Electronic Notes in Theoretical Computer Science 50 No. 1 (2001) – Proc. ATMOS 2001
URL: http://www.elsevier.nl/locate/entcs/volume50.html 15 pages

Seat Reservation Allowing Seat Changes

Joan Boyar [1,2] Susan Krarup [3] Morten N. Nielsen [1,4]

Department of Mathematics and Computer Science,
University of Southern Denmark,
Odense, Denmark

Abstract

We consider a new variant of the Seat Reservation Problem [4] in which seat changes are allowed. We analyze the problem using the competitive ratio and the competitive ratio on accommodating sequences [4]. A very promising algorithm defined in this paper is Min-Change, which will ask passengers to change seat, only if they would otherwise have been rejected. Min-Change belongs to a large class of *conservative* algorithms, for which we prove very high performance guarantees. For instance when assuming that all of the passengers could have been seated by an optimal off-line algorithm, at least $\frac{2}{3}$ of the passengers can be seated on-line with only one seat change and at least $\frac{3}{4}$ will be seated if two seat changes are allowed. This should be compared to the performance guarantee of $\frac{1}{2}$ for the best deterministic on-line algorithm when no seat changes are allowed [2].

1 Introduction

In [4] the Seat Reservation Problem was analyzed. This is an on-line problem in which the ticket agents attempt to maximize income from the tickets sold. In this paper we consider an interesting new variant of the Seat Reservation Problem, in which a seat reservation could involve a change of seats during the trip. For this variant much more promising results can be shown. We use competitive analysis: the standard competitive ratio [6,8,10], and the competitive ratio on accommodating sequences [4]. Two different pricing policies are considered, the natural *proportional price* policy and the *unit price* policy, which is motivated by the train system in Denmark. For the unit price problem on accommodating sequences, a ratio of $\frac{1}{2}$ was shown to hold for the most

[1] Supported in part by the Danish Natural Science Research Council (SNF) and in part by the IST Programme of the EU under contract number IST-1999-14186 (ALCOM-FT).
[2] Email: joan@imada.sdu.dk
[3] Email: susan@imada.sdu.dk
[4] Email: nyhave@imada.sdu.dk

natural deterministic on-line algorithms in [4]. Later [2] it was shown that $\frac{1}{2}$ is best possible for all deterministic on-line algorithms. Also for our problem, there are algorithms obtaining a ratio of $\frac{1}{2}$ only, but we can show that a ratio of at least $\frac{s+1}{s+2}$ can be promised for a very natural class of algorithms, where s denotes the number of seat changes allowed. This result can be compared directly with the ratio of $\frac{1}{2}$ from [2,4], since in the two problems, the optimal algorithm has exactly the same power when considering accommodating sequences.

The restriction to accommodating sequences has been extended to a function of restrictions, the accommodating function, in a number of papers [1,3,5]. This function was used to distinguish between different algorithms having the same competitive ratio and competitive ratio on accommodating sequences. These papers give reason to believe that, other than the competitive ratio, accommodating sequences are the most important special case of the accommodating function. In particular, for a variant of Bin Packing a general theorem extending results obtained on accommodating sequences to results on the accommodating function is given [1].

Let s denote the number of seat changes allowed. In this paper we show the following results for the unit price problem:

	Accommodating Sequences	All Sequences Competitive Ratio
General	$= 1/2$	$O((n+s)/(n+k))$
Conservative	$= (s+1)/(s+2)$	$O(1/k)$ OPT general $O(s^2/k)$ OPT conservative
First-Fit* Min-Change	$= (s+1)/(s+2)$	$O(1/k)$ OPT general $O(s/k)$ OPT conservative

The upper bound on First-Fit* for the unit price problem restricted to accommodating sequences can be found in the full paper.

For the proportional price problem, we prove a tight lower bound of $\Omega(\frac{s}{k})$ on the competitive ratio on accommodating sequences. This bound is matched by the behavior of Min-Change. For the competitive ratio of the proportional price problem, we prove an upper bound of $O(\frac{s}{k})$ for all algorithms. In the full paper we also prove an upper bound of $O(\frac{s^2}{k})$ for First-Fit* on accommodating sequences.

2 Problem Definition

The set-up from [4] is as follows: A train with n seats travels from a start station to an end station, stopping at $k \geq 2$ stations, including the first and last. Reservations can be made for any trip from a station u to a station

v, if $u < v$. The passenger is given a single seat number when the ticket is purchased, which can be any time before departure. The algorithms (ticket agents) attempt to maximize income, i.e., the sum of the prices of the tickets sold. For political reasons, the problem must be solved in a *fair* manner, i.e., the ticket agent may not refuse a passenger if it is possible to accommodate him when he attempts to make his reservation. All algorithms, even OPT, the optimal off-line algorithm, must be fair.

