Competitive Analysis for Two Variants of Online Metric Matching Problem*

Toshiya Itoh[1] [0000−0002−1149−7046], Shuichi Miyazaki[2] [0000−0003−0369−1970], and Makoto Satake[3]

[1] Department of Mathematical and Computing Science, Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro-ku, Tokyo 152-8550, Japan
titoh@c.titech.ac.jp
[2] Academic Center for Computing and Media Studies, Kyoto University, Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501, Japan
shuichi@media.kyoto-u.ac.jp
[3] Graduate School of Informatics, Kyoto University, Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501, Japan
satake@net.ist.i.kyoto-u.ac.jp

Abstract. In this paper, we study two variants of the online metric matching problem. The first problem is the online metric matching problem where all the servers are placed at one of two positions in the metric space. We show that a simple greedy algorithm achieves the competitive ratio of 3 and give a matching lower bound. The second problem is the online facility assignment problem on a line, where servers have capacities, servers and requests are placed on 1-dimensional line, and the distances between any two consecutive servers are the same. We show lower bounds $1+\sqrt{6}$ (> 3.44948), $\frac{4+\sqrt{73}}{3}$ (> 4.18133) and $\frac{13}{3}$ (> 4.33333) on the competitive ratio when the numbers of servers are 3, 4 and 5, respectively.

Keywords: Online algorithm, Competitive analysis, Online matching problem

1 Introduction

The online metric matching problem was introduced independently by Kalyanasundaram and Pruhs [7] and Khuller, Mitchell and Vazirani [10]. In this problem, n *servers* are placed on a given metric space. Then n *requests*, which are points on the metric space, are given to the algorithm one-by-one in an online fashion. The task of an online algorithm is to match each request immediately to one of n servers. If a request is matched to a server, then it incurs a cost which is equivalent to the distance between them. The goal of the problem is to minimize the sum of the costs. The papers [7] and [10] presented a deterministic online

* This work was partially supported by the joint project of Kyoto University and Toyota Motor Corporation, titled "Advanced Mathematical Science for Mobility Society" and JSPS KAKENHI Grant Numbers JP16K00017 and JP20K11677.

algorithm (called *Permutation* in [7]) and showed that it is $(2n-1)$-competitive and optimal.

In 1998, Kalyanasundaram and Pruhs [8] posed a question whether we can have a better competitive ratio by restricting the metric space to a line, and introduced the problem called the *online matching problem on a line*. Since then, this problem has been extensively studied, but there still remains a large gap between the best known lower bound 9.001 [5] and upper bound $O(\log n)$ [16] on the competitive ratio.

In 2020, Ahmed, Rahman and Kobourov [1] proposed a problem called the *online facility assignment problem* and considered it on a line, which we denote *OFAL* for short. In this problem, all the servers (which they call *facilities*) and requests (which they call *customers*) lie on a 1-dimensional line, and the distance between every pair of adjacent servers is the same. Also, each server has a *capacity*, which is the number of requests that can be matched to the server. In their model, all the servers are assumed to have the same capacity. Let us denote OFAL(k) the OFAL problem where the number of servers is k. Ahmed et al. [1] showed that for OFAL(k) the greedy algorithm is $4k$-competitive for any k and a deterministic algorithm *Optimal-fill* is k-competitive for any $k > 2$.

1.1 Our contributions

In this paper, we study a variant of the online metric matching problem where all the servers are placed at one of two positions in the metric space. This is equivalent to the case where there are two servers with capacities. We show that a simple greedy algorithm achieves the competitive ratio of 3 for this problem, and show that any deterministic online algorithm has competitive ratio at least 3.

We also study OFAL(k) for small k. Specifically, we show lower bounds $1+\sqrt{6}$ (> 3.44948), $\frac{4+\sqrt{73}}{3}$ (> 4.18133) and $\frac{13}{3}$ (> 4.33333) on the competitive ratio for OFAL(3), OFAL(4) and OFAL(5), respectively. We remark that our lower bounds $1+\sqrt{6}$ for OFAL(3) and $\frac{4+\sqrt{73}}{3}$ for OFAL(4) do not contradict the above-mentioned upper bound of Optimal-fill, since upper bounds by Ahmed et al. [1] are with respect to the *asymptotic* competitive ratio, while our lower bounds are with respect to the *strict* competitive ratio (see Sec. 2.3).

