denote the size of the smallest item in R. We divide the proof into two cases depending on the size of s. The first case is easy.

Case 1: $s > \frac{k}{2}$: Since the smallest item in R is larger than $\frac{k}{2}$, the items in $R \cup B$ are all larger than $\frac{k}{2}$. Thus, since all items can be packed in n bins, $|R| + |B| \leq n$, or $|R| \leq n - |B|$. Furthermore, at most one small item can be alone in a bin: $|A| \geq 2n - |B| - 1$. Thus, the performance ratio is

$$\frac{|A|}{|A|+|R|} \geq \frac{2n-|B|-1}{2n-|B|-1+n-|B|} \geq \frac{2n-1}{3n-1} = \frac{2}{3} - \frac{1}{9n-3}.$$

Case 2: $s \leq \frac{k}{2}$: Since we consider the competitive ratio on accommodating sequences, an optimal off-line algorithm, OPT, can pack all items in S. It may be instructive to view the optimal packing as being done in 3 phases:

1. UFF is run on S.
2. The packed items are rearranged, creating room for the rejected items.
3. The rejected items are packed.

The packing after Phase 1 is denoted by P_{UFF}, and the packing after Phase 3 is denoted by P_{OPT}. Similarly, E_{UFF} and E_{OPT} are used to denote the total empty space after Phase 1 and Phase 3 respectively. We assume without loss of generality that no large item is moved during Phase 2.

We divide the rejected items into two disjoint sets: R_b which contains large items, and R_s which contains small items. We use the following equation to bound the number of small items rejected.

$$|R_s| \leq \frac{1}{s} \cdot \left(E_{\text{UFF}} - E_{\text{OPT}} - \frac{k}{2}|R_b| \right)$$

It is easy to see that $|R| < n$, since the empty space in any bin in P_{UFF} is less than s and all rejected items have size at least s. Thus, if all bins contain at least two items each, $\frac{|A|}{|A|+|R|} > \frac{2n}{2n+n} = \frac{2}{3}$ and we are through. Therefore, assume that some bins contain only one item. Since the empty space in any bin is less than $\frac{k}{2}$, such items must be large. Thus, the items that are alone in a bin are exactly the items in B.

It is now clear that $|A| \geq 2n - |B|$. However, if some bins contain more than two items, this lower bound is too pessimistic. Therefore, we try to "spread out" the items a little more. Assume that the items in P_{UFF} are labeled with consecutive numbers in each bin according to their arrival time, i.e., the first item in a bin is labeled 1, the next one is labeled 2, and so on. We split Phase 2 into two Subphases, 2A and 2B, such that in Subphase 2A only items with labels higher than 2 are moved and in Subphase 2B the remaining moves are performed. Note that the packing produced during Subphase 2A is only technical and used for counting purposes; it might be illegal in that some bins might contain a total volume larger than k.

If some of the items moved during Subphase 2A are moved to bins containing items from B, a better lower bound on $|A|$ can now be obtained (Lemma 1). The set of items that are still alone after Subphase 2A is divided into two sets: X, containing the items that are still alone after Subphase 2B, and L, containing those that are not. Any item that is alone after Subphase 2A was alone in P_{UFF} as well. Since no such item can be combined with an item belonging to R, each item in X is also alone in P_{OPT}. Therefore, the bins containing an item from X do not contribute to $E_{\text{UFF}} - E_{\text{OPT}}$.

Lemma 1. $|A| \geq 2n - |L| - |X|$.

Proof. $L \cup X$ is the set of objects that are alone after Subphase 2A. □

The following easy lemma is used to prove Lemma 3 below which, loosely speaking, shows that if we cannot guarantee that most of the bins contain at least two items after Subphase 2A, then much of the empty space in P_{UFF} is used by large rejected items.

Let t denote the time just after the last large item was accepted by UFF and let A_t denote the set of items accepted at time t.

Lemma 2. $|R_b| \geq \frac{1}{2}|A_t| - 1$.

Proof. Since a large item was accepted just before time t, all items previously rejected are large items and therefore contained in R_b. Since the item was accepted, $\frac{|A_t|-1}{|A_t|-1+|R_b|+1} < \frac{2}{3}$. Solving for $|R_b|$, we get $|R_b| > \frac{1}{2}|A_t| - \frac{3}{2}$, and since $|R_b|$ must be integer, we get $|R_b| \geq \frac{1}{2}|A_t| - 1$. □

Assume that at time t all small items accepted by UFF are marked.

