OPEN ACCESS
Discrete Mathematics, Algorithms and Applications
Vol. 13, No. 6 (2021) 2150156 (16 pages)

DOI: 10.1142/S1793830921501561

Competitive analysis for two variants of online metric matching problem

Toshiya Itoh

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

Shuichi Miyazaki

Academic Center for Computing and Media Studies
Kyoto University, Yoshida-Honmachi
Sakyo-ku, Kyoto 606-8501, Japan
shuichi@media.kyoto-u.ac.jp

Makoto Satake

Graduate School of Informatics, Kyoto University
Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501, Japan
satake@net.ist.i.kyoto-u.ac.jp

Received 29 January 2021
Revised 20 April 2021
Accepted 1 June 2021
Published 10 July 2021

Communicated by Dachuan Xu

In the online metric matching problem, there are servers on a given metric space and requests are given one-by-one. The task of an online algorithm is to match each request immediately and irrevocably with one of the unused servers. In this paper, we pursue competitive analysis for two variants of the online metric matching problem. The first variant is a restriction where each server is placed at one of two positions, which is denoted by OMM(2). We show that a simple greedy algorithm achieves the competitive ratio of 3 for OMM(2). We also show that this greedy algorithm is optimal by showing that the competitive ratio of any deterministic online algorithm for OMM(2) is at least 3. The second variant is the online facility assignment problem on a line. In this problem, the metric space is a line, the servers have capacities, and the distances between any two consecutive servers are the same. We denote this problem by OFAL(k), where k is the number of servers. We first observe that the upper and lower bounds for OMM(2) also hold for OFAL(2), so the competitive ratio for OFAL(2) is exactly 3. We then show lower bounds on the competitive ratio $1+\sqrt{6}$ (>3.44948), $\frac{4+\sqrt{73}}{3}$ (>4.18133) and $\frac{13}{3}$ (>4.33333) for OFAL(3), OFAL(4) and OFAL(5), respectively.

Keywords: Online algorithm; competitive analysis; online metric matching; online facility assignment problem.

Mathematics Subject Classification 2020: 68W27, 68W40

1. Introduction

Online problems capture the nature of real-time computation, in which pieces of input, generally called *requests*, are given to an algorithm one-by-one, and an online algorithm must decide how to deal with the current request before receiving the next one. This decision is irrevocable in that an algorithm may not change it later. The performance of online algorithms are typically measured by competitive analysis, which was initiated by Sleator and Tarjan [27] who applied it to the list update problem and the paging problem. Informally speaking, an online algorithm A is *c-competitive* (or the *competitive ratio* of A is at most c) if the cost of A's output is at most c times worse than the optimal cost.

Kalyanasundaram and Pruhs [14] and Khuller *et al.* [17] independently introduced and studied the *online metric matching problem*, which is an online variant of the minimum cost bipartite matching problem. 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 with one of n servers. If a request is matched with 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. Papers [14, 17] presented a deterministic online algorithm (called *Permutation* in [14]) and showed that it is $(2n-1)$-competitive and optimal.

In 1998, Kalyanasundaram and Pruhs [15] posed a question whether one can have a better competitive ratio by restricting the metric space to a line (1-dimensional Euclidean space), and introduced the problem called 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 [12, 18], respectively. This problem has been extensively studied [2, 3, 13, 23, 25, 26] and the currently best upper bound is $O(\log n)$ [23, 26] achieved by the *Robust Matching* algorithm [25]. The best lower bound had been 9.001 [12] for more than 15 years, but very recently it was improved to $\Omega(\sqrt{\log n})$ [24].

In 2020, Ahmed *et al.* [1] proposed a problem called the *online facility assignment problem* and considered it on a line, which we denote by *OFAL* for short. In this problem, all the servers (which they call *facilities*) and requests (which they call *customers*) lie on a 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 by OFAL(k) the OFAL problem where the number of servers is k. Ahmed *et al.* [1] showed that for any k, a greedy algorithm is $4k$-competitive for OFAL(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 and the online facility assignment problem when the number of servers is a small constant.