1.2 Related work

In 1990, Karp, Vazirani and Vazirani [9] first studied an online version of the matching problem. They studied the online matching problem on unweighted bipartite graphs with $2n$ vertices that contain a perfect matching, where the goal is to maximize the size of the obtained matching. In [9], they first showed that a deterministic greedy algorithm is $\frac{1}{2}$-competitive and optimal. They also presented a randomized algorithm *Ranking* and showed that it is $(1-\frac{1}{e})$-competitive and optimal. See [12] for a survey of the online matching problem.

As mentioned before, Kalyanasundaram and Pruhs [7] studied the online metric matching problem and showed that the algorithm *Permutation* is $(2n-1)$-competitive and optimal. Probabilistic algorithms for this problem were studied in [4, 13].

Kalyanasundaram and Pruhs [8] studied the online matching problem on a line. They gave two conjectures that the competitive ratio of this problem is 9 and that the *Work-Function* algorithm has a constant competitive ratio, both of which were later disproved in [11] and [5], respectively. This problem was studied in [2, 3, 6, 14–16], and the best known deterministic algorithm is the *Robust Matching* algorithm [15], which is $\Theta(\log n)$-competitive [14, 16].

Besides the problem on a line, Ahmed, Rahman and Kobourov [1] studied the online facility assignment problem on an unweighted graph $G(V, E)$. They showed that the greedy algorithm is $2|E|$-competitive and *Optimal-Fill* is $\frac{|E|k}{r}$-competitive, where $|E|$ is the number of edge of G and r is the radius of G.

2 Preliminaries

In this section, we give definitions and notations.

2.1 Online metric matching problem with two servers

We define the online metric matching problem with two servers, denoted OMM(2) for short. Let (X, d) be a metric space, where X is a (possibly infinite) set of points and $d(\cdot, \cdot)$ is a distance function. Let $S = \{s_1, s_2\}$ be a set of servers and $R = \{r_1, r_2, \ldots, r_n\}$ be a set of requests. A server s_i is characterized by the position $p(s_i) \in X$ and the capacity c_i that satisfies $c_1 + c_2 = n$. This means that s_i can be matched with at most c_i requests ($i = 1, 2$). A request r_i is also characterized by the position $p(r_i) \in X$.

S is given to an online algorithm in advance, while requests are given one-by-one from r_1 to r_n. At any time of the execution of an algorithm, a server is called *free* if the number of requests matched with it is less than its capacity, and *full* otherwise. When a request r_i is revealed, an online algorithm must match r_i with one of free servers. If r_i is matched with the server s_j, the pair (r_i, s_j) is added to the current matching and the cost $d(r_i, s_j)$ is incurred for this pair. The cost of the matching is the sum of the costs of all the pairs contained in it. The goal of OMM(2) is to minimize the cost of the final matching.

2.2 Online facility assignment problem on a line

We give the definition of the online facility assignment problem on a line with k servers, denoted OFAL(k). We state only differences from Sec. 2.1. The set of servers is $S = \{s_1, s_2, \ldots, s_k\}$ and all the servers have the same capacity ℓ, i.e., $c_i = \ell$ for all i. The number of requests must satisfy $n \leq \sum_{i=1}^{k} c_i = k\ell$. All the servers and requests are placed on a real number line, so their positions are expressed by a real, i.e., $p(s_i) \in \mathbb{R}$ and $p(r_j) \in \mathbb{R}$. Accordingly, the distance function is written as $d(r_i, s_j) = |p(r_i) - p(s_j)|$. We assume that the servers are placed in an increasing order of their indices, i.e., $p(s_1) \leq p(s_2) \leq \ldots \leq p(s_k)$. In this problem, any distance between two consecutive servers is the same, that is, $|p(s_i) - p(s_{i+1})| = d$ $(1 \leq i \leq k-1)$ for some constant d. Without loss of generality, we let $d = 1$.

2.3 Competitive ratio

To evaluate the performance of an online algorithm, we use the *strict competitive ratio*. (Hereafter, we omit "strict".) For an input σ, let $ALG(\sigma)$ and $OPT(\sigma)$ be the costs of the matchings obtained by an online algorithm ALG and an optimal offline algorithm OPT, respectively. Then the competitive ratio of ALG is the supremum of c that satisfies $ALG(\sigma) \leq c \cdot OPT(\sigma)$ for any input σ.

3 Online Metric Matching Problem with Two Servers

3.1 Upper bound

In this section, we define a greedy algorithm $GREEDY$ for OMM(2) and show that it is 3-competitive.

