computation

MDPI

Article

Online Bottleneck Matching Problem with Two Heterogeneous Sensors in a Metric Space

Man Xiao *, **Yaru Yang** and **Weidong Li**

School of Mathematics and Statistics, Yunnan University, Kunming 650504, China
* Correspondence: man1205@163.com

Abstract: In this paper, we consider the online matching problem with two heterogeneous sensors s_1 and s_2 in a metric space (X, d). If a request r is assigned to sensor s_1, the service cost of r is the distance $d(r, s_1)$. Otherwise, r is assigned to sensor s_2, and the service cost of r is $\frac{d(r,s_2)}{w}$, where $w \geq 1$ is the weight of sensor s_2. The goal is to minimize the maximum matching cost, we design an optimal online algorithm with a competitive ratio of $1 + w + \frac{1}{w}$ for $1 \leq w \leq 1.839$, and an optimal online algorithm with a competitive ratio of $\frac{w+1+\sqrt{w^2+6w+1}}{2}$ for $w > 1.839$.

Keywords: online algorithm; bottleneck matching; two heterogeneous sensors

Citation: Xiao, M.; Yang, Y.; Li, W. Online Bottleneck Matching Problem with Two Heterogeneous Sensors in a Metric Space. *Computation* **2022**, *10*, 217. https://doi.org/10.3390/computation10120217

Academic Editors: Akbar Ali, Guojun Li, Mingchu Li, Rao Li, Colton Magnant and Madhumangal Pal

Received: 30 September 2022
Accepted: 6 December 2022
Published: 9 December 2022

1. Introduction

In a metric space (X, d), X is a set of points and $d(\cdot, \cdot)$ is a distance function. $S = \{s_1, s_2, \ldots, s_n\} \subseteq$ is a set of sensors, and $R = \{r_1, r_2, \ldots, r_n\} \subseteq X$ is a set of requests. Each request arrives one by one in an online fashion. When a request $r_j \in R$ arrives, it must be immediately and irrevocably matched to some unmatched sensor s_i. The cost of matching r_j to s_i is the distance $d(r_j, s_i)$.

For a minimization problem and an input instance I, let $C^A(I)$ (C^A for short) and $C^{OPT}(I)$ (C^{OPT} for short) be the costs of the feasible solution obtained by an online algorithm A and an optimal off-line algorithm, respectively. An online algorithm A is *ρ-competitive* (or the *competitive ratio* of A is at most ρ) if $C^A \leq \rho C_{OPT}$ for any input instance I. For an online problem, if there is no algorithm with competitive ratio less than r, then r is a lower bound of the problem. If there is an algorithm whose competitive ratio matches the lower bound, this algorithm is called an optimal online algorithm.

Online problem and algorithm can be abstracted as a request-answer game between the algorithm designer and adversary. Each time the adversary gives a request, the algorithm decides how to respond based on the previous responses of both parties and the current request, and the adversary gives the next request based on the responses of both parties so far. The goal of the algorithm is to make it perform as well as possible, while the goal of the adversary is exactly the opposite.

Online-matching based models have many applications, e.g., when assigning cars to parking spots, advertisers to ad slots, skis to skiers at a rental station, drivers and riders in Uber. The classical Online Minimum Matching (OMM) is to find a matching M such that the total cost of matching all requests is minimized. Kalyanasundaram and Pruhs [1] and Khuller et al. [2] independently introduced the OMM problem and proved that the PERMUTATION is an optimal online algorithm with a competitive ratio of $2m - 1$. Meanwhile, Kalyanasundaram and Pruhs [1] also proved that the greedy algorithm is $(2^m - 1)$-competitive, and this bound is tight. Meyerson et al. [3] presented a randomized online algorithm with competitive ratio of $O(\log^3 m)$. Bansal et al. [4] proposed an $O(\log^2 m)$-competitive randomized algorithm.

If the metric space (X, d) is a line, Gupta and Lewi [5] gave an $O(\log n)$-competitive randomized algorithm for the OMM problem. Fuchs et al. [6] gave a lower bound 9.001 for the OMM problem. Antoniadis et al. [7] designed a deterministic online algorithm with competitive ratio $O(n^{\log(3+\epsilon)-1}/\epsilon)$ for any $\epsilon > 0$. Nayyar and Raghvendra [8] proved that the competitive ratio of the deterministic online algorithm proposed in [9] is $O(\log^2 n)$, which is improved to $O(\log n)$ [10] for the OMM problem. Recently, Peserico and Scquizzato [11] proved that the competitive ratio of any randomized online algorithm for the OMM problem exceeds $\sqrt{\log_2(n+1)}/15$.

