Improving the Competitive Ratios of the Seat Reservation Problem

Shuichi Miyazaki[1] and Kazuya Okamoto[2]

[1] Academic Center for Computing and Media Studies, Kyoto University
shuichi@media.kyoto-u.ac.jp
[2] Graduate School of Informatics, Kyoto University kazuya@kuhp.kyoto-u.ac.jp

Abstract. In the seat reservation problem, there are k stations, s_1 through s_k, and one train with n seats departing from the station s_1 and arriving at the station s_k. Each passenger orders a ticket from station s_i to station s_j ($1 \leq i < j \leq k$) by specifying i and j. The task of an online algorithm is to assign one of n seats to each passenger online, i.e., without knowing future requests. The purpose of the problem is to maximize the total price of the sold tickets. There are two models, the unit price problem and the proportional price problem, depending on the pricing policy of tickets. In this paper, we improve upper and lower bounds on the competitive ratios for both models: For the unit price problem, we give an upper bound of $\frac{4}{k-2\sqrt{k-1}+4}$, which improves the previous bound of $\frac{8}{k+5}$. We also give an upper bound of $\frac{2}{k-2\sqrt{k-1}+2}$ for the competitive ratio of Worst-Fit algorithm, which improves the previous bound of $\frac{4}{k-1}$. For the proportional price problem, we give upper and lower bounds of $\frac{3+\sqrt{13}}{k-1+\sqrt{13}}(\simeq \frac{6.6}{k+2.6})$ and $\frac{2}{k-1}$, respectively, on the competitive ratio, which improves the previous bounds of $\frac{4+2\sqrt{13}}{k+3+2\sqrt{13}}(\simeq \frac{11.2}{k+10.2})$ and $\frac{1}{k-1}$, respectively.

1 Introduction

The *seat reservation problem*, first introduced by Boyar and Larsen [4], is the following online problem. There are k stations s_1 through s_k, and one train with n seats numbered 1 through n. The train departs from the station s_1 and is destined for the station s_k. An input is a sequence of requests, where each request specifies an interval of the form $[i, j)$ ($1 \leq i < j \leq k$), meaning that the current passenger wants to buy a ticket from station s_i to station s_j. The task of an online algorithm is to select which seat to assign to this passenger (if there are more than one available seats), without knowing future requests. In this problem, any algorithm must be *fair*, i.e., if there is a seat available for the current passenger, it cannot reject her request. The purpose of the problem is to maximize the income, i.e., the sum of the prices of the sold tickets.

There are two models, *the unit price problem* and *the proportional price problem*, depending on the pricing policy of tickets. In the unit price problem, all tickets has the same price of 1. In the proportional price problem, the price of a ticket is proportional to the distance traveled, i.e., the price of a ticket from s_i to s_j is $j-i$.

The performance of an online algorithm is evaluated by the *competitive analysis*. Let ALG be an online algorithm and σ be an input sequence. Let OPT be an optimal offline algorithm, namely, it optimally works after knowing the complete information of σ. Also, let $p_{ALG}(\sigma)$ and $p_{OPT}(\sigma)$ be the income obtained by ALG and OPT, respectively, for σ. If $p_{ALG}(\sigma) \geq r \cdot p_{OPT}(\sigma) - d$ for any input σ, where d is a constant independent of σ, we say that ALG is r-competitive*.

Boyar and Larsen [4] studied the competitive ratios for both the unit price and the proportional price models. In particular, they studied three natural algorithms, *First-Fit*, *Best-Fit*, and *Worst-Fit*. First-Fit assigns each request to the available seat with the smallest number. Best-Fit assigns a request to a seat such that the empty space containing the current request interval is minimized (ties are broken arbitrarily). For example, suppose that there are eight stations and three seats, and that the current configuration is like Fig. 1, where shaded areas are assigned. Suppose that the next request is for the interval $[4, 6)$. We cannot assign it to the first seat. The empty space of the second (third, resp.) seat containing this interval is from s_2 to s_6 (from s_4 to s_7, resp.) and is of size 4 (3, resp.). So, Best-Fit selects the third seat for this request. Conversely, Worst-Fit assigns a request to a seat such that the empty space containing the current request interval is maximized (again, ties are broken arbitrarily). In an example of Fig 1, if Worst-Fit receives a request for $[4, 6)$, then it assigns it to the second seat. Table 1, taken from [6], summarizes the best-known results on the competitive ratios.