We first consider the online metric matching problem where all the servers are placed at one of two positions in the metric space, which we denote by OMM(2). 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 OMM(2). To do so, we first give two properties that the worst case inputs satisfy, and show that the competitive ratio of the greedy algorithm is at most 3 for such inputs. We also show that any deterministic online algorithm for OMM(2) has a competitive ratio at least 3, giving a matching lower bound.

We also study OFAL(k) for small k. We first remark that the above results for OMM(2) hold also for OFAL(2), which implies a matching upper and lower bound on the competitive ratio of 3 for OFAL(2). We then show lower bounds on the competitive ratio for OFAL(k) when $k = 3, 4$, and 5. 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 by Ahmed *et al.* [1], since their upper bounds 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*

As mentioned before, Kalyanasundaram and Pruhs [14] 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 [7, 21].

Besides the problem on a line, Ahmed *et al.* [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 edges of G and r is the radius of G. Recently, Muttakee *et al.* [22] presented new results for the online facility assignment problem. They showed competitive ratios of the greedy algorithm and Optimal-Fill for grid graphs and Optimal-Fill for arbitrary graphs. They also studied competitiveness for plane metric and line metric.

Recently, an extension of the online metric matching problem that allows delay has been studied enthusiastically. There, an online algorithm is allowed to defer a decision for a given request at the cost of a "time cost" incurred depending on the waiting time. The goal of the problem is to minimize the sum of a matching cost plus all the time costs. This problem was first considered by Emek *et al.* [10]. Randomized algorithms were studied in [4, 5, 10, 19], and the current best upper and lower bounds on the competitive ratio are $O(\log n)$ [5] and $\Omega(\sqrt{\frac{\log n}{\log\log n}})$ [4], respectively, where n is the number of points in the metric space. Deterministic algorithms were studied in [6, 8, 9, 11] and the best known upper bound on the competitive ratio is $O(m^{\log_2(\frac{3}{2}+\epsilon)}) \simeq O(m^{0.59})$ [6], where m is the number of requests.

A different version of an online matching problem was initiated by Karp *et al.* [16]. In this model, they considered a bipartite graph, where the vertices in one partition L are given in advance, and the vertices in the other side R are given one-by-one with edges incident to vertices in L. The task of an algorithm is to match an arriving vertex with one of unmatched vertices of L or leave it unmatched, and the goal is to maximize the size of the final matching. Since this problem has application to ad auction, several variants have been studied in decades. See [20] for a survey.

2. Preliminaries

In Secs. 2.1 and 2.2, we give definitions of the two problems we study and in Sec. 2.3 we give the definition of the competitive ratio.

2.1. *Online metric matching problem with two servers*

In this section, we define the online metric matching problem with two servers, denoted by 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 $p(r_i) \in X$.

The server set 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 so far 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*

In this section, we give the definition of the online facility assignment problem on a line with k servers, denoted by OFAL(k). 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., $p(s_1) \leq p(s_2) \leq \cdots \leq p(s_k)$. 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 $d = 1$.

The task of an online algorithm and the goal of the problem is the same as OMM(2): The server set is initially known to an algorithm. When receiving a request, the algorithm must match it with one of free servers, incurring a cost of the distance between matched server and request. The purpose of the algorithm is to minimize the matching cost.

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(σ) and OPT(σ) 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 (for a minimization problem) is the infimum of c that satisfies $\text{ALG}(\sigma) \leq c \cdot \text{OPT}(\sigma)$ for any input σ. The competitive ratio is at least 1, and an algorithm with smaller competitive ratio is better.

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 3.1. 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 two properties of inputs.

Definition 3.2. Let σ be an input to OMM(2). If every request in σ is matched with a different server by GREEDY and OPT, namely if each request is of type $\langle s_1, s_2 \rangle$ or $\langle s_2, s_1 \rangle$, then σ is called *anti-opt*.

Definition 3.3. Let σ be an input to OMM(2). Suppose that GREEDY matches its first request r_1 with the server $s_x \in \{s_1, s_2\}$. If GREEDY matches r_1 through r_{c_x} with s_x (note that c_x is the capacity of s_x) and r_{c_x+1} through r_n with the other server s_{3-x}, then σ is called *one-sided-priority*.

By the following two lemmas, we show that, to prove an upper bound on the competitive ratio of GREEDY, it suffices to consider inputs that are anti-opt and one-sided-priority. For an input σ, we define Rate(σ) as

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

Lemma 3.4. *For any input σ, there exists an anti-opt input σ' such that* $\mathrm{Rate}(\sigma') \geq \mathrm{Rate}(\sigma)$.