The version considered in this paper allows the seat reservation system to assign seats so that each passenger may change seats a fixed number of times during the trip. We use s to denote the number of seat changes allowed. Two different pricing policies are considered, the natural *proportional price* policy where the price of a ticket is proportional to the length of the trip, and the *unit price* policy where the price is 1 for any ticket.

We use competitive analysis to measure the performance quality. Let $\mathbb{A}(I)$ denote the benefit obtained by the on-line algorithm $\mathbb{A}$ when run on I and let $\mathrm{OPT}(I)$ denote the maximal benefit which can be obtained by any algorithm. $\mathbb{A}$ is said to be c-competitive, if $\mathbb{A}(I) \geq c \cdot \mathrm{OPT}(I)$ for all input sequences I. The competitive ratio is the supremum of all c for which the algorithm is c-competitive. A sequence for which OPT can accept all requests using only n seats is called an *accommodating sequence*. In this paper we are particularly interested in the competitive ratio when the input sequences are restricted to accommodating sequences. This assumption, that there are enough seats for the optimal off-line algorithm, is appropriate whenever the management has done a reasonable job of predicting ticket demand and has thus assigned an appropriate number of cars to the train.

The requests considered are intervals. Since we allow the algorithms to perform s seat changes, these intervals might be broken into (at most) $s+1$ subintervals. Two (sub)intervals u and v overlap, if $u \cap v \neq \emptyset$. Note that our intervals are half-open, i.e., we do not consider it an overlap if one request starts at the station where the other request ends. For a station t, let $d(t, t+1)$ denote the number of intervals containing $[t, t+1)$. For a sequence I we let $d(I) = \max_{1 \leq t \leq k-1} d(t, t+1)$ denote the *density* of the sequence. The Seat Reservation Problem is similar to the problem of coloring an interval graph on-line: Each request is a node in the graph. If two requests overlap, there is an edge between them. The difference is that the number of colors is limited in the Seat Reservation Problem; instead of attempting to minimize the number of colors, the algorithms are attempting to maximize the sum of the prices on the intervals which are given colors. Interval graphs are *perfect* [7], so the size of the largest clique is exactly the number of colors needed. Thus, if $d(I) \leq n$ the optimal off-line algorithm will be able to accommodate all requests. The same clearly holds for any fair on-line algorithm if an arbitrary number of seat changes is allowed. For any set S, we use $|S|$ to denote the number of elements in S.

For two (sub)intervals u and v, we define $\mathrm{left}(u, v)$ to be the maximal

subinterval of u which contains the leftmost endpoint of u and which can be seated on the same seat as v without overlapping v. Moreover, if u and v have the same start station, we let $u \setminus v$ denote the possibly empty subinterval ranging from the end station of v to the end station of u. Otherwise $u \setminus v$ is undefined.

In this paper we use the following definition of the algorithm First-Fit*: An interval is seated on the first seat available. This definition is simpler than the one given in the full paper. This is possible since, for the sequences considered here, all accepted intervals have length one.

Definition 2.1 Suppose an algorithm accepts a new interval u by seating $l \leq s+1$ subintervals. If the first $l-1$ subintervals are placed such that an already seated (sub)interval starts right after the subinterval ends, we call the algorithm *conservative*.

Having decided to seat a (sub)interval on seat i, a conservative algorithm will seat as much as possible of the remaining journey on seat i before changing to another seat. First-Fit*, as defined in the full paper, is an example of a conservative algorithm.

An algorithm will be called Min-Change if it is conservative and it places an interval such that as few seat changes as possible are made for that interval. For the sequences considered here, all accepted intervals can be seated without seat changes, and we assume that Min-Change uses the First-Fit seating strategy for these sequences.

In the full paper, we do not restrict Min-Change's seating strategy to First-Fit but we consider Min-Change to be a class of conservative algorithms for which ties are broken arbitrarily under the restriction that the minimum number of seat changes is performed.

Later in this paper we will use the sequences W and W' below as a first part of different worst case sequences. Assume $s+1$ divides $k-1$ and n and that $k-1$ divides $n(s+1)$. Let $r = \frac{k-1}{s+1}$. Let W denote the sequence:

- For $i = 0, \ldots, s$ and for $j = 1 \ldots n$: $[p, p+1)$, where $p = ((ir + j - 1) \mod k) + 1$.

See Figure (a) for an example packing.

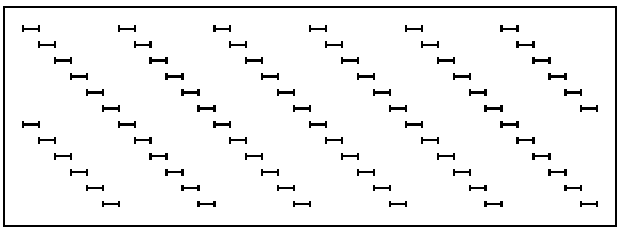
Under the same assumptions, let W' denote the sequence (having the same density $d(s, s+1)$ as W for every station $1 \leq s < k$):

- For $i = 0, \ldots, s$ and for $j = 0 \ldots \frac{n}{s+1} - 1$: $[p, p+s+1)$, where $p = ((ir + j(s+1)) \mod k) + 1$.