The OMM problem is closely related to the Online Bottleneck Matching (OBM) problem which is to find a matching M such that the maximum cost is minimized. Kalyanasundaram and Pruhs [1] also introduced the OBM problem and proved that the PERMUTATION algorithm is $(2m-1)$-competitive, and gave a lower bound of $m+1$. Idury and Schaffer [12] gave a lower bound approximately $1.44m$ for the OBM problem even if the metric space is a real line. Anthony and Chung [13] proved that the greedy algorithm achieves a competitive ratio of $m2^{m-1}$.

A generalized version of the OMM problem, which is called online b-matching [1], online transportation [1], the fire station problem [14], the school assignment problem [14], or online facility assignment [15], is also considered, where each sensor can be matched multiple times. Recently, Itoh et al. [16] presented several lower bounds on the competitive ratio for this problem with different number of sensors. Xiao and Li [17] considered the semi-matching problem with two heterogeneous sensors, and proposed two optimal online algorithms.

In this paper, we describe our problems and some preliminaries in Section 2. In Section 3, we consider an OBM problem with two heterogeneous sensors. We gives two optimal online algorithms, and the competitive ratios are shown in Figure 1. Finally, we make a summary in Section 4.

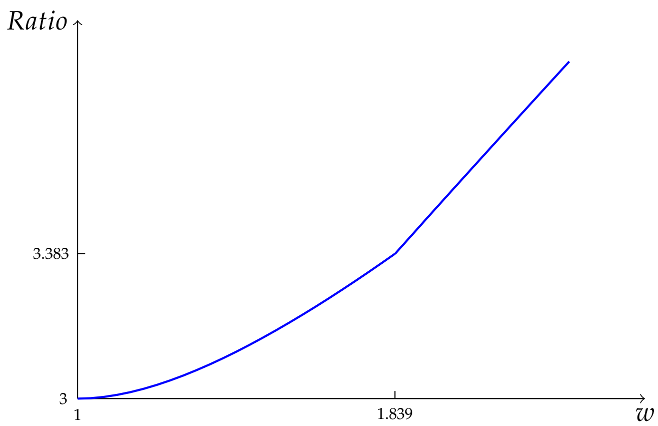

Figure 1. Two competitive ratios in this paper.

2. Preliminaries

Let (X, d) be a metric space, where X is a set of points and $d(\cdot,\cdot)$ is a distance function. We are given a set of two heterogeneous sensors $S = \{s_1, s_2\}$ and two requests r_1, r_2 in X, where each sensor s_i is characterized by the position $p(s_i) \in X$ and the weight $w_i > 0$, $i = 1, 2$. A request r_j is also characterized by the position $p(r_j) \in X$, $j = 1, 2$.

In the online fashion, the positions of the sensors are known in advance. The requests arrive one-by-one and the request r_1 must be assigned to some unmatched sensor before the request r_2 arrives. A sensor s_i is called **available** if the s_i is not matched by one request. If r_j is matched with the sensor s_i, the cost of pair (r_j, s_i) is

$$c(r_j, s_i) = d(r_j, s_i)/w_i$$

For convenience, let $\sigma^{-1}(i)$ be the request matched by sensor s_i for $i = 1, 2$. Without loss of generality, we assume that $w_1 = 1$ and $w_2 = w \geq 1$.

The online bottleneck matching problem with two heterogeneous sensors ($OBM(2)$) is to find a matching σ such that the maximum cost

$$\max\{\max_{j:r_j=\sigma^{-1}(1)} d(r_j, s_1), \max_{j:r_j=\sigma^{-1}(2)} d(r_j, s_2)/w\}$$

is minimized. Clearly, if $w = 1$, this problem is exactly the problem considered in [18] and has an optimal online algorithm with competitive ratio 3.

3. The OBM(2) Problem

In this section, for different ranges of w, we use the adversary approach to obtain the worst case competitive ratios of the OBM(2) problem. Let $\beta \approx 1.839$ be the real root of the equation $x^3 - x^2 - x - 1 = 0$, implying that

$$\beta^3 - \beta^2 - \beta - 1 = 0.$$

We propose two optimal online algorithms for $1 < w \le \beta$ and $w > \beta$ in next two subsections respectively.