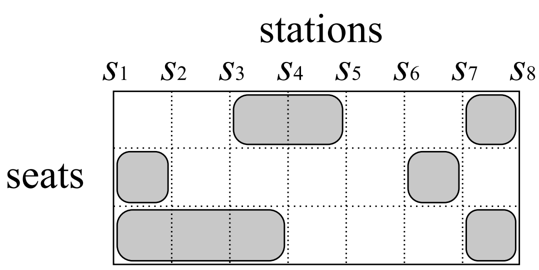

Fig. 1. An example configuration of assignment.

Our Contributions. In this paper, we improve both upper and lower bounds on the competitive ratios. Our results are summarized in Table 2. For the unit price problem, we improve an upper bound from $\frac{8}{k+5}$ to $\frac{4}{k-2\sqrt{k-1}+4}$. To improve a lower bound, we can see from Table 1 that it is hopeless to try to sophisticate the analysis for First-Fit or Best-Fit because an almost tight upper bound is already

* There is an alternative definition such that competitive ratios are always at least 1. But here we use this definition following the previous seat reservation papers.

Table 1. Upper and lower bounds on the competitive ratios.

	Unit Price	Proportional Price
Any deterministic algorithm	$\frac{2}{k} \le r \le \frac{8}{k+5}$	$\frac{1}{k-1} \le r \le \frac{4+2\sqrt{13}}{k+3+2\sqrt{13}} (\simeq \frac{11.2}{k+10.2})$
Worst-Fit	$\frac{2}{k} \le r \le \frac{4}{k+1}$	$r = \frac{1}{k-1}$
First-Fit/Best-Fit	$\frac{2}{k} \le r \le \frac{2-\frac{1}{k-1}}{k-1}$	$\frac{1}{k-1} \le r \le \frac{4}{k+2}$

Table 2. New results (results obtained in this paper are highlighted in boldface).

	Unit Price	Proportional Price
Any deterministic algorithm	$\frac{2}{k} \le r \le \boldsymbol{\frac{4}{k-2\sqrt{k-1}+4}}$	$\frac{1}{k-1} \le r \le \boldsymbol{\frac{3+\sqrt{13}}{k-1+\sqrt{13}} (\simeq \frac{6.6}{k+2.6})}$
Worst-Fit	$\frac{2}{k} \le r \le \boldsymbol{\frac{2}{k-2\sqrt{k-1}+2}}$	$r = \frac{1}{k-1}$
First-Fit/Best-Fit	$\frac{2}{k} \le r \le \frac{2-\frac{1}{k-1}}{k-1}$	$\boldsymbol{\frac{2}{k-1}} \le r \le \frac{4}{k+2}$

known for these algorithms, but there is some room for Worst-Fit. However, we show that Worst-Fit is also hopeless by improving its upper bound from $\frac{4}{k-1}$ to $\frac{2}{k-2\sqrt{k-1}+2}$. For the proportional price problem, we improve both upper and lower bounds. We improve an upper bound from $\frac{4+2\sqrt{13}}{k+3+2\sqrt{13}} (\simeq \frac{11.2}{k+10.2})$ to $\frac{3+\sqrt{13}}{k-1+\sqrt{13}} (\simeq \frac{6.6}{k+2.6})$. For a lower bound, we show that First-Fit and Best-Fit achieve the competitive ratio of $\frac{2}{k-1}$, which improves the previous bound of $\frac{1}{k-1}$. As a result, we improve the lower bound of the problem itself also. Note that previous lower bounds were obtained by using only the fact that algorithms are fair, and hence such bounds hold for any fair online algorithms. In contrast, the result in this paper is obtained by considering properties of First-Fit and Best-Fit that do not necessarily hold for all the fair online algorithms.

Related Results. Besides the competitive analysis, Boyar and Larsen [4] analyzed the problem using the *accommodating ratio*, which takes not all the possible input sequences but only *accommodating sequences* into account. An accommodating sequence is a sequence for which an optimal offline algorithm can accommodate all the requests. They gave upper and lower bounds of $\frac{8k-9}{10k-15}$ and $\frac{1}{2}$, respectively, on the accommodating ratio for the unit price problem [4]. Later, Bach et al. [1] gave the matching upper bound of $\frac{1}{2}$.