Lemma 3. $|R_b| \geq |L| + \frac{1}{2}|X| - 1$.

Proof. It is shown that $|A_t| \geq 2|L| + |X|$, which will complete the proof, since, by Lemma 2, $|R_b| \geq \frac{1}{2}|A_t| - 1$. To each item $o \in L$, a marked item is assigned in the following way. Since no item in L is alone after Phase 2, we can assume that the bin b_o containing o will receive at least one item, o', labeled 1 or 2 during Phase 2. If o' is marked, it is assigned to o. Otherwise, it must be labeled 2, since all items labeled 1 in bins before b_o are marked. The item which was packed below o' in P_{UFF} was alone at time t. Therefore, this item is not moved to any item in L. This item (labeled 1) can be assigned to o. In this way, every item in L has an item assigned which arrived before time t and which is not in $L \cup X$. Since $L \cup X \subseteq A_t$, $|A_t| \geq 2|L| + |X|$. □

Subcase 2a: $s \leq \frac{k}{3}$. Since the smallest item in R has size s, the empty space in each bin in P_{UFF} is smaller than s. Thus, we can use $s(n - |X|)$ as an upper bound on $E_{\text{UFF}} - E_{\text{OPT}}$:

$$\begin{aligned}|R_s| &\leq \frac{1}{s} \cdot \left(E_{\text{UFF}} - E_{\text{OPT}} - \frac{k}{2}|R_b|\right) < \frac{1}{s}\left(s(n - |X|) - \frac{k}{2}|R_b|\right) \\ &= n - |X| - \frac{k}{2s}|R_b| \leq \ n - |X| \ - \frac{3}{2}|R_b|.\end{aligned}$$

Now, using Lemma 3, we get

$$\begin{aligned}|R| &= |R_s| + |R_b| \leq n - |X| - \frac{1}{2}|R_b| \leq n - |X| - \frac{1}{2}\left(|L| + \frac{1}{2}|X| - 1\right) \\ &= n - \frac{5}{4}|X| - \frac{1}{2}|L| + \frac{1}{2}.\end{aligned}$$

Thus,

$$\begin{aligned}\frac{|A|}{|A| + |R|} &\geq \frac{2n - |L| - |X|}{2n - |L| - |X| + (n - \frac{5}{4}|X| - \frac{1}{2}|L| + \frac{1}{2})} \\ &\geq \frac{2n - (|L| + |X|) + \frac{1}{3}}{3n - \frac{3}{2}(|L| + |X|) + \frac{1}{2}} - \frac{\frac{1}{3}}{3n - \frac{3}{2}(|L| + |X|) + \frac{1}{2}} \\ &\geq \frac{2}{3} - \frac{2}{12n - 3},\end{aligned}$$

since $|L| + |X| \leq \frac{2}{3}(n + 1)$, which follows from the fact that the number of large items is at most n: $n \geq |R_b| + |L| + |X| \geq (|L| + \frac{1}{2}|X| - 1) + |L| + |X| \geq \frac{3}{2}(|L| + |X|) - 1$.

Subcase 2b: $\frac{k}{3} < s \leq \frac{k}{2}$. In this case, $s(n - |X|)$ is not a good bound on $E_{\mathrm{UFF}} - E_{\mathrm{OPT}}$, but we will show that even in this case, $E_{\mathrm{UFF}} - E_{\mathrm{OPT}}$ is "almost" bounded by $\frac{k}{3}(n - |X|)$, if $n \geq 9$ and $\frac{|A|}{|A|+|R|} < \frac{2}{3}$. Lemma 4 below is used for this purpose.

Lemma 4. *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 volume 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 sum of the sizes 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. □

Assuming $n \geq 9$, Lemma 4 combined with Lemma 5 below says that the average empty space in bins containing more than one item can be assumed to be at most $\frac{k}{3}$.