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, $\mathrm{GREEDY}(\sigma'') = \mathrm{GREEDY}(\sigma) - d(r, s_x)$ and $\mathrm{OPT}(\sigma'') = \mathrm{OPT}(\sigma) - d(r, s_x)$. If $\mathrm{OPT}(\sigma'') > 0$, then clearly $\mathrm{OPT}(\sigma) > 0$ and hence

$$\begin{aligned} \mathrm{Rate}(\sigma'') &= \frac{\mathrm{GREEDY}(\sigma'')}{\mathrm{OPT}(\sigma'')} \\ &= \frac{\mathrm{GREEDY}(\sigma) - d(r, s_x)}{\mathrm{OPT}(\sigma) - d(r, s_x)} \\ &\geq \frac{\mathrm{GREEDY}(\sigma)}{\mathrm{OPT}(\sigma)} \\ &= \mathrm{Rate}(\sigma). \end{aligned}$$

If $\mathrm{OPT}(\sigma'') = 0$ and $\mathrm{GREEDY}(\sigma'') > 0$, then $\mathrm{Rate}(\sigma'') = \infty$ and $\mathrm{Rate}(\sigma'') \geq \mathrm{Rate}(\sigma)$ holds. If $\mathrm{OPT}(\sigma'') = \mathrm{GREEDY}(\sigma'') = 0$, then $\mathrm{OPT}(\sigma) = \mathrm{GREEDY}(\sigma) = d(r, s_x)$. In this case, $\mathrm{Rate}(\sigma'') = \mathrm{Rate}(\sigma) = 1$. Thus in all cases $\mathrm{Rate}(\sigma'') \geq \mathrm{Rate}(\sigma)$ holds.

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

Proof. If σ is already one-sided-priority, we can set $\sigma' = \sigma$ and we are done. 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 $\mathrm{OPT}(\sigma'') = \mathrm{OPT}(\sigma)$. In the following, we show that GREEDY matches each request with the same server in σ and σ''. The sequence of requests up to r_{i-1} is unchanged, 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 $\mathrm{GREEDY}(\sigma'') = \mathrm{GREEDY}(\sigma)$ and it results that $\mathrm{Rate}(\sigma'') = \mathrm{Rate}(\sigma)$. 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 3.6. *The competitive ratio of* GREEDY *is at most* 3 *for OMM*(2).

Proof. By Lemma 3.4, 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 3.5, 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

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

and

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

When serving $r_1, r_2, \ldots, r_{n/2}$, both servers are free but GREEDY matched them with s_1. Hence (1) $d(r_i, s_1) \le d(r_i, s_2)$ holds for $1 \le i \le n/2$. By the triangle inequality, we have that (2) $d(r_i, s_2) \le d(s_1, s_2) + d(r_i, s_1)$ for $n/2 + 1 \le i \le n$. Again, by the triangle inequality, we have that (3) $d(s_1, s_2) \le d(r_i, s_1) + d(r_i, s_2)$ for $1 \le i \le n$.

From these inequalities, we have that

$$\begin{aligned}\mathrm{GREEDY}(\sigma) &= \sum_{i=1}^{n/2} d(r_i, s_1) + \sum_{i=n/2+1}^{n} d(r_i, s_2)\\ &\le \sum_{i=1}^{n/2} d(r_i, s_2) + \sum_{i=n/2+1}^{n} (d(s_1, s_2) + d(r_i, s_1)) \quad \text{(by (1) and (2))}\end{aligned}$$

$$= \mathrm{OPT}(\sigma) + \frac{n}{2} d(s_1, s_2)$$

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

$$\leq \mathrm{OPT}(\sigma) + \frac{1}{2} \sum_{i=1}^{n} (d(r_i, s_1) + d(r_i, s_2)) \quad \text{(by (3))}$$

$$= \mathrm{OPT}(\sigma) + \frac{1}{2} (\mathrm{OPT}(\sigma) + \mathrm{GREEDY}(\sigma))$$

$$= \frac{3}{2} \mathrm{OPT}(\sigma) + \frac{1}{2} \mathrm{GREEDY}(\sigma).$$

Thus $\mathrm{GREEDY}(\sigma) \leq 3\mathrm{OPT}(\sigma)$ and the competitive ratio of GREEDY is at most 3. □

3.2. *Lower bound*

We now give a matching lower bound.