There are some results on randomized algorithms. Boyar and Larsen [4] gave an upper bound of $\frac{8k-9}{10k-15}$ on the accommodating ratio for the unit price problem in the oblivious adversary model. Furthermore, Bach et al. [1] improved both upper and lower bounds for this problem and gave a matching bound of $\frac{7}{9}$.

Boyar, Larsen, and Nielsen [5] generalized the accommodating ratio. They introduced a variable $\alpha(\geq 1)$ and allowed *α-sequences* as possible input sequences. An α-sequence is a sequence for which an optimal offline algorithm can accommodate all the requests using αn seats. Then, they gave upper and lower bounds on the generalized accommodating ratio for the unit price problem. Boyar et al. [2] extended the above performance guarantees to more general ones for $\alpha(\leq 1)$ and gave several upper and lower bounds of First-Fit, Worst-Fit, and other online algorithms.

Boyar and Medvedev [6] used the *relative worst order ratio* to compare the performance of online algorithms (without using optimal offline algorithms). They showed that for both the unit price and the proportional price problems, First-Fit and Best-Fit are better than Worst-Fit.

Boyar, Krarup, and Nielsen [3] proposed a variant that allows x seat changes for each request, i.e., one ticket can be divided into at most $x+1$ tickets for sub-intervals. They obtained several upper and lower bounds on the competitive and accommodating ratios.

Kohrt and Larsen [7] proposed a problem that lies in between the offline and online models. The task of an algorithm is not to assign a seat to a request but only to decide whether the request can be accepted or not (by arranging the previously accepted requests). They proposed an algorithm as well as an appropriate data structure, and proved that its running time is optimal.

2 The Unit Price Problem

For better understanding, we give a simple example for $k = 4$ and $n = 2$ (see Fig. 2). Consider the following input sequence $\sigma = (r_1, r_2, r_3, r_4, r_5)$, where r_1, r_2, r_3, r_4, and r_5 are requests for intervals $[1, 2)$, $[3, 4)$, $[1, 4)$, $[2, 4)$, and $[1, 2)$, respectively. Suppose that an online algorithm A assigns both r_1 and r_2 to the first seat. Then, it must assign r_3 to the second seat because it must be fair. So, it can accept neither r_4 nor r_5 and hence its income is 3. On the other hand, an optimal online algorithm for σ assigns r_1 and r_2 into different seats. It can then reject r_3 and accommodate both r_4 and r_5. So the income of this algorithm is 4.

2.1 An Upper Bound

We first improve a general upper bound.

Theorem 1. *No online algorithm for the unit price problem is more than* $\frac{4}{k-2\sqrt{k-1}+4}$*-competitive.*

Proof. Let A be an arbitrary online algorithm. Let m and c be arbitrary positive integers, and define $k = m^2 + 1$ be the number of stations and $n = 2cm$ be the number of seats. Our adversary first gives the request sequence σ_1 consisting

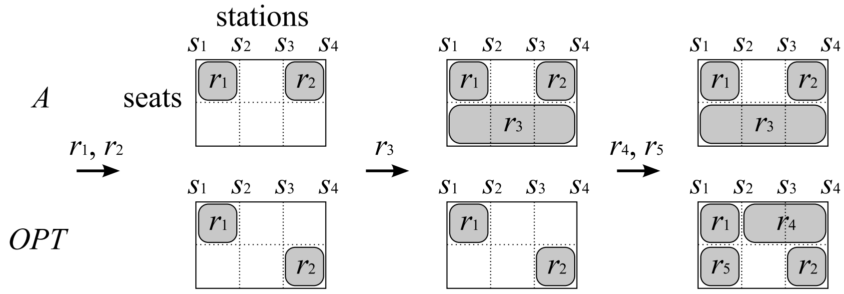

Fig. 2. An example of the unit price problem.

of $2c$ requests for the interval $[1, 2)$, $2c$ requests for the interval $[2, 3)$, ..., $2c$ requests for the interval $[m, m+1)$.

Let R be the set of seats to which A assigns at least one request for σ_1. We give a current assignment configuration in Fig. 3, in which seats are sorted appropriately: In region (i), at least one request is assigned for each seat. There may be or may not be assigned requests in region (ii). In region (iii), one request for the interval $[m, m+1)$ is assigned for each seat. No request is assigned in region (iv).