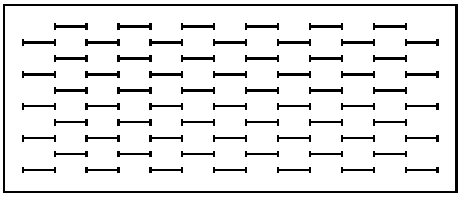
We use the term *hole* to denote the empty space on a seat. The size of a hole is the number of consecutive trips of unit size it spans. It is easy to verify the following:

Lemma 2.2 *The following four properties hold for the sequences W and W':*

- $|W| = n(s+1)$.



(a) A placement of W for $k = 37$, $n = 12$ and $s = 5$.



(b) Checker board for $k = 27$ and $n = 10$.

- $|W'| = n$.
- *In Figure (a) all holes have size at most* $\frac{k-1}{s+1} - 1$.
- *The density of* W *and* W' *is* $d(s, s+1) = \frac{n(s+1)}{k-1}$ *for every station* $1 \leq s < k$.

3 Unit Price, Accommodating Sequences

In this section we prove upper and lower bounds on the competitive ratio on accommodating sequences for different classes of algorithms for the unit price problem.

Theorem 3.1 *There exists an algorithm for the unit price problem with a competitive ratio on accommodating sequences of at most* $\frac{1}{2}$*, if* $s \in o(k)$.

Proof. Assume n is even and $2(s+1)$ divides $k-1$. Given $\frac{n}{2}$ of each of the following intervals,

- for $i = 0, \ldots, \frac{k-1}{2(s+1)} - 1$, let $p = 2i(s+1) + 1$: $[p, p + 2(s+1))$,

there exists a deterministic algorithm which would split these requests into $s+1$ intervals each of size 2, and seat them as shown in Figure (b). Then the adversary can give $\frac{n}{2}$ times the following:

- For $i = 0, \ldots, \frac{k-1}{2(s+1)} - 2$, let $p = 2i(s+1) + 2$ and give $[p, p + 2(s+1))$.

None of these intervals can be accepted by the algorithm using only s seat changes. OPT will seat an interval on the first available seat and accept all requests. The performance ratio is $\frac{\frac{n(k-1)}{4(s+1)}}{\frac{n(k-1)}{4(s+1)} + \frac{n}{2}(\frac{k-1}{2(s+1)} - 1)} = \frac{k-1}{k-1+(k-1)-2(s+1)}$. □

In order to prove a lower bound (positive result) on different classes of algorithms, we prove a lower bound on the number of requests an on-line algorithm $\mathbb{A}$ accepts and an upper bound on how many requests it rejects. First, we group the (sub)intervals $\mathbb{A}$ accepts into components.

Considering a seating performed by an algorithm, $\mathbb{A}$, we will merge two (sub)intervals, if one starts right after the other ends. If this kind of merge is performed repeatedly seat by seat, we end up having larger (not connected) *components*. We use γ to denote the number of such components.

Theorem 3.2 *Let $\mathbb{A}$ denote an algorithm for the unit price problem. If A denotes the set of intervals accepted by $\mathbb{A}$ and γ denotes the number of components in $\mathbb{A}$'s seating, then*

(i) $\gamma \leq (s+1)|A|$.

(ii) *If $\mathbb{A}$ is a conservative algorithm, $\gamma \leq |A|$.*

Proof. To prove (1) it is enough to note that every interval accepted by $\mathbb{A}$ can be broken into at most $s+1$ subintervals (components), since at most s seat changes are allowed. To prove (2): When a new interval is considered, we mark the entire interval, if no seat changes are performed. If $l > 0$ seat changes are performed, we mark the last subinterval, while all other intervals are left unmarked. The number of marks is exactly $|A|$. We prove by induction on the number of elements accepted that every component contains at least one mark. When the first interval is accepted, a new component is created. This component is marked, since the interval will be seated on one seat by the definition of a conservative algorithm. Suppose m intervals have been accepted, and every component contains a marked (sub)interval. Consider the next interval to be accepted. If this interval is not broken into subinterval, it is marked and the claim is true. If the interval is broken into $l+1$ subintervals, the first l subintervals must be connected to an existing component, by the definition of a conservative algorithm, and the last subinterval, which might create a new component, is marked. Again the claim is true. □

Theorem 3.3 *Consider a sequence in which all requests could be accepted by OPT, i.e., an accommodating sequence. If U denotes the set of intervals rejected by some algorithm $\mathbb{A}$, and γ denotes the number of components in $\mathbb{A}$'s seating, then $(s+1)|U| \leq \gamma$.*

Proof. Let $\mathbb{A}$ denote an arbitrary algorithm for the unit price problem. Let I denote a worst case sequence for $\mathbb{A}$ which can be accepted by an optimal algorithm on n seats. Define S to be the set of components in $\mathbb{A}$'s seating, so $\gamma = |S|$. We will define a partial function $f : S \to U$ having the following properties:

- $|f^{-1}(u)| = s+1$ for all $u \in U$.
- For all $u_1, u_2 \in U$ where $u_1 \neq u_2 : f^{-1}(u_1) \cap f^{-1}(u_2) = \emptyset$.