Definition 1. *When a request is given, $GREEDY$ matches it with the closest free server. If a given request is equidistant from the two servers and both servers are free, $GREEDY$ matches this request with s_1.*

In the following discussion, we fix an optimal offline algorithm OPT. If a request r is matched with the server s_x by $GREEDY$ and with s_y by OPT, we say that r is of *type* $\langle s_x, s_y \rangle$. We then define some properties of inputs.

Definition 2. *Let σ be an input to OMM(2). If every request in σ is matched with a different server by $GREEDY$ and OPT, σ is called* anti-opt.

Definition 3. *Let σ be an input to OMM(2). Suppose that $GREEDY$ matches its first request r_1 to the server $s_x \in \{s_1, s_2\}$. If $GREEDY$ matches r_1 through r_{c_x} to s_x (note that c_x is the capacity of s_x) and r_{c_x+1} through r_n to the other server s_{3-x}, σ is called* one-sided-priority.

By the following two lemmas, we show that it suffices to consider inputs that are anti-opt and one-sided-priority. For an input σ, we define $Rate(\sigma)$ as

$$Rate(\sigma) = \begin{cases} \dfrac{GREEDY(\sigma)}{OPT(\sigma)} & (\text{if } OPT(\sigma) \neq 0) \\ 1 & (\text{if } OPT(\sigma) = GREEDY(\sigma) = 0) \\ \infty & (\text{if } OPT(\sigma) = 0 \text{ and } GREEDY(\sigma) > 0) \end{cases}$$

Lemma 1. *For any input σ, there exists an anti-opt input σ' such that $Rate(\sigma') \geq Rate(\sigma)$.*

Proof. If σ is already anti-opt, we can set $\sigma' = \sigma$. Hence, in the following, we assume that σ is not anti-opt. Then there exists a request r in σ that is matched with the same server s_x by OPT and $GREEDY$. Let σ'' be an input obtained from σ by removing r and subtracting the capacity of s_x by 1. By this modification, neither OPT nor $GREEDY$ changes a matching for the remaining requests. Therefore, $GREEDY(\sigma'') = GREEDY(\sigma) - d(r, s_x)$ and $OPT(\sigma'') = OPT(\sigma) - d(r, s_x)$, which implies $Rate(\sigma'') \geq Rate(\sigma)$.

Let σ' be the input obtained by repeating this operation until the input sequence becomes anti-opt. Then σ' satisfies the conditions of this lemma. □

Lemma 2. *For any anti-opt input σ, there exists an anti-opt and one-sided-priority input σ' such that $Rate(\sigma') = Rate(\sigma)$.*

Proof. If σ is already one-sided-priority, we can set $\sigma' = \sigma$. Hence, in the following, we assume that σ is not one-sided-priority.

Since σ is anti-opt, σ contains only requests of type $\langle s_1, s_2\rangle$ or $\langle s_2, s_1\rangle$. Without loss of generality, assume that in execution of $GREEDY$, the server s_1 becomes full before s_2, and let r_t be the request that makes s_1 full (i.e., r_t is the last request of type $\langle s_1, s_2\rangle$).

Because σ is not one-sided-priority, σ includes at least one request r_i of type $\langle s_2, s_1\rangle$ before r_t. Let σ'' be the input obtained from σ by moving r_i to just after r_t. Since the set of requests is unchanged in σ and σ'', an optimal matching for σ is also optimal for σ'', so $OPT(\sigma'') = OPT(\sigma)$. In the following, we show that $GREEDY$ matches each request to the same server in σ and σ''. The sequence of requests up to r_{i-1} are the same in σ'' and σ, so the claim clearly holds for r_1 through r_{i-1}. The behavior of $GREEDY$ for r_{i+1} through r_t in σ'' is also the same for those in σ, because when serving these requests, both s_1 and s_2 are free in both σ and σ''. Just after serving r_t in σ'', s_1 becomes full, so $GREEDY$ matches $r_i, r_{t+1}, \ldots, r_n$ with s_2 in σ''. Note that these requests are also matched with s_2 in σ. Hence $GREEDY(\sigma'') = GREEDY(\sigma)$ and it results that $Rate(\sigma'') = Rate(\sigma)$. Note that σ'' remains anti-opt.

Let σ' be the input obtained by repeating this operation until the input sequence becomes one-sided-priority. Then σ' satisfies the conditions of this lemma. □

We can now prove the upper bound.