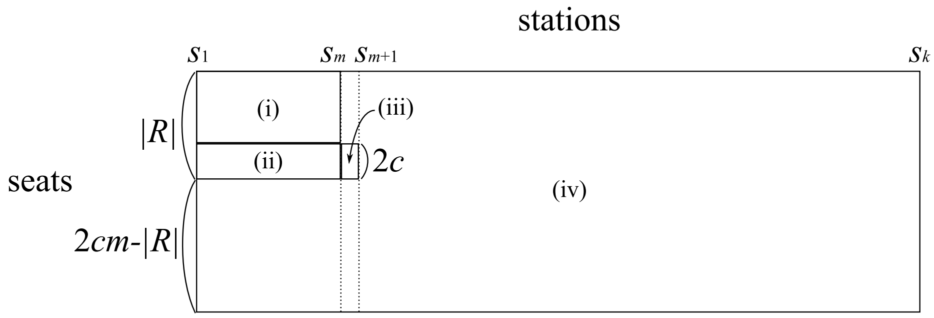

Fig. 3. Assignment configuration for σ_1 by algorithm A.

The adversary selects subsequent sequences depending on the size of R. It executes Case (1) if $|R| < c(m+1)$ and Case (2) otherwise.

Case (1): The adversary gives the following request sequences σ_2, σ_3, and σ_4 in this order: σ_2 consists of $2cm - |R|$ requests for the interval $[1, k)$. σ_3 consists of $|R| - 2c$ requests for the interval $[m, k)$. σ_4 consists of $2c$ requests for the interval $[m+1, k)$. It is easy to see that A accepts all the requests in σ_2, σ_3, and σ_4 because of the fairness, and hence after receiving σ_4, the whole region (iv) in Fig. 3 is filled by those requests. Finally, the adversary gives the sequence σ_5 consisting of

$2cm-|R|$ requests for the interval $[m, m+1)$, $2cm-|R|$ requests for the interval $[m+1, m+2)$, ..., and $2cm-|R|$ requests for the interval $[k-1, k)$, all of which are rejected by A. Thus, the income of A is $2cm+(2cm-|R|)+(|R|-2c)+2c$.

On the other hand, an optimal offline algorithm assigns each request of σ_1 to different seats. Then, it can reject all the requests in σ_2, and hence can accept all the requests in σ_3, σ_4, and σ_5. Thus, the income of the optimal offline algorithm is $2cm+(|R|-2c)+2c+(k-m)(2cm-|R|)$. Hence, the competitive ratio in this case is

$$\begin{aligned}
&\frac{2cm+(2cm-|R|)+(|R|-2c)+2c}{2cm+(|R|-2c)+2c+(k-m)(2cm-|R|)} \\
&= \frac{4cm}{2cm+|R|+(k-m)(2cm-|R|)} \\
&< \frac{4}{k-2\sqrt{k-1}+4}
\end{aligned}$$

because $|R| < c(m+1)$.

Case (2): The adversary gives the request sequences σ_2, σ_2', σ_3, and σ_4 in this order, where σ_2, σ_3, and σ_4 are the same as before and σ_2' consists of $|R|-2c$ requests for the interval $[1, m+1)$. It is easy to see that A rejects all the requests in σ_2' but accepts all the requests in σ_2, σ_3, and σ_4. So, again, the whole region (iv) in Fig. 3 is filled by those requests. Finally, the adversary gives the sequence σ_5' consisting of $|R|-2c$ requests for the interval $[m+1, m+2)$, $|R|-2c$ requests for the interval $[m+2, m+3)$, ..., and $|R|-2c$ requests for the interval $[k-1, k)$, all of which are rejected by A. Thus, the income of A is $2cm+(2cm-|R|)+(|R|-2c)+2c$.

On the other hand, an optimal offline algorithm assigns each request of σ_1 using First-Fit. Then, it accepts all the requests in σ_2, σ_2', σ_4, and σ_5', but rejects all the requests in σ_3. Thus, its income is $2cm+(2cm-|R|)+(|R|-2c)+2c+(k-m-1)(|R|-2c)$. Hence, the competitive ratio in this case is

$$\begin{aligned}
&\frac{2cm+(2cm-|R|)+(|R|-2c)+2c}{2cm+(2cm-|R|)+(|R|-2c)+2c+(k-m-1)(|R|-2c)} \\
&= \frac{4cm}{4cm+(k-m-1)(|R|-2c)} \\
&\leq \frac{4}{k-2\sqrt{k-1}+4}
\end{aligned}$$

because $|R| \geq c(m+1)$. □

2.2 An Upper Bound for Worst-Fit

Recall from Sec. 1 that Worst-Fit assigns each request to a seat such that the empty space containing the current request interval is maximized. As we have mentioned in Sec. 1, Worst-Fit has been a good candidate for improving a lower bound. But we rule out this possibility by giving almost tight upper bound for it.