Lemma 5. *Assume that $n \geq 9$ and $s \leq \frac{k}{2}$. Then, in P_{UFF}, at least three bins contain two or more items.*

Proof. Assume for the sake of contradiction that fewer than three bins contain at least two items. Since $s \leq \frac{k}{2}$, no bin contains a single item of size at most $\frac{k}{2}$. Therefore, at least $n-2$ bins contain large items, which all arrived before time t, i.e., $A_t \geq n-2$. By Lemma 2, at least $\frac{1}{2}A_t - 1$ large items are rejected. Adding these up and noting that there can be at most n large items, we get $n - 2 + \frac{n-2}{2} - 1 \leq n$. Solving for n yields $n \leq 8$, which is a contradiction. □

Our goal is now, roughly speaking, to show that the average empty space in *all* n bins is bounded by approximately $\frac{k}{3}$. Number the bins from left to right, and let l be the number of the bin in which the last large item was placed. Let e denote the largest empty space in bins containing an item from B. In the proof of Lemma 7 we will show a lower bound on the number of bins to the right of l of approximately $\frac{|B|}{2}$. Each of these bins contains at least two items of size larger than e. Thus, even if $e > \frac{k}{3}$, the average empty space in the B-bins and the bins to the right of l will be bounded above by approximately $\left(|B|\,e + (k-2e)\frac{|B|}{2}\right) / \frac{3|B|}{2} = \frac{k|B|}{2} \cdot \frac{2}{3|B|} = \frac{k}{3}$. Lemma 4 combined with Lemma 6 below says that we can assume that the rest of the bins have an average empty space of at most $\frac{k}{3}$.

Lemma 6. *Assume that $n \geq 9$, $s \leq \frac{k}{2}$, $e \geq \frac{k}{3}$, and $\frac{|A|}{|A|+|R|} < \frac{2}{3}$. Then, in P_{UFF} at least three of the first l bins contain two or more items.*

Proof. We count the total number of items of size larger than e. Since $|A| \geq 2n - |B|$, more than $n - \frac{|B|}{2}$ items are rejected, because otherwise we have a performance ratio of $\frac{2}{3}$, which is a contradiction. After bin l, there are $n-l$ bins containing at least two items each. All of the rejected items and those in the last $n-l$ bins are larger than e and there are more than $n - \frac{|B|}{2} + 2(n-l)$ of them. Bins containing items from B cannot accept any of these items, and only two can be put together since $e \geq \frac{k}{3}$. Thus, $n - \frac{|B|}{2} + 2(n-l) \leq 2(n - |B|)$. Solving for l, we get $l \geq \frac{n}{2} + \frac{3}{4}|B|$. This shows that at least $\frac{n}{2} - \frac{|B|}{4}$ bins to the left of l contain two or more items. By Lemma 5, $|B| \leq n-3$. Thus, $\frac{n}{2} - \frac{|B|}{4} \geq \frac{n}{2} - \frac{n-3}{4} = \frac{n+3}{4} \geq 3$, since $n \geq 9$. □

Lemma 7. *Assume that $n \geq 9$, $s \leq \frac{k}{2}$, and $\frac{|A|}{|A|+|R|} < \frac{2}{3}$. Then, $E_{UFF} - E_{OPT} < (n - |X|)\frac{k}{3} + \frac{k}{2}$.*