Theorem 3.7. *The competitive ratio of any deterministic online algorithm for OMM*(2) *is at least* 3.

Proof. We prove this lower bound on a line metric. We set the positions of servers as $p(s_1) = -d$ 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 $p(s_2)$. An optimal offline algorithm 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 and the competitive ratio is unbounded.

Next, suppose that ALG matches all these requests with the same server as OPT. Then the adversary gives the next request r at the origin 0. Let s_x be the server that ALG matches r with. Then, the adversary gives the last request r' at $p(s_x)$. ALG matches it with s_{3-x} and its total cost is $3d$. On the other hand, OPT matches r with s_{3-x} and r' with s_x, so its cost is d. This completes the proof. □

4. Online Facility Assignment Problem on Line

Since OFAL(2) is a special case of OMM(2), Theorem 3.6 applies also for OFAL(2). Further, recall that, in the proof of Theorem 3.7, adversarial requests are constructed on line metric. Hence this proof is also valid for OFAL(2). Thus we have the following corollary.

Corollary 4.1. (i) *The competitive ratio of* GREEDY *is at most* 3 *for OFAL*(2). (ii) *The competitive ratio of any deterministic online algorithm for OFAL*(2) *is at least* 3.

Next, we show lower bounds on the competitive ratio of OFAL(k) for $k = 3, 4$, and 5. To simplify the proofs, we use Definitions 4.2 and 4.3 and Proposition 4.4 from [3, 18], that allow us to restrict online algorithms to consider.

Definition 4.2. When a request r is given, the *surrounding servers for* r are the closest free server to the left of r (if any) and the closest free server to the right of r (if any).

Definition 4.3. 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 4.4. *For any algorithm* ALG, *there exists a surrounding-oriented algorithm* ALG$'$ *such that* ALG$'(\sigma) \leq$ ALG(σ) *for any input* σ.

The proof idea is given in [3], but for completeness, we formally prove it here.

Proof. Suppose that ALG is not surrounding-oriented and let σ be an input for which ALG is not surrounding-oriented. 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. Without loss of generality, we can assume that $p(r) < p(s)$. Also, let s' be the surrounding server (for r) on the same side as s. Then we have that $p(r) \leq p(s') < p(s)$. Finally, let 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). If $p(r') \leq p(s')$, then ALG$''(\sigma) =$ ALG(σ) (Fig. 1). If $p(r') > p(s')$, then ALG$''(\sigma) <$ ALG(σ) (Fig. 2). In either case, we have that ALG$''(\sigma) \leq$ ALG(σ).

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(σ) and ALG$'''$ is surrounding-oriented for σ.

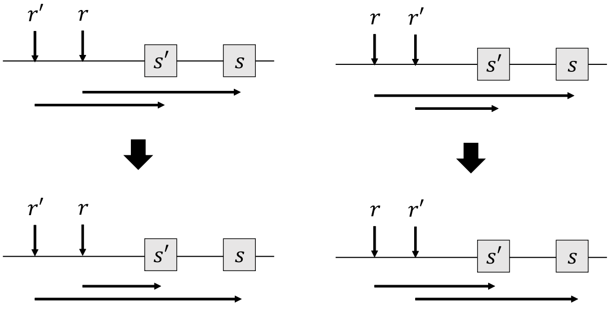

Fig. 1. Modifying the matching in the proof of Proposition 4.4(1).

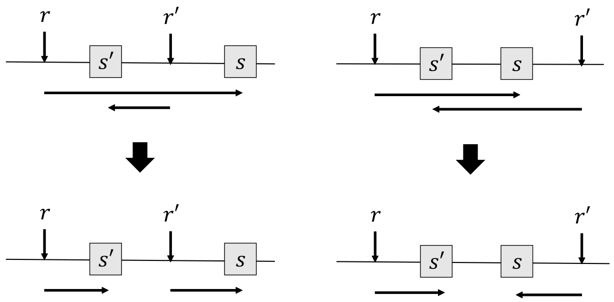

Fig. 2. Modifying the matching in the proof of Proposition 4.4(2).