Theorem 2. *The competitive ratio of Worst-Fit for the unit price problem is at most* $\frac{2}{k-2\sqrt{k-1}+2}$.

Proof. As in the proof of Theorem 1, let m and c be arbitrary positive integers, and let $k = m^2 + 1$ and $n = 2cm$. First, we give the sequence σ_1 consisting of $2c$ requests for the interval $[1, 2)$, $2c$ requests for $[2, 3)$, $\ldots$, $2c$ requests for $[m, m+1)$. Worst-Fit assigns these $n = 2cm$ requests to different seats. Next, we give σ_2, σ_3, and σ_4 in this order where σ_2 consists of $2cm - 2c$ requests for the interval $[1, m+1)$, σ_3 consists of $2cm-2c$ requests for the interval $[m, k)$, and σ_4 consists of $2c$ requests for the interval $[m+1, k)$. Worst-Fit rejects all the requests in σ_2 and accommodates all the requests in σ_3 and σ_4. So, after receiving σ_4, all the seats are full in the interval $[m + 1, k)$. Finally, we give σ_5 consisting of $2cm - 2c$ requests for $[m + 1, m + 2)$, $2cm - 2c$ requests for $[m + 2, m + 3)$, $\ldots$, $2cm - 2c$ requests for $[k - 1, k)$. Worst-Fit rejects all these requests. The income of Worst-Fit is then $2cm + (2cm - 2c) + 2c$.

On the other hand, an optimal offline algorithm assigns requests in σ_1 using First-Fit. Then it can accommodate all the requests in σ_2, and it rejects all the requests in σ_3. Hence, it can accept all the requests in σ_4 and σ_5, so its income is $2cm + (2cm - 2c) + 2c + (k - m - 1)(2cm - 2c)$. Thus the competitive ratio is

$$\begin{aligned}
&\frac{2cm + (2cm - 2c) + 2c}{2cm + (2cm - 2c) + 2c + (k - m - 1)(2cm - 2c)} \\
&= \frac{4cm}{4cm + (k - m - 1)(2cm - 2c)} \\
&= \frac{2}{k - 2\sqrt{k-1} + 2}.
\end{aligned}$$

□

3 The Proportional Price Problem

Recall that in the proportional price problem, the price of a ticket from s_i to s_j is $j - i$.

3.1 An Upper Bound

Theorem 3. *No online algorithm for the proportional price problem is more than* $\frac{3+\sqrt{13}}{k-1+\sqrt{13}}$*-competitive.*

Proof. Consider an arbitrary online algorithm A, and let k and $n(= 2m$ for a positive integer $m)$ be the numbers of stations and seats, respectively. The adversary first gives the sequence σ_1 consisting of m requests for the interval $[1, 2)$ and σ_2 consisting of m requests for the interval $[2, 3)$. Let R be the set of seats to which A assigns both requests of σ_1 and σ_2. The current configuration is given in Fig. 4, in which assigned areas are shaded.

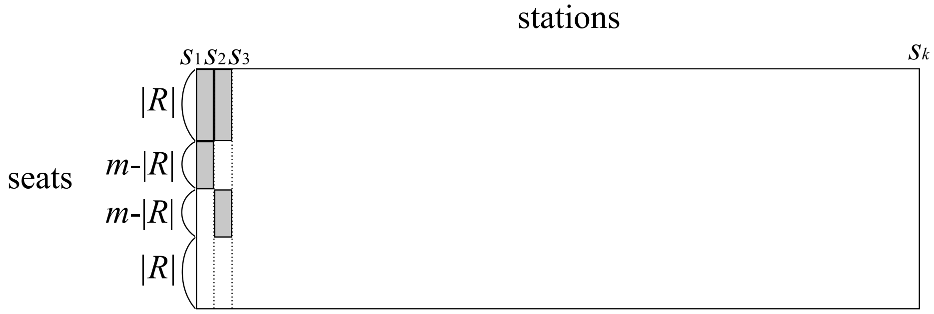

Fig. 4. Assignment configuration for σ_1 and σ_2 by algorithm A.