Proof. In the case where $e \leq \frac{k}{3}$, we have an upper bound of $\frac{k}{3}$ on the average empty space in bins with one item as well as bins with more items. Thus, $E_{\mathrm{UFF}} - E_{\mathrm{OPT}} \leq (n - |X|)\frac{k}{3}$. Now, assume that $e > \frac{k}{3}$. First we show an upper bound on l. At time t no two bins can contain only one small item each. Therefore, $|A_t| \geq 2l - |B| - 1$. The total number of large items is $|R_b| + |B| \geq \frac{1}{2}|A_t| - 1 + |B| \geq l + \frac{|B|}{2} - \frac{3}{2}$. Since OPT must pack all these items in separate bins, we have $l + \frac{|B|}{2} - \frac{3}{2} \leq n$. Define $z \geq 0$ such that $n - l = z + \frac{|B|}{2} - \frac{3}{2}$. Since every bin after bin l has two items of size greater than e, we have the following upper bound on the empty space in these $n-l$ bins and the bins with an item from $B \setminus X$: $e(|B| - |X|) + (k-2e)(n-l) = e|B| - e|X| + (k-2e)(z + \frac{|B|}{2} - \frac{3}{2}) < e|B| - \frac{k}{3}|X| + (k-2e)\frac{|B|}{2} + (k-2e)(z - \frac{3}{2}) = \frac{k|B|}{2} - \frac{k}{3}|X| + (k-2e)(z - \frac{3}{2}) \leq \frac{k|B|}{2} - \frac{k}{3}|X| + (k-2e)z < \frac{k|B|}{2} - \frac{k}{3}|X| + \frac{k}{3}z$. Among the remaining bins, $l - |B| = n - z - \frac{3|B|}{2} + \frac{3}{2}$ bins do not contain an item from X. All of these bins have at least two items, and according to Lemma 6, enough of these bins exist for us to conclude, by Lemma 4, that the empty space is at most $\frac{k}{3}(n - z - \frac{3|B|}{2} + \frac{3}{2})$. The total empty space is then less than $\frac{k|B|}{2} - \frac{k}{3}|X| + \frac{k}{3}z + \frac{k}{3}(n - z - \frac{3|B|}{2} + \frac{3}{2}) = (n - |X| + \frac{3}{2})\frac{k}{3}$. □

Then, by Lemma 7, if $n \geq 9$,

$$
\begin{aligned}
|R_s| &\leq \frac{1}{s} \cdot \left(E_{\mathrm{UFF}} - E_{\mathrm{OPT}} - \frac{k}{2}|R_b| \right) < \frac{1}{s}\left(\frac{k}{3}(n - |X|) + \frac{k}{2} - \frac{k}{2}|R_b| \right) \\
&\leq n - |X| + \frac{3}{2} - \frac{3}{2}|R_b|.
\end{aligned}
$$

Using Lemma 3 as in Subcase 2a, we get

$$
|R| < n - |X| + \frac{3}{2} - \frac{1}{2}(|L| + \frac{1}{2}|X| - 1) = n - \frac{5}{4}|X| - \frac{1}{2}|L| + 2, \text{ for } n \geq 9.
$$

Thus,

$$\begin{aligned}\frac{|A|}{|A|+|R|} &\geq \frac{2n-|L|-|X|}{2n-|L|-\frac{5}{4}|X|-\frac{1}{2}+2}\\ &\geq \frac{2n-(|L|+|X|)+\frac{4}{3}}{3n-\frac{3}{2}(|L|+|X|)+2} - \frac{\frac{4}{3}}{3n-\frac{3}{2}(|L|+|X|)+2}\\ &= \frac{2}{3} - \frac{4}{6n+3}, \text{ for } n \geq 9.\end{aligned}$$

This bound is lower than the lower bounds obtained in Case 1 and Subcase 2a for all n. It is shown in [5] that $\frac{7}{11}$ is an upper bound on FF's competitive ratio on accommodating sequences. For $n \geq 22$, $\frac{2}{3} - \frac{4}{6n+3} > \frac{7}{11}$. Thus, for $n \geq 22$, UFF has a better competitive ratio than FF on accommodating sequences. □

Remark: It is easy to see that UFF's competitive ratio is $\frac{1}{k}$. If it is less than $\frac{2}{3}$, then R is nonempty, so at least n items are accepted. OPT can accept at most nk items, so the competitive ratio is at least $\frac{1}{k}$. For the upper bound, if $\frac{3}{2}n$ items of size k followed by nk items of size 1 are given, UFF will accept n items of size k, while OPT will accept all of the small ones, giving a ratio of $\frac{1}{k}$. Note that this means that $\mathcal{A}_{\mathrm{UFF}}(\alpha) = \frac{1}{k}$, for $\alpha \geq \frac{5}{2}$. Furthermore, if $2n$ items of size $\frac{k}{2}$ are given, followed by $(\alpha-1)nk$ items of size 1, UFF will accept $2n$ items of size $\frac{k}{2}$, while OPT can accept $2n + (\alpha-1)n(k-2)$ items, giving a ratio of $\frac{2}{2+(\alpha-1)(k-2)}$. Thus, for any constant $c > 0$, $\mathcal{A}_{\mathrm{UFF}}(\alpha) \in \Theta(\frac{1}{k})$, if $\alpha \geq 1 + c$.