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

By Proposition 4.4, it suffices to consider only surrounding-oriented algorithms for lower bound arguments.

Theorem 4.5. *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 with the same server as OPT. Let $x = \sqrt{6} - 2$ ($\simeq$0.44949). The adversary gives a request r_1 at $p(s_2) + x$.

Case 1. ALG matches r_1 with s_3.

See Fig. 3. 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 $\frac{4-\sqrt{6}}{\sqrt{6}-2} = 1 + \sqrt{6}$.

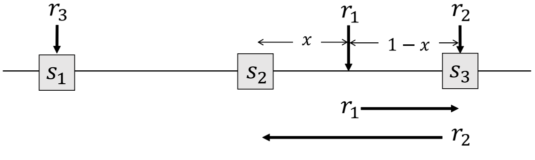

Fig. 3. Requests and ALG's matching for Case 1 of Theorem 4.5.

Case 2. ALG matches r_1 with s_2.

Let $y = 3\sqrt{6} - 7$ ($\simeq$0.34847). 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. 4. The adversary gives a request r_3 at $p(s_1)$ and ALG matches it with s_3. The cost of ALG is $3 + x - y = 8 - 2\sqrt{6}$. OPT matches r_1, r_2, and r_3 with s_3, s_2, and s_1, respectively, and its cost is $1 - x + y = 2\sqrt{6} - 4$. The ratio is $\frac{8-2\sqrt{6}}{2\sqrt{6}-4} = 1 + \sqrt{6}$.

Case 2-2. ALG matches r_2 with s_3.

See Fig. 5. The adversary gives a request r_3 at $p(s_3)$ and ALG matches it with s_1. The cost of ALG is $3 + x + y = 4\sqrt{6} - 6$. OPT matches r_1, r_2, and r_3 with s_2, s_1, and s_3, respectively, and its cost is $1 + x - y = 6 - 2\sqrt{6}$. The ratio is $\frac{4\sqrt{6}-6}{6-2\sqrt{6}} = 1 + \sqrt{6}$.

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

Theorem 4.6. *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 4.5, the adversary first gives $\ell - 1$ requests at $p(s_i)$ for each $i = 1, 2, 3$, and 4, and we can assume that OPT and ALG match each of these requests with 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). The adversary gives a request r_2 at $p(s_1) + x$. We consider two cases depending on the behavior of ALG.

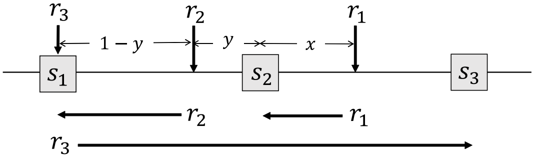

Fig. 4. Requests and ALG's matching for Case 2-1 of Theorem 4.5.

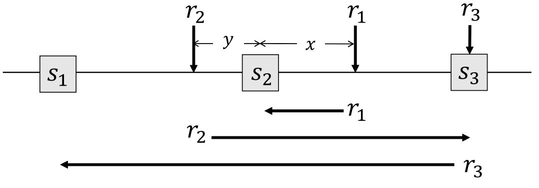

Fig. 5. Requests and ALG's matching for Case 2-2 of Theorem 4.5.

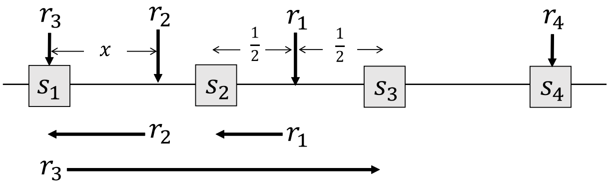

Fig. 6. Requests and ALG's matching for Case 1 of Theorem 4.6.

Case 1. ALG matches r_2 with s_1.

See Fig. 6. The adversary gives the next request r_3 at $p(s_1)$. Then 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 $\frac{5}{2} + x = \frac{15-\sqrt{73}}{2}$. OPT matches r_1, r_2, r_3, and r_4 with s_3, s_2, s_1, and s_4, respectively, and its cost is $\frac{3}{2} - x = \frac{\sqrt{73}-7}{2}$. The ratio is $\frac{15-\sqrt{73}}{\sqrt{73}-7} = \frac{4+\sqrt{73}}{3}$.