3.1. The Case $1 \le w \le \beta$

Theorem 1. *When $1 \le w \le \beta$, any online algorithm A for OBM(2) has a competitive ratio at least $1 + w + \frac{1}{w}$, even if (X, d) is a line.*

Proof. As shown in Figure 2, let $p(s_1) = 0$, $p(s_2) = 1$, and $x = \frac{1}{1+w}$. The first request arrives at position $p(r_1) = p(s_2) - x$. If r_1 is matched with sensor s_2, the last request r_2 arrives at position $p(r_2) = p(s_2) + (1 - x)w$. Therefore, $C^A \ge d(r_2, s_1) = 1 + (1 - x)w$, and $C^{OPT} = d(r_1, s_1) = \frac{d(r_2,s_2)}{w} = (1 - x)$, implying that

$$\frac{C^A}{C^{OPT}} \ge \frac{1 + (1-x)w}{1-x} = \frac{1}{1-x} + w = 1 + w + \frac{1}{w}.$$

If r_1 is matched with sensor s_1, the last request r_2 arrives at position $p(r_2) = p(s_1) - \frac{x}{w}$. Therefore, $C^A \ge \frac{d(r_2,s_2)}{w} = \frac{1}{w} + \frac{x}{w^2}$, $C^{OPT} = \frac{d(r_1,s_2)}{w} = d(r_2, s_1) = \frac{x}{w}$, implying that

$$\frac{C^A}{C^{OPT}} \ge \frac{\frac{1}{w} + \frac{x}{w^2}}{\frac{x}{w}} = \frac{1}{x} + \frac{1}{w} = 1 + w + \frac{1}{w}.$$

According to the above analysis, there is no online algorithm which competitive ratio strict smaller than $1 + w + \frac{1}{w}$. □

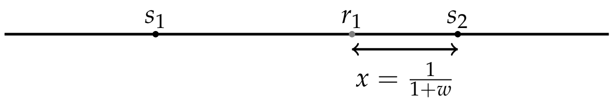

Figure 2. The locations of sensors and requests.

When $1 \le w \le \beta$, our Algorithm 1 is described as follows.

Algorithm 1: A1

1 When a new request r arrives,
2 **if** $d(r, s_1) \le wd(r, s_2)$, **then**
3 match r with the first available sensor in the sequence (s_1, s_2).
4 **else**
5 match r with the first available sensor in the sequence (s_2, s_1).
6 If there is a new request, go to **line 2.** Otherwise, stop.

Theorem 2. *The competitive ratio of Algorithm 1 is at most* $1 + w + \frac{1}{w}$.

Proof. Let σ^* be the off-line optimal solution, and σ be the feasible solution produced by the Algorithm 1. Let r_j be the request attaching the maximum in the σ, and r_k be the request attaching the maximum in the σ^*, $j, k \in \{1, 2\}$. For $j = 1, 2$, if $\sigma^*(r_j) = \sigma(r_j)$, implying that Algorithm 1 produces an optimal solution. Otherwise,

$$\sigma^*(r_j) \ne \sigma(r_j), \forall j = 1, 2, \tag{1}$$

we distinguish the following four cases.

Case 1. $\sigma(r_j) = s_1$ and $\sigma^*(r_k) = s_1$ ($\sigma(r_k) = s_2$ and $\sigma^*(r_j) = s_2$).

In this case, we have $C^{A1} = d(r_j, s_1)$ and $C^{OPT} = d(r_k, s_1)$. By (1), we have $j \ne k$. If $j < k$, by the choice of Algorithm 1, we have $d(r_j, s_1) \le wd(r_j, s_2)$. Since $w \le \beta$, we have

$$w^2 \le 1 + w + \frac{1}{w}, \tag{2}$$

implying that

$$C^{A1} = d(r_j, s_1) \le wd(r_j, s_2) = wd(r_j, \sigma^*(r_j)) \le w^2 C^{OPT} \le (1 + w + \frac{1}{w})C^{OPT}.$$