The adversary selects subsequent sequences depending on the size of R. It executes Case (1) if $|R| < \frac{(\sqrt{13}-2)m}{3}$ and Case (2) otherwise.

Case (1): The adversary gives σ_3 and σ_4 in this order such that σ_3 consists of $|R|$ requests for the interval $[1, 3)$ and σ_4 consists of $m - |R|$ requests for the interval $[1, k)$. A accepts all the requests in σ_3 but rejects all the requests in σ_4, so that its income is $2m + 2|R|$.

On the other hand, an optimal offline algorithm uses m seats to assign both requests of σ_1 and σ_2. Then, it can accomodate all the requests in σ_3 and σ_4 and hence its income is $2m + 2|R| + (k - 1)(m - |R|)$. The competitive ratio is then

$$\begin{aligned} &\frac{2m + 2|R|}{2m + 2|R| + (k-1)(m - |R|)} \\ &< \frac{2 + 2\frac{\sqrt{13}-2}{3}}{2 + 2\frac{\sqrt{13}-2}{3} + (k-1)(1 - \frac{\sqrt{13}-2}{3})} \\ &= \frac{3 + \sqrt{13}}{k + 2 + \sqrt{13}} \end{aligned}$$

because $|R| < \frac{(\sqrt{13}-2)m}{3}$.

Case (2): The adversary gives σ_3, σ'_4, and σ'_5 in this order where σ_3 is the same as before, σ'_4 consists of $m - |R|$ requests for the interval $[2, 3)$, and σ'_5 consists of $|R|$ requests for the interval $[2, k)$. A accommodates all the requests of σ_3 and σ'_4, but rejects all the requests of σ'_5, so, its income is $2m + 2|R| + (m - |R|)$.

On the other hand, an optimal offline algorithm assigns requests of σ_1 and requests of σ_2 to different seats, i.e., each of $2m$ seats contains exactly one request. Then, it can reject all the requests of σ_3 and can accommodate all the requests of σ'_4 and σ'_5, and hence its income is $2m + (m - |R|) + (k-2)|R|$. The competitive ratio is

$$\begin{aligned}
&\frac{2m + 2|R| + (m - |R|)}{2m + (m - |R|) + (k-2)|R|} \\
&\leq \frac{3 + \frac{\sqrt{13}-2}{3}}{3 - \frac{\sqrt{13}-2}{3} + (k-2)\frac{\sqrt{13}-2}{3}} \\
&= \frac{3 + \sqrt{13}}{k - 1 + \sqrt{13}}
\end{aligned}$$

because $|R| \geq \frac{(\sqrt{13}-2)m}{3}$. □

3.2 Lower Bounds for First-Fit and Best-Fit

Recall that First-Fit assigns each request to the available seat with the smallest number, and Best-Fit assigns a request to a seat such that the empty space containing the current request interval is minimized. We improve lower bounds on the competitive ratio for these algorithms, improving a general lower bound for the proportional price problem.

Theorem 4. *Both First-Fit and Best-Fit are $\frac{2}{k-1}$-competitive for the proportional price problem.*

Proof. We give a proof for First-Fit (denoted FF hereafter). The proof for Best-Fit is exactly the same. Consider an arbitrary input σ. If, for every seat, the total length of intervals assigned by FF is at least two, then we are done since FF earns at least $2n$ and OPT's income can be at most $(k-1)n$ for an instance with k stations and n seats. If there is a seat such that FF assigns no request, then FF rejects no request in σ, and again we are done. Hence, we assume that there is a seat q to which only an interval, say $I = [i, i+1)$, of length 1 is assigned. Let r be the request assigned to q by FF. We can see that no seat has a vacant space for I since if such a seat q' exists, assigned intervals of q and q' do not overlap, contradicting the definition of FF.