Theorem 1. *The competitive ratio of $GREEDY$ is at most 3 for OMM(2).*

Proof. By Lemma 1, it suffices to analyze only anti-opt inputs. In an anti-opt input, the number of requests of type $\langle s_1, s_2\rangle$ and that of type $\langle s_2, s_1\rangle$ are the same and the capacities of s_1 and s_2 are $n/2$ each. By Lemma 2, it suffices to analyze only the inputs where the first $n/2$ requests are of type $\langle s_1, s_2\rangle$ and the remaining $n/2$ requests are of type $\langle s_2, s_1\rangle$.

Let σ be an arbitrary such input. Then we have that

$$GREEDY(\sigma) = \sum_{i=1}^{n/2} d(r_i, s_1) + \sum_{i=n/2+1}^{n} d(r_i, s_2)$$

and

$$OPT(\sigma) = \sum_{i=1}^{n/2} d(r_i, s_2) + \sum_{i=n/2+1}^{n} d(r_i, s_1).$$

When serving $r_1, r_2, \ldots, r_{n/2}$, both servers are free but GREEDY matched them with s_1. Hence $d(r_i, s_1) \leq d(r_i, s_2)$ for $1 \leq i \leq n/2$. By the triangle inequality, we have $d(r_i, s_2) \leq d(s_1, s_2) + d(r_i, s_1)$ for $n/2 + 1 \leq i \leq n$. Again, by the triangle inequality, we have $d(s_1, s_2) \leq d(r_i, s_1) + d(r_i, s_2)$ for $1 \leq i \leq n$.

From these inequalities, we have that

$$\begin{aligned}
GREEDY(\sigma) &= \sum_{i=1}^{n/2} d(r_i, s_1) + \sum_{i=n/2+1}^{n} d(r_i, s_2) \\
&\leq \sum_{i=1}^{n/2} d(r_i, s_2) + \sum_{i=n/2+1}^{n} (d(s_1, s_2) + d(r_i, s_1)) \\
&= OPT(\sigma) + \frac{n}{2} d(s_1, s_2) \\
&= OPT(\sigma) + \frac{1}{2} \sum_{i=1}^{n} d(s_1, s_2) \\
&\leq OPT(\sigma) + \frac{1}{2} \sum_{i=1}^{n} (d(r_i, s_1) + d(r_i, s_2)) \\
&= OPT(\sigma) + \frac{1}{2} (OPT(\sigma) + GREEDY(\sigma)) \\
&= \frac{3}{2} OPT(\sigma) + \frac{1}{2} GREEDY(\sigma).
\end{aligned}$$

Thus $GREEDY(\sigma) \leq 3OPT(\sigma)$ and the competitive ratio of $GREEDY$ is at most 3. ☐

3.2 Lower bound

Theorem 2. *The competitive ratio of any deterministic online algorithm for OMM(2) is at least 3.*

Proof. We prove this lower bound on a 1-dimensional real line metric. Let $p(s_1) = -d$ and $p(s_2) = d$ for a constant d. Consider any deterministic algorithm ALG. First, our adversary gives $c_1 - 1$ requests at $p(s_1)$ and $c_2 - 1$ requests at $p(s_2)$. OPT matches the first $c_1 - 1$ requests with s_1 and the rest with s_2. If there exists a request that ALG matches differently from OPT, the adversary gives two more requests, one at $p(s_1)$ and the other at $p(s_2)$. Then, the cost of OPT is zero, while the cost of ALG is positive, so the ratio of them becomes infinity.

Next, suppose that ALG matches all these requests with the same server as OPT. Then the adversary gives the next request at the origin 0. Let s_x be the server that ALG matches this request with. Then OPT matches this request with the other server s_{3-x}. After that, the adversary gives the last request at $p(s_x)$. ALG has to match it with s_{3-x} and OPT matches it with s_x. The costs of ALG and OPT for this input are $3d$ and d, respectively. This completes the proof. □

4 Online Facility Assignment Problem on Line

In this section, we show lower bounds on the competitive ratio of OFAL(k) for $k = 3, 4$, and 5. To simplify the proofs, we use Definitions 4 and 5 and Proposition 1, observed in [3, 11], that allow us to restrict online algorithms to consider.

Definition 4. *When a request r is given, the* surrounding servers *for r are the closest free server to the left of r and the closest free server to the right of r.*

Definition 5. *If an algorithm ALG matches every request of an input σ with one of the surrounding servers, ALG is called* surrounding-oriented *for σ. If ALG is surrounding-oriented for any input, then ALG is called* surrounding-oriented.