To each element in U, f assigns $s+1$ different components from S, with no component in S used twice. This means that $\gamma = |S| \geq (s+1)|U|$, proving the theorem.

During the construction of f, we maintain two sets U' and S'. Initially, let $S' = S$ and let $U' = \{(u, s) \mid u \in U\}$. The two sets should be interpreted as follows: An element $(v, t) \in U'$ represents an (sub)interval v which is allowed to change seat t times. S' will at any point represent a legal seating.

The (sub)intervals will be broken into smaller pieces, and the counter t will be decreased during the construction. Furthermore, we will at any point

maintain the following invariants:

(i) No element (u,t) from U' can be accepted (added to S') even if t seat changes are allowed.

(ii) $d(\pi_1(U') \cup S') \leq n$, where $\pi_1(U')$ denotes the set of intervals represented by U'.

Note that these invariants hold initially. Order the elements $(v,t) \in U'$ according to the left endpoint (starting station) of v. This ordering will be maintained when new subintervals are added to U'. For the first element (v,t) in this ordering, let p denote the start station of the (sub)interval represented by (v,t). Since $d(\pi_1(U') \cup S') \leq n$, there is at least one free seat in the seating represented by S' from station p to station $p+1$. By the first invariant, there must be a first (leftmost) element $s' \in S'$ on this seat blocking the (sub)interval. Define $f(s') = v$, let $v' = \text{left}(v, s')$, and let $v'' = v \setminus v'$. We update U' in the following way: Delete the pair representing v, and if $t > 0$, add a new pair $(v'', t-1)$. We modify S' in the following way: Insert the item $v' \cup s'$ instead of s'. The net effect is that a counter has possibly been decreased by one and a subinterval has moved from U' to S' which will not increase the density, $d(\pi_1(U') \cup S')$. Thus, the second invariant still holds. If v'' could be accepted using $t-1$ seat changes, we could accept v using t seat changes because of the way v'' was constructed. But this cannot be the case, since the first invariant held before this step. Thus, it also holds after this step. Note that according to the ordering of U', no element from S' will be a first blocking (sub)interval twice. Since every element starts with a counter s and part of it is kept in U' until this counter is 0, it will be a first element in the ordering of U' exactly $s+1$ times. Every time this happens a new element from S will be mapped to it. Therefore, $|f^{-1}(u)| = s+1$, for all $u \in U$. □

Corollary 3.4 *Any algorithm for the unit price problem has a competitive ratio on accommodating sequences of at least* $\frac{1}{2}$.

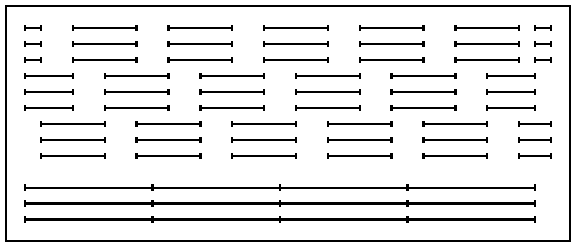
Proof. Let $\mathbb{A}$ denote an algorithm for the unit price problem. According to Theorem 3.2, the number of components γ in $\mathbb{A}$'s seating is at most $(s+1)|A|$, where A denotes the set of intervals accepted by $\mathbb{A}$. From Theorem 3.3, we have that $(s+1)|U| \leq \gamma$. Inserting this into the performance ratio, we get $\frac{|A|}{|A|+|U|} \geq \frac{|A|}{|A|+|A|} = \frac{1}{2}$. □

Theorem 3.1 shows that this result is tight.

Corollary 3.5 *Any conservative algorithm for the unit price problem has a competitive ratio on accommodating sequences of at least* $\frac{s+1}{s+2}$.

Proof. Using the same notation and arguments as in the previous corollary, we get $\gamma \leq |A|$ and $(s+1)|U| \leq \gamma$. This gives a performance ratio at least $\frac{|A|}{|A|+\frac{|A|}{s+1}} = \frac{s+1}{s+2}$. □

Thus the restriction to conservative algorithms, which might be made for



(c) Placement for $k = 34$, $n = 9$ and $s = 3$.

political reasons, appears to also be an advantage in terms of performance. We now show that the result is tight due to Min-Change's behavior.

Theorem 3.6 *The competitive ratio on accommodating sequences for Min-Change is at most* $\frac{(s+1)k+4s+4}{(s+2)k+2s+1}$.