If $j > k$, by the choice of Algorithm 1, we have $d(r_k, s_1) > wd(r_k, s_2)$. Hence,

$$\begin{aligned}
C^{A1} &= d(r_j, s_1) \\
&\le d(r_j, s_2) + d(s_1, s_2) \\
&\le d(r_j, s_2) + d(r_k, s_1) + d(r_k, s_2) \\
&\le d(r_j, \sigma^*(r_j)) + (1 + \frac{1}{w})d(r_k, s_1) \\
&= d(r_j, \sigma^*(r_j)) + (1 + \frac{1}{w})d(r_k, \sigma^*(r_k)) \\
&\le wC^{OPT} + (1 + \frac{1}{w})C^{OPT} \\
&= (1 + w + \frac{1}{w})C^{OPT}.
\end{aligned}$$

Case 2. $\sigma(r_j) = s_1$ and $\sigma^*(r_k) = s_2$.

In this case, we have $C^{A1} = d(r_j, s_1)$ and $C^{OPT} = \frac{d(r_k, \sigma^*(r_k))}{w} = \frac{d(r_k, s_2)}{w}$. By (1) and $j, k = 1, 2$, we have $j = k$.

If $j = 1$, by the choice of Algorithm 1, we have $d(r_1, s_1) \le wd(r_1, s_2)$. Therefore, by (2),

$$C^{A1} = d(r_j, s_1) \le wd(r_j, s_2) = w^2 C^{OPT} \le (1 + w + \frac{1}{w})C^{OPT}.$$

If $j = 2$, by the choice of Algorithm 1, we have $d(r_1, s_1) > wd(r_1, s_2)$. Hence,

$$\begin{aligned} C^{A1} &= d(r_2, s_1) \\ &\leq d(r_2, s_2) + d(s_1, s_2) \\ &\leq d(r_2, s_2) + d(r_1, s_1) + d(r_1, s_2) \\ &\leq d(r_2, \sigma^*(r_2)) + (1 + \frac{1}{w})d(r_1, s_1) \\ &= d(r_2, \sigma^*(r_2)) + (1 + \frac{1}{w})d(r_1, \sigma^*(r_1)) \\ &\leq wC^{OPT} + (1 + \frac{1}{w})C^{OPT} \\ &= (1 + w + \frac{1}{w})C^{OPT}. \end{aligned}$$

Case 3. $\sigma(r_j) = s_2$ and $\sigma^*(r_k) = s_2$ ($\sigma(r_k) = s_1$ and $\sigma^*(r_j) = s_1$).

In this case, we have $C^{A1} = \frac{d(r_j, s_2)}{w}$ and $C^{OPT} = \frac{d(r_k, s_2)}{w} = \frac{d(r_k, \sigma^*(r_k))}{w}$. By (1), we have $j \neq k$. If $j < k$, by the choice of Algorithm 1, we have $d(r_j, s_1) > wd(r_j, s_2)$. Hence,

$$C^{A1} = \frac{d(r_j, s_2)}{w} < \frac{d(r_j, s_1)}{w^2} = \frac{d(r_j, \sigma^*(r_j))}{w^2} \leq \frac{1}{w^2}\frac{d(r_k, \sigma^*(r_k))}{w} \leq \frac{1}{w^2}C^{OPT} \leq C^{OPT}.$$

If $j > k$, by the choice of Algorithm 1, we have $d(r_k, s_1) \leq wd(r_k, s_2)$. Hence,

$$\begin{aligned} C^{A1} &= \frac{d(r_j, s_2)}{w} \\ &\leq \frac{1}{w}(d(r_j, s_1) + d(s_1, s_2)) \\ &\leq \frac{1}{w}(d(r_j, s_1) + d(r_k, s_1) + d(r_k, s_2)) \\ &\leq \frac{1}{w}(d(r_j, \sigma^*(r_j)) + (1 + w)d(r_k, s_2)) \\ &= \frac{1}{w}(d(r_j, \sigma^*(r_j)) + (1 + w)d(r_k, \sigma^*(r_k))) \\ &\leq \frac{1}{w}(C^{OPT} + (1 + w)wC^{OPT}) \\ &= (1 + w + \frac{1}{w})C^{OPT}. \end{aligned}$$

Case 4. $\sigma(r_j) = s_2$ and $\sigma^*(r_k) = s_1$.

In this case, we have $C^{A1} = \frac{d(r_j, s_2)}{w}$ and $C^{OPT} = d(r_k, s_1) = d(r_k, \sigma^*(r_k))$. By (1), we have $j = k$.