Proposition 1. *For any algorithm ALG, there exists a surrounding-oriented algorithm ALG' such that $ALG'(\sigma) \leq ALG(\sigma)$ for any input σ.*

The proof of Proposition 1 is omitted in [3, 11], so we prove it here for completeness.

Proof. Suppose that ALG is not surrounding-oriented for σ. Then ALG matches at least one request of σ with a non-surrounding server. Let r be the earliest one among such requests and s be the server matched with r by ALG. Also let s' be the surrounding server (for r) on the same side as s and r' be the request matched with s' by ALG.

We modify ALG to ALG'' so that ALG'' matches r with s' and r' with s (and behaves the same as ALG for other requests). Without loss of generality, we can assume that $p(r) < p(s)$. Then we have that $p(r) \leq p(s') < p(s)$. If $p(r') \leq p(s')$, then $ALG''(\sigma) = ALG(\sigma)$ and if $p(r') > p(s')$, then $ALG''(\sigma) < ALG(\sigma)$. In either case, we have that $ALG''(\sigma) \leq ALG(\sigma)$.

Let ALG''' be the algorithm obtained by applying this modification as long as there is a request in σ matched with a non-surrounding server. Then $ALG'''(\sigma) \leq ALG(\sigma)$ and ALG''' is surrounding-oriented for σ.

We do the above modification for all the inputs for which ALG is not surrounding-oriented, and let ALG' be the resulting algorithm. Then $ALG'(\sigma) \leq ALG(\sigma)$ and ALG' is surrounding-oriented, as required. □

By Proposition 1, it suffices to consider only surrounding-oriented algorithms for lower bound arguments.

Theorem 3. *The competitive ratio of any deterministic online algorithm for OFAL(3) is at least $1 + \sqrt{6}$ (> 3.44948).*

Proof. Let ALG be any surrounding-oriented algorithm. Our adversary first gives $\ell - 1$ requests at $p(s_i)$ for each $i = 1, 2$ and 3. OPT matches every request r with the server at the same position $p(r)$. If ALG matches some request r with a server not at $p(r)$, then the adversary gives three more requests, one at each position of the server. The cost of ALG is positive and the cost of OPT is zero, so the ratio of the costs is infinity.

Next, suppose that ALG matches all these requests to the same server as OPT. Let $x = \sqrt{6} - 2$ ($\simeq 0.44949$) and $y = 3\sqrt{6} - 7$ ($\simeq 0.34847$). The adversary gives a request r_1 at $p(s_2) + x$.

Case 1. *ALG* matches r_1 with s_3.
See Fig. 1. The adversary gives the next request r_2 at $p(s_3)$. ALG matches it with s_2. Finally, the adversary gives a request r_3 at $p(s_1)$ and ALG matches it with s_1. The cost of ALG is $2 - x = 4 - \sqrt{6}$ and the cost of OPT is $x = \sqrt{6} - 2$. The ratio is $\frac{4-\sqrt{6}}{\sqrt{6}-2} = 1 + \sqrt{6}$.

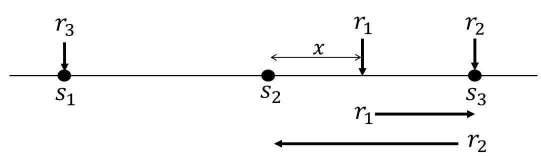

Fig. 1. Requests and ALG's matching for Case 1 of Theorem 3.

Case 2. *ALG* matches r_1 with s_2.
The adversary gives the next request r_2 at $p(s_2) - y$. We have two subcases.

Case 2-1. *ALG* matches r_2 with s_1.
See Fig. 2. The adversary gives a request r_3 at $p(s_1)$ and ALG matches it with s_3. The cost of ALG is $3+x-y = 8-2\sqrt{6}$ and the cost of OPT is $1-x+y = 2\sqrt{6}-4$. The ratio is $\frac{8-2\sqrt{6}}{2\sqrt{6}-4} = 1 + \sqrt{6}$.

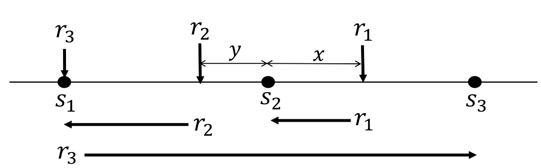

Fig. 2. Requests and ALG's matching for Case 2-1 of Theorem 3.