4 The Accommodating Function

Suppose that, for each sequence I of items, the on-line algorithm knows, beforehand, the number αn of bins needed to pack the items in I (or a good upper bound on α). Then an accommodating function can be achieved for which the function value is constant (that is, independent of k and n) when evaluated at a constant α.

4.1 A Randomized Algorithm

One way of exploiting the extra knowledge is to use αn "virtual" bins. At the beginning the randomized algorithm $\mathbb{R}$ randomly decides 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, 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 1-sequence of items in βn bins for some constant β. In this case, $\alpha\beta n$ virtual bins are used. According to [7], for the algorithm Harmonic+1, $\beta \leq 1.588720$. Using Harmonic+1 for packing items in the virtual bins and randomly choosing the n bins for B_A gives $\mathcal{A}(\alpha) \geq \frac{1}{1.58872\alpha} \approx \frac{0.629}{\alpha}$. According to [7], even for randomized algorithms, $\beta \geq 1.536$. Since $\frac{1}{1.536} \approx 0.651$, this approach cannot give an accommodating function as good as the method described above using αn virtual bins can.

Remark: Amos Fiat [11] 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 α 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 the volume of S, denoted by $V(S)$, be the sum of the sizes of the items in S.

It follows from Lemma 4 that the volume 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 $s \geq \frac{1}{2}\mathrm{size}(o)$, 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, $V(O_F) \leq \sum_{i=0}^{\lceil \log_2 k \rceil - 1} \frac{k}{2^i} < \sum_{i=0}^{\infty} \frac{k}{2^i} = 2k$.

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 $n \geq 10$, $\mathcal{A}(\alpha) \geq \frac{3}{20\alpha}$.

References

1. B. Awerbuch, Y. Azar, and S. Plotkin. Throughput-Competitive On-Line Routing. In *34th IEEE Symposium on Foundations of Computer Science*, pages 32–40, 1993.
2. B. Awerbuch, Y. Bartal, A. Fiat, and A. Rosén. Competitive Non-Preemptive Call Control. In *Proc. 5th Annual ACM-SIAM Symp. on Discrete Algorithms*, pages 312–320, 1994.
3. B. Awerbuch, R. Gawlick, T. Leighton, and Y. Rabani. On-line Admission Control and Circuit Routing for High Performance Computation and Communication. In *35th IEEE Symposium on Foundations of Computer Science*, pages 412–423, 1994.
4. J. Boyar and K. S. Larsen. The Seat Reservation Problem. *Algorithmica*, 25:403–417, 1999.
5. J. Boyar, K. S. Larsen, and M. N. Nielsen. The Accommodating Function — A Generalization of the Competitive Ratio. Tech. report 24, Department of Mathematics and Computer Science, University of Southern Denmark, Main Campus: Odense University, 1998. Extended version submitted for journal publication 1999.
6. J. Boyar, K. S. Larsen, and M. N. Nielsen. The Accommodating Function — A Generalization of the Competitive Ratio. In *Sixth International Workshop on Algorithms and Data Structures*, volume 1663 of *Lecture Notes in Computer Science*, pages 74–79. Springer-Verlag, 1999.
7. E. G. Coffman, Jr., M. R. Garey, and D. S. Johnson. Approximation Algorithms for Bin Packing: A Survey. In Dorit S. Hochbaum, editor, *Approximation Algorithms for NP-Hard Problems*, chapter 2, pages 46–93. PWS Publishing Company, 1997.
8. E. G. Coffman, Jr., J. Y-T. Leung, and D. W. Ting. Bin Packing: Maximizing the Number of Pieces Packed. *Acta Informatica*, 9:263–271, 1978.
9. E. G. Coffman, Jr. and Joseph Y-T. Leung. Combinatorial Analysis of an Efficient Algorithm for Processor and Storage Allocation. *SIAM J. Comput.*, 8:202–217, 1979.
10. J. Csirik and G. Woeginger. On-Line Packing and Covering Problems. In Gerhard J. Woeginger Amos Fiat, editor, *Online Algorithms*, volume 1442 of *Lecture Notes in Computer Science*, chapter 7, pages 147–177. Springer-Verlag, 1998.
11. A. Fiat. Personal communication, 1999.