Case 2. ALG matches r_2 with s_3.

Let $y = \frac{11\sqrt{73}-93}{8}$ ($\simeq$0.12301). 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. 7. The adversary gives a request r_4 at $p(s_4)$. ALG has to match it with s_1. The cost of ALG is $\frac{13}{2} - x - y = \frac{105-7\sqrt{73}}{8}$. OPT matches r_1, r_2, r_3, and r_4 with s_2, s_1, s_3, and s_4, respectively, and its cost is $\frac{1}{2} + x + y = \frac{7\sqrt{73}-49}{8}$. The ratio is $\frac{105-7\sqrt{73}}{7\sqrt{73}-49} = \frac{4+\sqrt{73}}{3}$.

Case 2-2. ALG matches r_3 with s_1.

See Fig. 8. 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 $\frac{15}{2} - x + y = \frac{15\sqrt{73}-73}{8}$. OPT matches r_1, r_2, r_3, and r_4 with s_3, s_2, s_4, and s_1, respectively, and its cost is $\frac{5}{2} - x - y = \frac{73-7\sqrt{73}}{8}$. The ratio is $\frac{15\sqrt{73}-73}{73-7\sqrt{73}} = \frac{4+\sqrt{73}}{3}$.

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

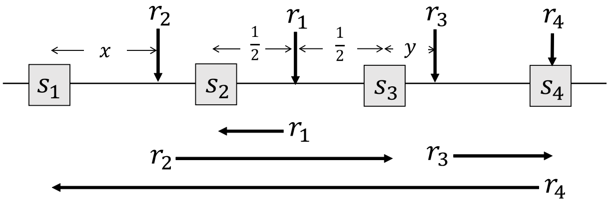

Fig. 7. Requests and ALG's matching for Case 2-1 of Theorem 4.6.

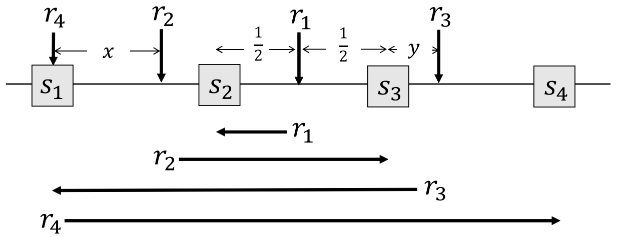

Fig. 8. Requests and ALG's matching for Case 2-2 of Theorem 4.6.

Theorem 4.7. *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 4.5, the adversary first gives $\ell - 1$ requests at $p(s_i)$ for each $i = 1, 2, 3, 4$, and 5, and we can assume that OPT and ALG match each of these requests with 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. 9. 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 $\frac{39}{8}$. OPT matches r_1, r_2, r_3, r_4, and r_5 with s_3, s_4, s_2, s_1, and s_5, respectively, and its cost is $\frac{9}{8}$. The ratio is $\frac{13}{3}$.

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.

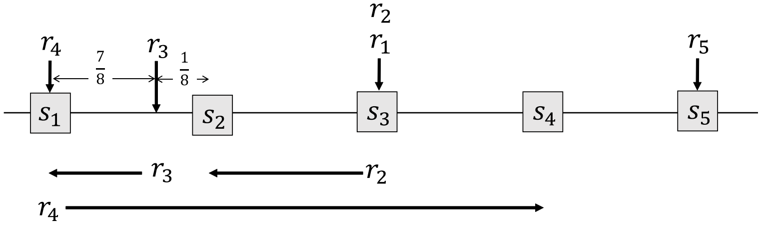

Fig. 9. Requests and ALG's matching for Case 1 of Theorem 4.7.

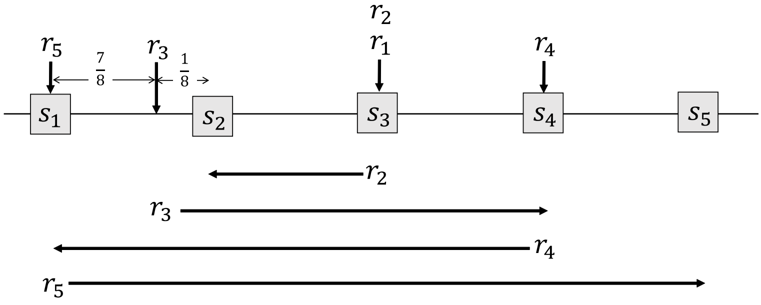

Fig. 10. Requests and ALG's matching for Case 2-1 of Theorem 4.7.