Proof. Assume that $k-4$ is divisible by 6. Consider the request sequence which starts with $\frac{n}{3}$ requests for each interval in each of the following three phases (See Figure (c)):

- $[1,2)$, $[k-1,k)$, and $[6i-2,6i+2)$ for $i = 1,\ldots,\frac{k-4}{6}$.
- $[1,4)$, $[k-4,k-1)$, and $[6i,6i+4)$ for $i = 1,\ldots,\frac{k-4}{6}-1$.
- $[k-2,k)$ and $[6i-4,6i)$ for $i = 1,\ldots,\frac{k-4}{6}$.

When these requests are accepted by Min-Change, none of them have to change seats. Phase 1 is accepted on the first $\frac{n}{3}$ seats, Phase 2 is accepted on the next $\frac{n}{3}$ seats, and Phase 3 will use the remaining $\frac{n}{3}$ seats. Min-Change will accept these $\frac{n}{3}(3\frac{k-4}{6}+4) = \frac{k+4}{6}n$ requests, but none of the following requests:

- $\frac{n}{3}$ times $[2(s+1)i-2s-1, 2(s+1)i+1)$ for $i = 1,\ldots,\lfloor\frac{k-1}{2(s+1)}\rfloor$.

OPT uses $\frac{2n}{3}$ seats to accommodate the first three phases, since the density between any two stations is $\frac{2n}{3}$. OPT can use the remaining $\frac{n}{3}$ seats to accept the last phase. We get the following ratio: $\frac{\frac{k+4}{6}n}{\frac{k+4}{6}n+\frac{1}{3}n(\frac{k-1}{2(s+1)}-1)} = \frac{k+4}{k+4+2(\frac{k-1}{2(s+1)}-1)} = \frac{(s+1)k+4s+4}{(s+2)k+2s+1}$. □

Clearly, the ratio could be made even closer to $\frac{s+1}{s+2}$ by shortening many of the holes to length 1.

4 Unit Price, Competitive Ratio

We showed above that the competitive ratios are constant, when the input sequences are restricted to accommodating sequences. In this section we consider the competitive ratio (on all sequences) and obtain much more pessimistic performance guarantees.

Lemma 4.1 *If* $n = 2$, *the competitive ratio of any algorithm is at most* $\frac{s+5}{k+s+2}$.

Proof. Let $\mathbb{A}$ denote an arbitrary algorithm. The adversary gives $s+2$ unit intervals starting at the first stations except station 3:

- $[i, i+1)$ for $i = 1, \ldots s+3$ and $i \neq 3$.

We divide the proof into two cases depending on how $\mathbb{A}$ placed these intervals.

- Case 1: There exist two consecutive intervals on the same seat. (We also consider $[2,3)$ and $[4,5)$ to be consecutive.)
- Case 2: All intervals were placed alternating between the two seats.

If Case 1 occurs, the adversary gives

- $[1, k)$,

which can be accepted by $\mathbb{A}$ using at most s seat changes. Note that from station $s+4$ to k, this request will change seat at most s times. Therefore, the next interval

- $[s+4, k)$

can be accepted using at most s seat changes (it would change seat at the same positions). At this point $\mathbb{A}$ has filled both seats except between stations 3 and 4, where only one interval is given. Therefore, the following unit size intervals must be rejected by $\mathbb{A}$:

- $[i, i+1)$ for $i = 1, \ldots, k-1$ and $i \neq 3$.

OPT can arrange the first $s+2$ intervals alternating between the two seats. Therefore, the interval of length $k-1$ is rejected and there is room for the remaining intervals.

If Case 2 occurs, the adversary gives

- $[3, 4)$.

Now consider the $s+2$ intervals starting at stations 2, ..., $s+3$. Since the interval just given must be placed on the same seat as either $[2,3)$ or $[4,5)$, we are in a situation similar to Case 1, in which two consecutive intervals are on the same seat. Instead of giving $[1, k)$ as in Case 1, the adversary gives

- $[2, k)$.

The remaining intervals are the same except that $[1,2)$ should be left out, and [3,4) can be included.

In Case 2, the algorithm will get the best performance ratio. In this case its performance is $s+2+1+1+1 = s+5$. OPT has a performance of $s+2+1+1+(k-2) = k+s+2$. □

This result can be extended to larger n, as long as n remains small compared to s and k.

Corollary 4.2 *The competitive ratio for any algorithm for the unit price problem is at most* $\frac{n+s+3}{n+k+s}$.

Proof. For n arbitrary, the adversary first gives

- $n-2$ times $[1,k)$

which will be accepted by both algorithms. After this we are left with two seats, and we can apply the lemma above. Both algorithms will in this case accept $n-2$ extra intervals. □

In cases where the on-line algorithms are restricted to being conservative for political reasons, this would be part of the problem definition, so OPT should also be conservative. In the remainder of this section, we consider conservative algorithms and distinguish between the case where OPT is also restricted to being conservative and where it is not.