Let R_I be the set of requests for intervals containing I assigned by FF. By the above observation, $|R_I| = n$. Partition R_I into $R_I^{(1)}$ and $R_I^{(\geq 2)}$ so that $R_I^{(1)}$ is the set of requests for exactly the interval I, and $R_I^{(\geq 2)} = R_I \setminus R_I^{(1)}$ is the set of requests for intervals of length at least 2, containing I (see the upper figure of Fig. 5). Also, let $S^{(1)}$ and $S^{(\geq 2)}$ be the sets of seats to which requests in $R_I^{(1)}$ and $R_I^{(\geq 2)}$, respectively, are assigned. Note that $|S^{(1)}| = |R_I^{(1)}|$, $|S^{(\geq 2)}| = |R_I^{(\geq 2)}|$, and $|S^{(1)}| + |S^{(\geq 2)}| = n$.

Suppose that there is a request r' in $R_I^{(1)}$ that is rejected by OPT. Let R' be the set of requests for intervals containing I, accommodated by OPT. Since

OPT is fair but rejected r', $|R'| = n$ and any request in R' precedes r'. Since the interval I is full for both OPT and FF, and since r' is accepted by FF but rejected by OPT, there is a request $r'' \in R'$ rejected by FF. Note that r'' precedes r' since $r'' \in R'$, but FF rejected r'' while it accepted r'. So, the interval requested by r'' must include interval other than I, and when FF rejected r'', there must be a seat q'' in which the interval I was empty but some other intervals were assigned. If at this moment, FF has already received the request r and has assigned it to the seat q, then we can merge q and q'' without overlapping, contradicting the definition of FF. So, the request r has not been given to FF yet. But then q was empty for the whole interval at this moment, and FF could have assigned r'' to q, a contradiction. So, any request in $R_I^{(1)}$ is accepted by OPT.

Now, let S be the set of seats to which OPT assigns requests in $R_I^{(1)}$, and $R(S)$ be the set of requests assigned to S by OPT. Define $\overline{R} = R(S) \setminus R_I^{(1)}$ (see the lower figure of Fig. 5). Because FF is fair and the seat q (of FF) eventually contains only a request for the interval I, FF accommodates all the requests in $\overline{R}$. Also, since requests in $\overline{R}$ do not contain the interval I, $\overline{R}$, $R_I^{(1)}$, and $R_I^{(\geq 2)}$ are pairwise disjoint.

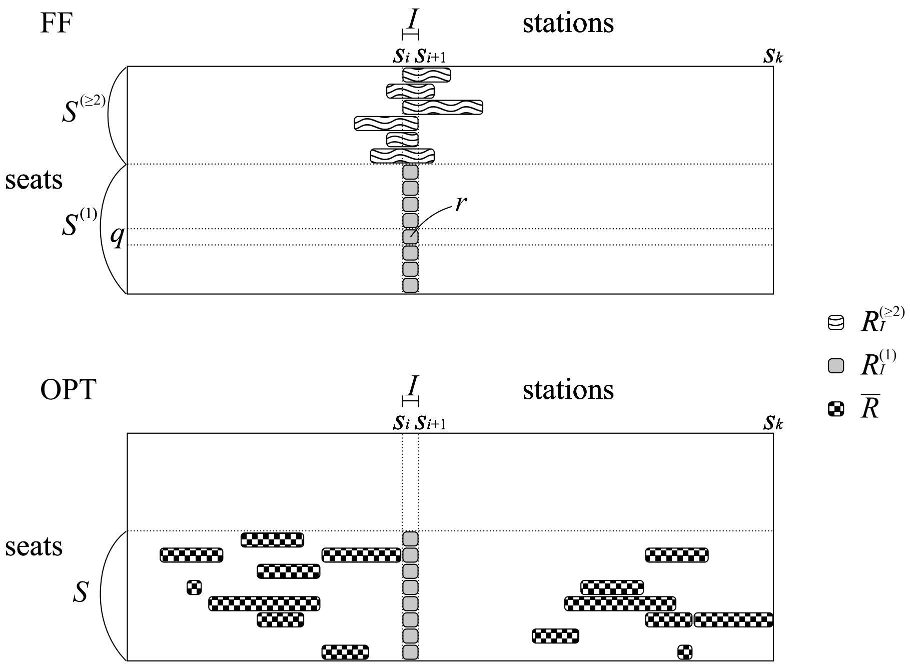

Fig. 5. Assignment configurations of FF and OPT for σ.