If $j = 1$, by the choice of Algorithm 1, we have $d(r_j, s_1) > wd(r_j, s_2)$. Therefore,

$$C^{A1} = \frac{d(r_j, s_2)}{w} < \frac{d(r_j, s_1)}{w^2} = \frac{d(r_j, \sigma^*(r_j))}{w^2} \leq \frac{1}{w^2}C^{OPT} \leq C^{OPT}.$$

If $j = 2$, by the choice of Algorithm 1, we have $d(r_1, s_1) \leq wd(r_1, s_2)$. Hence,

$$\begin{aligned} C^{A1} &= \frac{d(r_j, s_2)}{w} \\ &\leq \frac{1}{w}(d(r_j, s_1) + d(s_1, s_2)) \\ &\leq \frac{1}{w}(d(r_j, s_1) + d(r_1, s_1) + d(r_1, s_2)) \\ &\leq \frac{1}{w}(d(r_j, s_1) + (1 + w)d(r_1, s_2)) \\ &= \frac{1}{w}(d(r_j, \sigma^*(r_j)) + (1 + w)d(r_1, \sigma^*(r_1))) \\ &\leq \frac{1}{w}(C^{OPT} + (1 + w)wC^{OPT}) \\ &= (1 + w + \frac{1}{w})C^{OPT}. \end{aligned}$$

Therefore, the theorem holds. □

3.2. The Case $w > \beta$

Theorem 3. *When $w > \beta$, any online algorithm A for OBM(2) has a competitive ratio at least $\frac{w+1+\sqrt{w^2+6w+1}}{2}$, even if (X, d) is a line.*

Proof. Let $x = \frac{3w+1-\sqrt{w^2+6w+1}}{4w}$. As shown in Figure 3, assume that $p(s_1) = 0$ and $p(s_2) = 1$. The first request arrives at position $p(r_1) = p(s_2) - x$. If r_1 is matched with sensor s_2, the last request r_2 arrives at position $p(r_2) = p(s_2) + (1 - x)w$, implying $C^A \geq d(r_2, s_1) = 1 + (1 - x)w$ and $C^{OPT} = d(r_1, s_1) = \frac{d(r_2, s_2)}{w} = 1 - x$. Thus,

$$\frac{C^A}{C^{OPT}} \geq \frac{1 + (1 - x)w}{1 - x} = \frac{1}{1 - x} + w = \frac{w + 1 + \sqrt{w^2 + 6w + 1}}{2}.$$

If r_1 is matched with sensor s_1, the last request r_2 arrives at position $p(r_2) = p(s_1) - \frac{x}{w}$, implying that $C^A \geq d(r_1, s_1) = 1 - x$, and $C^{OPT} = \frac{d(r_1, s_2)}{w} = d(r_2, s_1) = \frac{x}{w}$. Thus,

$$\frac{C^A}{C^{OPT}} \geq \frac{1 - x}{\frac{x}{w}} = \frac{w + 1 + \sqrt{w^2 + 6w + 1}}{2}.$$

Therefore, the theorem holds. □

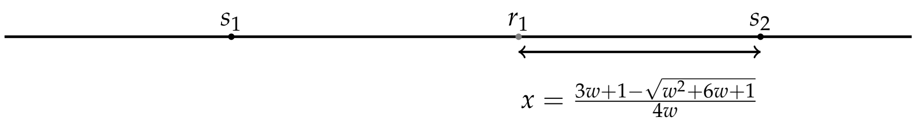

Figure 3. The locations of sensors and requests.

When $w > \beta$, for convenience, let

$$w_0 = \frac{w + 1 + \sqrt{w^2 + 6w + 1}}{2w}.$$

Our Algorithm 2 is described as follows.

Algorithm 2: A2

1 When a new request r arrives,
2 **if** $d(r, s_1) \leq w_0 d(r, s_2)$, **then**
3 match r with the first available sensor in the sequence (s_1, s_2).
4 **else**
5 match r with the first available sensor in the sequence (s_2, s_1).
6 If there is a new request, go to **line 2.** Otherwise, stop.

Theorem 4. *The competitive ratio of Algorithm 2 is at most* $\frac{w+1+\sqrt{w^2+6w+1}}{2}$.