Theorem 4.3 *For n arbitrary, any conservative algorithm has a competitive ratio of $O(1/k)$ for the unit price problem, if OPT is not restricted to being conservative.*

Proof. Let $\mathbb{A}$ denote a conservative algorithm. The adversary gives

- $\frac{n}{2}$ times $[1,s+2)$,
- $\frac{n}{2}$ times $[s+2,s+3)$.

Since $\mathbb{A}$ is conservative, no seat changes are performed for the first $\frac{n}{2}$ intervals. Therefore, the algorithm must accept the following:

- $\frac{n}{2}$ times $[1,k)$.

OPT will behave as in Figure (b) (but with holes of size 1) and reject all the long intervals, and therefore make room for:

- $\frac{n}{2}$ times $[s+3,k)$,
- $\frac{n}{2}$ times $[i,i+1)$ for $i=0,\ldots,k-1$.

The small intervals given at the end are rejected by $\mathbb{A}$. The performance ratio is $\frac{\frac{n}{2}+\frac{n}{2}+\frac{n}{2}+\frac{n}{2}}{\frac{n}{2}+\frac{n}{2}+\frac{n}{2}+\frac{n}{2}(k-1)} = \frac{4}{2+k}$. □

Theorem 4.4 *Any conservative algorithm has a competitive ratio of $O(\frac{s^2}{k})$ for the unit price problem, if OPT is conservative.*

Proof. Let $\mathbb{A}$ denote a conservative algorithm for the unit price problem. The adversary gives

- $\frac{n}{2}$ times $[i,i+1)$ for $i=1,\ldots,s+2$.

Let q denote the number of holes of size at least 2. If an interval containing $[1,s+3)$ is given, it will be accepted if there is at least one hole of size at least 2 (and the density is smaller then n). On the other hand, since an interval is allowed to change seat at most s times, at most $s+1$ of these holes will be used (and either disappear or reduce their size to 1). $\mathbb{A}$ will therefore have to accept x intervals containing $[1,s+3)$, where $\frac{q}{s+1} \leq x \leq q$. We divide the proof into two cases depending on q.

- Case 1: $q > \frac{s+1}{2(s+2)}n$.
- Case 2: $q \leq \frac{s+1}{2(s+2)}n$.

If Case 1 occurs, the adversary gives

- q times $[1, k)$.

In this case, we let x count the number of intervals of length $k-1$ accepted by $\mathbb{A}$. The adversary continues with

- $n - x$ times $[s+3, k)$,
- x times $[i, i+1)$ for $i = s+3, \ldots, k-1$.

OPT behaves as in Figure (b) (but with holes of size 1) and rejects all intervals of length $k-1$. The performance ratio is $\frac{\frac{n}{2}(s+2)+x+(n-x)}{\frac{n}{2}(s+2)+(n-x)+x(k-1)} \leq \frac{\frac{n}{2}(s+2)+n}{\frac{n}{2}(s+2)+n+\frac{q}{s+1}(k-2)} \leq \frac{\frac{n}{2}(s+2)+n}{\frac{n}{2}(s+2)+n+\frac{n}{2(s+2)}(k-2)} = \frac{(s+2)(s+4)}{(s+2)(s+4)+k-2}$.

If Case 2 occurs, the adversary gives

- $\frac{n}{2}$ times $[1, s+3)$.

In this case, we let x count the number of intervals of length $s+2$ accepted by $\mathbb{A}$. The adversary continues with

- $\frac{n}{2} - x$ times $[2, k)$,
- $\frac{n}{2} + x$ times $[s+3, k)$,
- $\frac{n}{2} - x$ times $[i, i+1)$ for $i = s+3, \ldots, k-1$.

When the intervals of the form $[2, k)$ are received, at most s seat changes can be necessary. So they are all accepted. The [s+3,k) intervals are accepted without seat changes, so the last group of requests must all be rejected. OPT can arrange to accept all of the [1,s+3) intervals and thus reject all of the [2,k) intervals, leaving space for the short intervals at the end. The performance ratio is $\frac{\frac{n}{2}(s+2)+x+\frac{n}{2}-x+\frac{n}{2}+x}{\frac{n}{2}(s+2)+\frac{n}{2}+\frac{n}{2}+x+(\frac{n}{2}-x)(k-(s+3))} = \frac{\frac{n}{2}(s+2)+n+x}{\frac{n}{2}(s+2)+\frac{3n}{2}+(\frac{n}{2}-x)(k-s-4)} \leq \frac{\frac{n}{2}(s+2)+n+q}{\frac{n}{2}(s+2)+\frac{3n}{2}+(\frac{n}{2}-q)(k-s-4)} \leq \frac{\frac{n}{2}(s+2)+n+\frac{s+1}{2(s+2)}n}{\frac{n}{2}(s+2)+\frac{3n}{2}+(\frac{n}{2}-\frac{s+1}{2(s+2)}n)(k-s-4)} = \frac{(s+2)(s+4)+s+1}{(s+2)(s+5)+k-s-4}$. □

Turning to the specific conservative algorithm, First-Fit*, we can improve the upper bound to $O(\frac{s}{k})$.