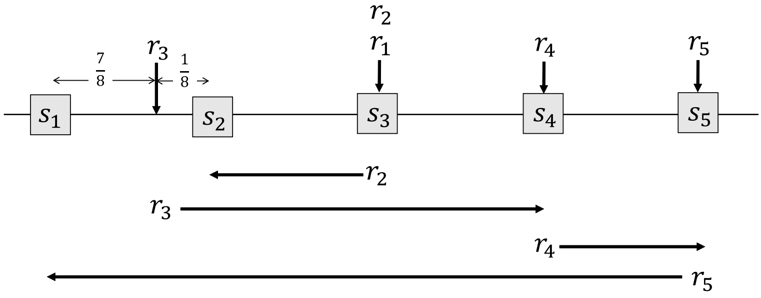

Fig. 11. Requests and ALG's matching for Case 2-2 of Theorem 4.7.

Case 2-1. ALG matches r_4 with s_1.

See Fig. 10. 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 $\frac{81}{8}$. OPT matches r_1, r_2, r_3, r_4, and r_5 with s_3, s_5, s_2, s_4, and s_1, respectively, and its cost is $\frac{17}{8}$. The ratio is $\frac{81}{17} > \frac{13}{3}$.

Case 2-2. ALG matches r_4 with s_5.

See Fig. 11. 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 $\frac{65}{8}$. OPT matches r_1, r_2, r_3, r_4, and r_5 with s_3, s_2, s_1, s_4, and s_5, respectively, and its cost is $\frac{15}{8}$. The ratio is $\frac{13}{3}$.

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 k servers, denoted by

OFAL(k). We investigated this problem when k is small. We first showed, as a corollary of the first result, that the competitive ratio is 3 for OFAL(2). We also showed lower bounds on the competitive ratio $1+\sqrt{6}$ (>3.44948), $\frac{4+\sqrt{73}}{3}$ (>4.18133) and $\frac{13}{3}$ (>4.33333) for OFAL(3), OFAL(4), and OFAL(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 derive upper bounds for OFAL(3), OFAL(4), and OFAL(5). The same argument as [1] shows that the competitive ratio of GREEDY is no better than $4k - 5$ for OFAL(k), which is far from our lower bounds. Hence, we need to device a new algorithm if our lower bounds are close to optimal.

Acknowledgments

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. A preliminary version of this paper appeared in the proceedings of the 14th Annual International Conference on Combinatorial Optimization and Applications (COCOA 2020).

References

[1] A. R. Ahmed, M. S. Rahman and S. G. Kobourov, Online facility assignment, *Theor. Comput. Sci.* **806** (2020) 455–467.

[2] A. Antoniadis, N. Barcelo, M. Nugent, K. Pruhs and M. Scquizzato, A $o(n)$-competitive deterministic algorithm for online matching on a line, in *Proc. Int. Workshop on Approximation and Online Algorithms* (Springer, 2014), pp. 11–22.

[3] A. Antoniadis, C. Fischer and A. Tönnis, A collection of lower bounds for online matching on the line, in *Proc. LATIN* (Springer, 2018), pp. 52–65.

[4] I. Ashlagi, Y. Azar, M. Charikar, A. Chiplunkar, O. Geri, H. Kaplan, R. Makhijani, Y. Wang and R. Wattenhofer, Min-cost bipartite perfect matching with delays, in *Proc. APPROX/RANDOM* (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017), pp. 1:1–1:20.

[5] Y. Azar, A. Chiplunkar and H. Kaplan, Polylogarithmic bounds on the competitiveness of min-cost perfect matching with delays, in *Proc. Symp. Discrete Algorithms* (ACM, 2017), pp. 1051–1061.

[6] Y. Azar and A. J. Fanani, Deterministic min-cost matching with delays, in *Proc. Int. Workshop on Approximation and Online Algorithms* (Springer, 2018), pp. 21–35.