Proof. As in the proof of Theorem 2, let r_j be the request attaching the maximum in the feasible solution σ produced by the Algorithm 2, and r_k be the request attaching the maximum in the optimal solution σ^*, $j, k = 1, 2$. We assume that

$$\sigma(r_j) \neq \sigma^*(r_j), \forall j = 1, 2, \tag{3}$$

and distinguish the following four cases.

Case 1. $\sigma(r_j) = s_1$ and $\sigma^*(r_k) = s_1$ ($\sigma(r_k) = s_2$ and $\sigma^*(r_j) = s_2$).

In this case, we have $C^{A2} = d(r_j, s_1)$ and $C^{OPT} = d(r_k, \sigma^*(r_k)) = d(r_k, s_1)$. By (3), we have $j \neq k$. If $j < k$, by the choice of Algorithm 2, we have $d(r_j, s_1) \leq w_0 d(r_j, s_2)$. Therefore,

$$C^{A2} = d(r_j, s_1) \leq w_0 d(r_j, s_2) = w_0 d(r_j, \sigma^*(r_j)) \leq w_0 w C^{OPT} = \frac{w+1+\sqrt{w^2+6w+1}}{2} C^{OPT}.$$

If $j > k$, by the choice of Algorithm 2, we have $d(r_k, s_1) > w_0 d(r_k, s_2)$. Hence,

$$\begin{aligned}
C^{A2} &= d(r_j, s_1) \\
&\leq d(r_j, s_2) + d(s_1, s_2) \\
&\leq d(r_j, s_2) + d(r_k, s_1) + d(r_k, s_2) \\
&\leq d(r_j, \sigma^*(r_j)) + (1 + \frac{1}{w_0}) d(r_k, s_1) \\
&= d(r_j, \sigma^*(r_j)) + (1 + \frac{1}{w_0}) d(r_k, \sigma^*(r_k)) \\
&\leq w C^{OPT} + (1 + \frac{1}{w_0}) C^{OPT} \\
&= (1 + w + \frac{1}{w_0}) C^{OPT} \\
&= \frac{w+1+\sqrt{w^2+6w+1}}{2} C^{OPT}.
\end{aligned}$$

Case 2. $\sigma(r_j) = s_1$ and $\sigma^*(r_k) = s_2$.

In this case, we have $C^{A2} = d(r_j, s_1)$ and $C^{OPT} = \frac{d(r_k, \sigma^*(r_k))}{w} = \frac{d(r_k, s_2)}{w}$. By (3) and $j, k = 1, 2$, we have $j = k$. If $j = 1$, by the choice of Algorithm 2, we have $d(r_j, s_1) \leq w_0 d(r_j, s_2)$. Therefore,

$$C^{A2} = d(r_j, s_1) \leq w_0 d(r_j, s_2) = w_0 w C^{OPT} = \frac{w+1+\sqrt{w^2+6w+1}}{2} C^{OPT}.$$

If $j = 2$, by the choice of Algorithm 2, we have $d(r_1, s_1) > w_0 d(r_1, s_2)$. Hence,

$$\begin{aligned} C^{A2} &= d(r_j, s_1) \\ &\leq d(r_j, s_2) + d(s_1, s_2) \\ &\leq d(r_j, s_2) + d(r_1, s_1) + d(r_1, s_2) \\ &\leq d(r_j, \sigma^*(r_j)) + (1 + \frac{1}{w_0}) d(r_1, s_1) \\ &= d(r_j, \sigma^*(r_j)) + (1 + \frac{1}{w_0}) d(r_1, \sigma^*(r_1)) \\ &\leq wC^{OPT} + (1 + \frac{1}{w_0}) C^{OPT} \\ &= (1 + w + \frac{1}{w_0}) C^{OPT} \\ &= \frac{w + 1 + \sqrt{w^2 + 6w + 1}}{2} C^{OPT}. \end{aligned}$$

Case 3. $\sigma(r_j) = s_2$ and $\sigma^*(r_k) = s_2$ ($\sigma(r_k) = s_1$ and $\sigma^*(r_j) = s_1$).