Theorem 4.5 *If OPT is conservative, First-Fit* will have a competitive ratio of* $O(\frac{s}{k})$.

Proof. Assume $s+1$ divides $k-1$ and n and that $k-1$ divides $n(s+1)$. Let $r = \frac{k-1}{s+1}$. Let $d = \frac{n(s+1)}{k-1}$. An adversary will give

- W
- $n - d$ times $[1, k)$,

which First-Fit* will accept. OPT will behave as in Figure (a) on W and reject all these intervals. The adversary then gives

- $n-d$ times $[i, i+1)$ for $i = 1, \ldots, k-1$,

which will be accepted by OPT and rejected by First-Fit. The performance ratio is $\frac{(s+1)n+(n-d)}{n(k-1)} \leq \frac{s+2}{k-1}$. □

5 Proportional Price, Accommodating Sequences

For the original proportional price problem without seat changes, there was not much of a difference between the competitive ratio and the competitive ratio on accommodating sequences [4]. The same is shown to be true for any Min-Change variant in the full paper (i.e. not only restricted to the First-Fit seating strategy as here). Here we focus on a positive result. For any set S, let $T(S)$ denote the sum of the lengths of the (sub)intervals in S.

Theorem 5.1 *Any algorithm for the proportional price problem has a competitive ratio on accommodating sequences of at least* $\frac{s+1}{k+s}$.

Proof. Let $\mathbb{A}$ denote an arbitrary algorithm for the proportional price problem, let S denote the set of components accepted by $\mathbb{A}$, and let U denote the set of intervals rejected. From Theorem 3.2 and Theorem 3.3 we know that $|S| \geq |U|(s+1)$. Since every component in S has size at least 1 and every element in U has size at most $k-1$, we obtain the claimed ratio: $\frac{T(S)}{T(S)+T(U)} \geq \frac{|U|(s+1)}{|U|(s+1)+|U|(k-1)}$. □

This bound is tight due to the following theorem.

Theorem 5.2 *If* $s \leq k-5$, *the competitive ratio on accommodating sequences of Min-Change is at most* $\frac{2s+10}{k+2s+9}$.

Proof. Let $p \geq s+3$ be chosen smallest possible such that 3 divides $p+1$. The adversary uses a sequence very similar to the sequence in Theorem 3.6, giving $\frac{n}{3}$ of each of the intervals:

- $[1, 2)$, $[p-1, p)$, and $[3i, 3i+2)$ for $i = 1, \ldots, \frac{p-4}{3}$,
- $[3i+1, 3i+3)$ for $i = 0, \ldots, \frac{p-4}{3}$,
- $[3i+2, 3i+4)$ for $i = 0, \ldots, \frac{p-4}{3}$.

The sequence above is contained in [1,p), where $s+3 \leq p \leq s+5$. For each station the sequence has a density of $\frac{2n}{3}$, giving Min-Change a performance of $p\frac{2n}{3}$. Min-Change will occupy all n seats such that no seat is free for two consecutive trips of unit size between station 1 and station p. This means that an element spanning all intervals given above, must be rejected. The adversary continues giving

- $\frac{n}{3}$ times [1,k),

which can all be accepted by OPT, since the density is nowhere more than n. OPT will have a performance of $p\frac{2n}{3} + \frac{n}{3}(k-1)$. The performance ratio is at most $\frac{p\frac{2n}{3}}{p\frac{2n}{3}+\frac{n}{3}(k-1)} \leq \frac{2(s+5)}{2(s+5)+(k-1)} = \frac{2s+10}{k+2s+9}$. □

6 Proportional Price, Competitive Ratio

Finally, we show that the same bound of $O(\frac{s}{k})$ can be obtained for all algorithms, if we do not restrict the adversary to accommodating sequences.

Theorem 6.1 *If $s \in o(k)$, any algorithm will have a competitive ratio of $O(\frac{s}{k})$ for the proportional price problem.*

Proof. Let $\mathbb{A}$ denote an arbitrary algorithm for the proportional price problem. The adversary gives

- $\frac{n}{2}$ times $[i, i+1)$ for $i = 1, \ldots, s+2$.

Let q be the number of times $[1, s+3)$ can be accepted by $\mathbb{A}$ after these intervals. We divide the proof into two cases:

- Case 1: $q \leq \frac{n}{4}$.
- Case 2: $q > \frac{n}{4}$.

In both cases the adversary gives

- q times $[1, s+3)$.

If Case 1 occurs, the adversary gives

- $\frac{n}{2} - q$ times $[1, k)$.