[7] N. Bansal, N. Buchbinder, A. Gupta and J. Naor, An $o(\log^2 k)$-competitive algorithm for metric bipartite matching, in *Proc. European Symp. Algorithms* (Springer, 2007), pp. 522–533.

[8] M. Bienkowski, A. Kraska, H. Liu and P. Schmidt, A primal-dual online deterministic algorithm for matching with delays, in *Proc. Int. Workshop on Approximation and Online Algorithms* (Springer, 2018), pp. 51–68.

[9] M. Bienkowski, A. Kraska and P. Schmidt, A match in time saves nine: Deterministic online matching with delays, in *Proc. Int. Workshop on Approximation and Online Algorithms* (Springer, 2017), pp. 132–146.

[10] Y. Emek, S. Kutten and R. Wattenhofer, Online matching: Haste makes waste! in *Proc. Symp. Theory of Computing* (ACM, 2016), pp. 333–334.

[11] Y. Emek, Y. Shapiro and Y. Wang, Minimum cost perfect matching with delays for two sources, *Int. Conf. Algorithms and Complexity* (Springer, 2017), pp. 209–221.

[12] B. Fuchs, W. Hochstättler and W. Kern, Online matching on a line, *Theor. Comput. Sci.* **332** (2005) 251–264.

[13] A. Gupta and K. Lewi, The online metric matching problem for doubling metrics, *Int. Colloquium on Automata, Languages, and Programming* (Springer, 2012), pp. 424–435.

[14] B. Kalyanasundaram and K. Pruhs, Online weighted matching, *J. Algorithms* **14**(3) (1993) 478–488.

[15] B. Kalyanasundaram and K. Pruhs, Online network optimization problems, *Online Algorithms*, Lecture Notes in Computer Science, Vol. 1442 (Springer, 1998), pp. 268–280.

[16] R. M. Karp, U. V. Vazirani and V. V. Vazirani, An optimal algorithm for on-line bipartite matching, in *Proc. Symp. Theory of Computing* (ACM, 1990), pp. 352–358.

[17] S. Khuller, S. G. Mitchell and V. V. Vazirani, On-line algorithms for weighted bipartite matching and stable marriages, *Theoretical Computer Science* **127**(2) (1994) 255–267.

[18] E. Koutsoupias and A. Nanavati, The online matching problem on a line, in *Proc. Int. Workshop on Approximation and Online Algorithms* (Springer, 2004), pp. 179–191.

[19] X. Liu, Z. Pan, Y. Wang and R. Wattenhofer, Impatient online matching, in *Proc. Int. Symp. Algorithms and Computation* (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018), pp. 62:1–62:12.

[20] A. Mehta, Online matching and ad allocation, *Found. Trends Theor. Comput. Sci.* **8**(4) (2012) 265–368.

[21] A. Meyerson, A. Nanavati and L. J. Poplawski, Randomized online algorithms for minimum metric bipartite matching, in *Proc. Symp. Discrete Algorithm* (ACM, 2006), pp. 954–959.

[22] S. A. Muttakee, A. R. Ahmed and M. S. Rahman, New results and bounds on online facility assignment problem, preprint (2020), arXiv:2009.01446.

[23] K. Nayyar and S. Raghvendra, An input sensitive online algorithm for the metric bipartite matching problem, in *Proc. Symp. Foundations of Computer Science* (IEEE, 2017), pp. 505–515.

[24] E. Peserico and M. Scquizzato, Matching on the line admits no $o(\sqrt{\log n})$-competitive algorithm, in *Proc. Int. Colloquium on Automata, Languages, and Programming* (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2021), pp. 103:1–103:3.

[25] S. Raghvendra, A robust and optimal online algorithm for minimum metric bipartite matching, in *Proc. APPROX/RANDOM*, Vol. 60 (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016), pp. 18:1–18:16.

[26] S. Raghvendra, Optimal analysis of an online algorithm for the bipartite matching problem on a line, in *Proc. Int. Symp. Computational Geometry* (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018), pp. 67:1–67:14.

[27] D. D. Sleator and R. E. Tarjan, Amortized efficiency of list update and paging rules, *Commun. ACM* **28**(2) (1985) 202–208.