Case 2-2. *ALG* matches r_2 with s_3.
See Fig. 3. The adversary gives a request r_3 at $p(s_3)$ and ALG matches it with s_1.

The cost of ALG is $3+x+y = 4\sqrt{6}-6$ and the cost of OPT is $1+x-y = 6-2\sqrt{6}$. The ratio is $\frac{4\sqrt{6}-6}{6-2\sqrt{6}} = 1+\sqrt{6}$.

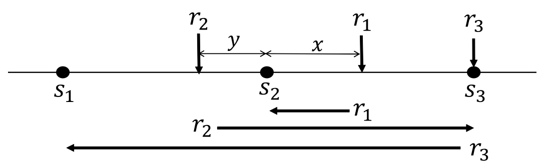

Fig. 3. Requests and ALG's matching for Case 2-2 of Theorem 3.

In any case, the ratio of ALG's cost to OPT's cost is $1+\sqrt{6}$. This completes the proof. □

Theorem 4. *The competitive ratio of any deterministic online algorithm for OFAL(4) is at least* $\frac{4+\sqrt{73}}{3}$ (> 4.18133).

Proof. Let ALG be any surrounding-oriented algorithm. In the same way as the proof of Theorem 3, the adversary first gives $\ell-1$ requests at $p(s_i)$ for $i = 1, 2, 3$, and 4, and we can assume that OPT and ALG match each of these requests to the server at the same position. Then, the adversary gives a request r_1 at $\frac{p(s_2)+p(s_3)}{2}$. Without loss of generality, assume that ALG matches it with s_2.

Let $x = \frac{10-\sqrt{73}}{2}$ $(\simeq 0.72800)$ and $y = \frac{11\sqrt{73}-93}{8}$ $(\simeq 0.12301)$. The adversary gives a request r_2 at $p(s_1)+x$. We consider two cases depending on the behavior of ALG.

Case 1. *ALG* matches r_2 with s_1.
See Fig. 4. The adversary gives the next request r_3 at $p(s_1)$. ALG has to match it with s_3. Finally, the adversary gives a request r_4 at $p(s_4)$ and ALG matches it with s_4. The cost of ALG is $\frac{5}{2}+x = \frac{15-\sqrt{73}}{2}$ and the cost of OPT is $\frac{3}{2}-x = \frac{\sqrt{73}-7}{2}$. The ratio is $\frac{15-\sqrt{73}}{\sqrt{73}-7} = \frac{4+\sqrt{73}}{3}$.

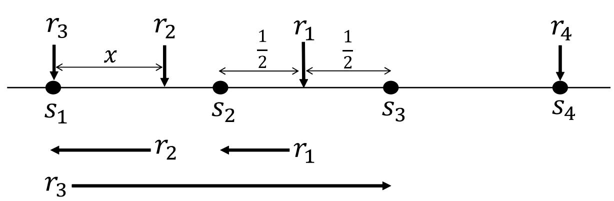

Fig. 4. Requests and ALG's matching for Case 1 of Theorem 4.

Case 2. *ALG* matches r_2 with s_3.
The adversary gives the next request r_3 at $p(s_3) + y$. We have two subcases.

Case 2-1. *ALG* matches r_3 with s_4.
See Fig. 5. The adversary gives a request r_4 at $p(s_4)$. *ALG* has to match it with s_1. The cost of *ALG* is $\frac{13}{2} - x - y = \frac{105-7\sqrt{73}}{8}$ and the cost of *OPT* is $\frac{1}{2} + x + y = \frac{7\sqrt{73}-49}{8}$. The ratio is $\frac{105-7\sqrt{73}}{7\sqrt{73}-49} = \frac{4+\sqrt{73}}{3}$.

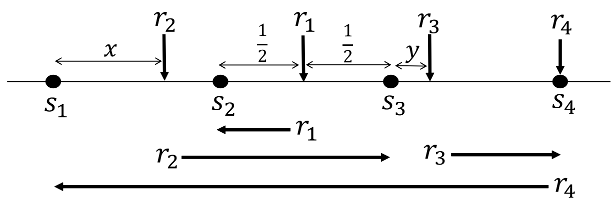

Fig. 5. Requests and *ALG*'s matching for Case 2-1 of Theorem 4.