$\mathbb{A}$ must reject the long intervals. If OPT seats an interval on the first available seat, it will accept all the given items. The performance ratio will be at most $\frac{\frac{n}{2}(s+2)+q(s+2)}{\frac{n}{2}(s+2)+q(s+2)+(\frac{n}{2}-q)(k-1)} \leq \frac{\frac{n}{2}(s+2)+\frac{n}{4}(s+2)}{\frac{n}{2}(s+2)+\frac{n}{4}(s+2)+\frac{n}{4}(k-1)} = \frac{3(s+2)}{3(s+2)+(k-1)}$.

If Case 2 occurs, the adversary gives

- $\frac{n}{2} - q$ times $[2, s+3)$,
- q times $[2, k)$.

Also in this case $\mathbb{A}$ will accept everything except the long intervals. OPT should pack the first intervals as in Figure (b) (but with holes of size 1), reject the intervals $[1, s+3)$, and accept the remaining intervals. The performance ratio is $\frac{\frac{n}{2}(s+2)+q(s+2)+(\frac{n}{2}-q)(s+1)}{\frac{n}{2}(s+2)+(\frac{n}{2}-q)(s+1)+q(k-2)} \leq \frac{n(s+2)-\frac{n}{4}}{\frac{3n}{4}(s+1)+\frac{n}{4}(k-2)} = \frac{4(s+2)-1}{3(s+1)+k-2}$. □

7 Conclusion and Open Problems

For the unit price problem, there is a tremendous difference between the competitive ratio and the competitive ratio on accommodating sequence; the competitive ratio appears far too pessimistic. For the competitive ratio on accommodating sequences, in order to find a performance improvement due to more seat changes being allowed, conservative algorithms were considered, so that seat changes were only made when necessary. This seems to be a natural requirement for an algorithm and leads to improved competitive ratios on accommodating sequences with more allowed seat changes.

For the original unit price problem, the accommodating function with $\alpha <$

1 was shown to be useful in distinguishing between fair algorithms [3]. It would be interesting to see if this is also the case for fair, conservative algorithms when a fixed number of seat changes are allowed. One indication that this might be the case is a more recent result by the authors: There is an on-line algorithm which will accept all passengers, if the density is at most $\frac{n}{2}$ and one seat change is allowed. The algorithm resembles that in [9] showing that when no seat changes are allowed all passengers can be accepted on-line, if the density is at most $\frac{n}{3}$.

8 Acknowledgments

We would like to thank Thore Husfeldt for suggesting the problem with one seat change, motivated by Sweden's train system. We would also like to thank the referees for their careful reading of the paper and their helpful suggestions for improvements.

References

[1] Azar, Y., J. Boyar, L. Epstein, L. M. Favrholdt, K. S. Larsen and M. N. Nielsen, *Fair versus Unrestricted Bin Packing*, Technical Report PP-2000-20, Department of Mathematics and Computer Science, University of Southern Denmark, Odense (2000), preliminary version at SWAT 2000.
URL `ftp://ftp.imada.sdu.dk/pub/papers/pp-2000/20.ps.gz`

[2] Bach, E., J. Boyar, L. Epstein, L. M. Favrholdt, T. Jiang, K. S. Larsen, G.-H. Lin and R. van Stee, *Tight Bounds on the Competitive Ratio on Accommodating Sequences for the Seat Reservation Problem*, Technical Report PP-2000-16, Department of Mathematics and Computer Science, University of Southern Denmark, Odense (2000), preliminary version at COCOON 2000.
URL `ftp://ftp.imada.sdu.dk/pub/papers/pp-2000/16.ps.gz`

[3] Boyar, J., L. M. Favrholdt, K. S. Larsen and M. N. Nielsen, *Extending the Accommodating Function* (2001), submitted.

[4] Boyar, J. and K. S. Larsen, *The Seat Reservation Problem*, Algorithmica **25** (1999), pp. 403–417.

[5] Boyar, J., 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*, Lecture Notes in Computer Science **1663** (1999), pp. 74–79, to appear in SICOMP.

[6] Graham, R. L., *Bounds for Certain Multiprocessing Anomalies*, Bell Systems Technical Journal **45** (1966), pp. 1563–1581.

[7] Jensen, T. R. and B. Toft, "Graph Coloring Problems," John Wiley & Sons, 1995.

[8] Karlin, A. R., M. S. Manasse, L. Rudolph and D. D. Sleator, *Competitive Snoopy Caching*, Algorithmica **3** (1988), pp. 79–119.

[9] Kierstead, H. A. and J. W. T. Trotter, *An Extremal Problem in Recursive Combinatorics*, Congressus Numerantium **33** (1981), pp. 143–153.

[10] Sleator, D. D. and R. E. Tarjan, *Amortized Efficiency of List Update and Paging Rules*, Comm. of the ACM **28** (1985), pp. 202–208.