For the set X of requests, let $p(X)$ be the total price of tickets for requests in X. Then, the income of FF is at least $p(R_I^{(1)}) + p(R_I^{(\geq 2)}) + p(\overline{R}) \geq |S^{(1)}| + 2|S^{(\geq 2)}| + p(\overline{R}) = |S| + 2(n-|S|) + p(\overline{R})$ because $|S^{(1)}| = |S|$ and $|S^{(1)}| + |S^{(\geq 2)}| = n$. On the other hand, the income of OPT is at most $(k-1)(n-|S|)+|S|+p(\overline{R})$. So, we have that

$$\frac{p_{FF}(\sigma)}{p_{OPT}(\sigma)} \geq \frac{2(n-|S|)+|S|+p(\overline{R})}{(k-1)(n-|S|)+|S|+p(\overline{R})} \geq \frac{2}{k-1},$$

which completes the proof. □

4 Concluding Remarks

In this paper, we narrowed the gap between upper and lower bounds on the competitive ratios for the seat reservation problem for both the unit price and the proportional price problems. An apparent future work is to narrow the gaps between upper and lower bounds for both models.

To obtain a better bound for the unit price problem, we need to develop other algorithms as we discussed in this paper. For the proportional price problem, there still remains a gap between upper and lower bounds for First-Fit and Best-Fit (see Table 2). Narrowing the gap for these algorithms is one of the next possible challenges. We finally give a short remark on this direction.

Let us generalize the problem to a *loop-line*, namely, $s_k = s_1$. So, there could be a request for an interval $[j, i)$ $(j > i)$, which means that the passenger is to get on the train at station s_j and go to station s_i by way of station s_k. (Strictly speaking, we must consider the number of *laps*. However, here we consider the case of only one lap, e.g., intervals $[2, 4)$ and $[5, 3)$ overlap. This definition may not be practical, but is meaningful for the analysis of First-Fit and Best-Fit, as one can see below.) For this setting, we can derive a matching bound of $\frac{2}{k-1}$ for First-Fit and Best-Fit. The upper bound will be proved below, and the lower bound can be derived from exactly the same way as Theorem 4 because the proof of Theorem 4 holds for the loop-line model also. This suggests that to improve the lower bound for First-Fit and Best-Fit, we need arguments that do not hold for the loop-line model.

Upper bound proofs for First-Fit and Best-Fit for loop-line model. We give a proof for First-Fit (FF). The proof for Best-Fit is exactly the same. Let k be the number of stations and $n = 2m$ be the number of seats. We give the following sequences to FF: σ_1 consisting of m requests for $[1, 2)$; σ_2 consisting of m requests for $[2, 3)$; σ_3 consisting of m requests for $[1, 3)$; σ_4 consisting of m requests for $[2, k)$; and σ_5 consisting of m requests for $[3, 2)$. It is not hard to see that FF accommodates all the requests in σ_1, σ_2, and σ_3, but rejects all the requests in σ_4 and σ_5. So, the income of FF is $4m$. On the other hand, an optimal offline algorithm assigns requests in σ_1 and requests in σ_2 to different seats. Then it can reject all the requests of σ_3, and can accept all the requests in σ_4 and σ_5, so its income is $2m(k-1)$.

References

1. Bach, E., Boyar, J., Epstein, L., Favrholdt, L.M., Jiang, T., Larsen, K.S., Lin, G.-H., Van Stee, R.: Tight bounds on the competitive ratio on accommodating sequences for the seat reservation problem. Journal of Scheduling 6(2), 131–147 (2003)
2. Boyar, J., Favrholdt, L.M., Larsen, K.S., Nielsen, M.N.: Extending the accommodating function, Acta Informatica 40(1), 3–35 (2003)
3. Boyar, J., Krarup, S., Nielsen, M.N.: Seat reservation allowing seat changes, Journal of Algorithms 52(2), 169–192 (2004)
4. Boyar, J., Larsen, K.S.: The seat reservation problem, Algorithmica 25(4), 403–417 (1999)
5. Boyar, J., Larsen, K.S., Nielsen, M.N.: The accommodating function: a generalization of the competitive ratio, SIAM Journal on Computing 31(1), 233–258 (2001)
6. Boyar, J., Medvedev, P.: The relative worst order ratio applied to seat reservation, ACM Transactions on Algorithms 4(4), Article No. 48 (2008)
7. Kohrt, J.S., Larsen, K.S.: Online seat reservation via offline seating arrangements, International Journal of Foundations of Computer Science 16(2), 381–397 (2005)