Case 2-2. *ALG* matches r_3 with s_1.
See Fig. 6. The adversary gives a request r_4 at $p(s_1)$ and *ALG* has to match it with s_4. The cost of *ALG* is $\frac{15}{2} - x + y = \frac{15\sqrt{73}-73}{8}$ and the cost of *OPT* is $\frac{5}{2} - x - y = \frac{73-7\sqrt{73}}{8}$. The ratio is $\frac{15\sqrt{73}-73}{73-7\sqrt{73}} = \frac{4+\sqrt{73}}{3}$.

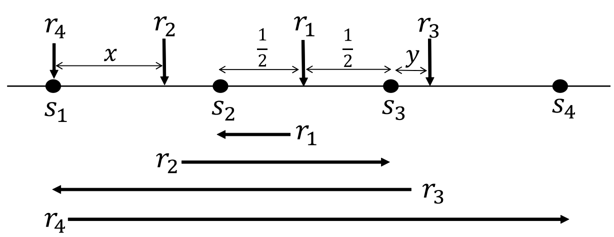

Fig. 6. Requests and *ALG*'s matching for Case 2-2 of Theorem 4.

In any case, the ratio of *ALG*'s cost to *OPT*'s cost is $\frac{4+\sqrt{73}}{3}$. This completes the proof. □

Theorem 5. *The competitive ratio of any deterministic online algorithms for OFAL(5) is at least* $\frac{13}{3}$ (> 4.33333).

Proof. Let *ALG* be any surrounding-oriented algorithm. In the same way as the proof of Theorem 3, the adversary first gives $\ell - 1$ requests at $p(s_i)$ for

$i = 1, 2, 3, 4$, and 5, and we can assume that OPT and ALG match each of these requests to the server at the same position.

Then, the adversary gives a request r_1 at $p(s_3)$. If ALG matches this with s_2 or s_4, the adversary gives the remaining requests at $p(s_1)$, $p(s_2)$, $p(s_4)$ and $p(s_5)$. OPT's cost is zero, while ALG's cost is positive, so the ratio is infinity. Therefore, assume that ALG matches r_1 with s_3. The adversary then gives a request r_2 at $p(s_3)$. Without loss of generality, assume that ALG matches it with s_2. Next, the adversary gives a request r_3 at $p(s_1) + \frac{7}{8}$. We consider two cases depending on the behavior of ALG.

Case 1. *ALG* matches r_3 with s_1.
See Fig. 7. The adversary gives the next request r_4 at $p(s_1)$. ALG has to match it with s_4. Finally, the adversary gives a request r_5 at $p(s_5)$ and ALG matches it with s_5. The cost of ALG is $\frac{39}{8}$ and the cost of OPT is $\frac{9}{8}$. The ratio is $\frac{13}{3}$.

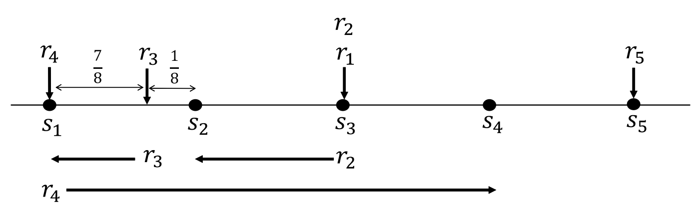

Fig. 7. Requests and ALG's matching for Case 1 of Theorem 5.

Case 2. *ALG* matches r_3 with s_4.
The adversary gives the next request r_4 at $p(s_4)$. We have two subcases.
Case 2-1. *ALG* matches r_4 with s_1.
See Fig. 8. The adversary gives a request r_5 at $p(s_1)$ and ALG has to match it with s_5. The cost of ALG is $\frac{81}{8}$ and the cost of OPT is $\frac{17}{8}$. The ratio is $\frac{81}{17} > \frac{13}{3}$.

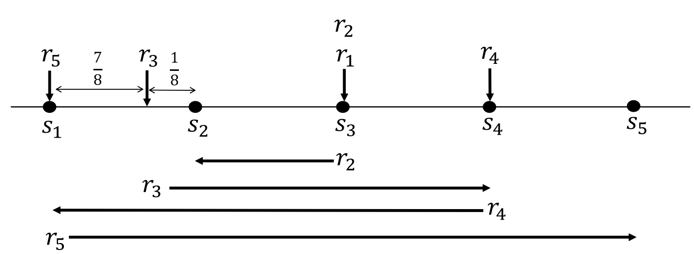

Fig. 8. Requests and ALG's matching for Case 2-1 of Theorem 5.

Case 2-2. *ALG* matches r_4 with s_5.
See Fig. 9. The adversary gives a request r_5 at $p(s_5)$ and ALG has to match it with s_1. The cost of ALG is $\frac{65}{8}$ and the cost of OPT is $\frac{15}{8}$. The ratio is $\frac{13}{3}$.

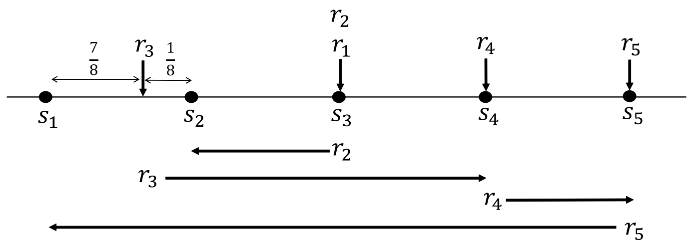

Fig. 9. Requests and ALG's matching for Case 2-2 of Theorem 5.

In any case, the ratio of ALG's cost to OPT's cost is at least $\frac{13}{3}$, which completes the proof. □

5 Conclusion

In this paper, we studied two variants of the online metric matching problem. The first is a restriction where all the servers are placed at one of two positions in the metric space. For this problem, we presented a greedy algorithm and showed that it is 3-competitive. We also proved that any deterministic online algorithm has competitive ratio at least 3, giving a matching lower bound. The second variant is the Online Facility Assignment Problem on a line with a small number of servers. We showed lower bounds on the competitive ratio $1 + \sqrt{6}$, $\frac{4+\sqrt{73}}{3}$, and $\frac{13}{3}$ when the numbers of servers are 3, 4, and 5, respectively.

One of the future work is to analyze the online metric matching problem with three or more server positions. Another interesting direction is to consider an optimal online algorithm for the Online Facility Assignment Problem on a line when the numbers of servers are 3, 4, and 5.

Acknowledgments. The authors would like to thank the anonymous reviewers for their valuable comments.

References

1. Ahmed, A., Rahman, M., Kobourov, S.: Online facility assignment. Theoretical Computer Science **806**, 455–467 (2020)
2. Antoniadis, A., Barcelo, M., Nugent, M., Pruhs, N., Scquizzato, M.: A $o(n)$-competitive deterministic algorithm for online matching on a line. WAOA 2012 pp. 11–22 (2014)

3. Antoniadis, A., Fischer, C., Tönnis, A.: A collection of lower bounds for online matching on the line. LATIN 2018 pp. 52–65 (2018)
4. Bansal, N., Buchbinder, N., Gupta, A., Naor, J.: An $o(\log^2 k)$-competitive algorithm for metric bipartite matching. ESA 2007 pp. 522–533 (2007)
5. Fuchs, B., Hochstättler, W., Kern, W.: Online matching on a line. Theoretical Computer Science **332**, 251–264 (2005)
6. Gupta, A., Lewi, K.: The online metric matching problem for doubling metrics. ICALP 2012 pp. 424–435 (2012)
7. Kalyanasundaram, B., Pruhs, K.: Online weighted matching. Journal of Algorithms **14**(3), 478–488 (1993)
8. Kalyanasundaram, B., Pruhs, K.: Online network optimization problems. Online Algorithms. Lecture Notes in Computer Science **1442**, 268–280 (1998)
9. Karp, R., Vazirani, U., Vazirani, V.: An optimal algorithm for on-line bipartite matching. STOC 1990 pp. 352–358 (1990)
10. Khuller, S., Mitchell, S., Vazirani, V.: On-line algorithms for weighted bipartite matching and stable marriages. Theoretical Computer Science **127**(2), 255–267 (1994)
11. Koutsoupias, E., Nanavati, A.: The online matching problem on a line. WAOA 2003 pp. 179–191 (2004)
12. Mehta, A.: Online matching and ad allocation. Theoretical Computer Science **8**(4), 265–368 (2012)
13. Meyerson, A., Nanavati, A., Poplawski, L.: Randomized online algorithms for minimum metric bipartite matching. SODA 2006 pp. 954–959 (2006)
14. Nayyar, K., Raghvendra, S.: An input sensitive online algorithm for the metric bipartite matching problem. FOCS 2017 pp. 505–515 (2017)
15. Raghvendra, S.: A robust and optimal online algorithm for minimum metric bipartite matching. APPROX/RANDOM 2016 **60**, 18:1–18:16 (2016)
16. Raghvendra, S.: Optimal analysis of an online algorithm for the bipartite matching problem on a line. SoCG 2018 pp. 67:1–67:14 (2018)