In this case, we have $C^{A2} = \frac{d(r_j, s_2)}{w}$ and $C^{OPT} = \frac{d(r_k, s_2)}{w} = \frac{d(r_k, \sigma^*(r_k))}{w}$. By (3), we have $j \neq k$. If $j < k$, by the choice of Algorithm 2, we have $d(r_j, s_1) > w_0 d(r_j, s_2)$. Hence,

$$C^{A2} = \frac{d(r_j, s_2)}{w} < \frac{d(r_j, s_1)}{ww_0} = \frac{d(r_j, \sigma^*(r_j))}{ww_0} \leq \frac{1}{ww_0} C^{OPT} \leq C^{OPT}.$$

If $j > k$, by the choice of Algorithm 2, we have $d(r_k, s_1) \leq w_0 d(r_k, s_2)$. Hence,

$$\begin{aligned} C^{A2} &= \frac{d(r_j, s_2)}{w} \\ &\leq \frac{1}{w}(d(r_j, s_1) + d(s_1, s_2)) \\ &\leq \frac{1}{w}(d(r_j, s_1) + d(r_k, s_1) + d(r_k, s_2)) \\ &\leq \frac{1}{w}(d(r_j, \sigma^*(r_j) + (1 + w_0) d(r_k, s_2))) \\ &= \frac{1}{w}(d(r_j, \sigma^*(r_j)) + (1 + w_0) d(r_k, \sigma^*(r_k))) \\ &\leq \frac{1}{w}(C^{OPT} + (1 + w_0) wC^{OPT}) \\ &= (1 + w_0 + \frac{1}{w}) C^{OPT} \\ &\leq \frac{w + 1 + \sqrt{w^2 + 6w + 1}}{2} C^{OPT}, \end{aligned}$$

where the last inequality follows from that $w > \beta$.

Case 4. $\sigma(r_j) = s_2$ and $\sigma^*(r_k) = s_1$.

In this case, we have $C^{A2} = \frac{d(r_j, s_2)}{w}$ and $C^{OPT} = d(r_k, s_1)$. By (3), we have $j = k$. If $j = 1$, by the choice of Algorithm 2, we have $d(r_j, s_1) > w_0 d(r_j, s_2)$. Therefore,

$$C^{A2} = \frac{d(r_j, s_2)}{w} < \frac{d(r_j, s_1)}{ww_0} = \frac{d(r_j, \sigma^*(r_j))}{ww_0} \leq \frac{1}{ww_0} C^{OPT} \leq C^{OPT}.$$

If $j = 2$, by the choice of Algorithm 2, we have $d(r_1, s_1) \leq w_0 d(r_1, s_2)$. Hence,

$$\begin{aligned} C^{A2} &= \frac{d(r_j, s_2)}{w} \\ &\leq \frac{1}{w}(d(r_j, s_1) + d(s_1, s_2)) \\ &\leq \frac{1}{w}(d(r_j, s_1) + d(r_1, s_1) + d(r_1, s_2)) \\ &\leq \frac{1}{w}(d(r_j, s_1) + (1 + w_0)d(r_1, s_2)) \\ &= \frac{1}{w}(d(r_j, \sigma^*(r_j)) + (1 + w_0)d(r_1, \sigma^*(r_1))) \\ &\leq \frac{1}{w}(C^{OPT} + (1 + w_0)wC^{OPT}) \\ &= (1 + w_0 + \frac{1}{w})C^{OPT} \\ &\leq \frac{w + 1 + \sqrt{w^2 + 6w + 1}}{2} C^{OPT}, \end{aligned}$$

where the last inequality follows from $w > \beta$.

Therefore, the theorem holds. □

4. Discussion

In this paper, we studied an online matching problem with two heterogeneous sensors in a metric space. For the bottleneck objective, we proposed two optimal online algorithms for different values of w. It is interesting to study the online matching problem with more than three sensors.

Author Contributions: M.X. proposed the methods, and wrote the manuscript; Y.Y. provided suggestions for the methods and reviewed the manuscript; W.L. proposed the problem, reviewed and modified the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: The work is supported in part by the National Natural Science Foundation of China [No. 12071417].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kalyanasundaram, B.; Pruhs, K. Online weighted matching. *J. Algorithms* **1993**, *14*, 478–488. [CrossRef]
2. Khuller, S.; Mitchell, S.G.; Vazirani, V.V. On-line algorithms for weighted bipartite matching and stable marriages. *Theor. Comput. Sci.* **1994**, *127*, 255–267. [CrossRef]
3. Meyerson, A.; Nanavati, A.; Poplawski, L. Randomized online algorithms for minimum metric bipartite matching. In Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm (SODA), Miami, FL, USA, 22–24 January 2006; pp. 954–959.
4. Bansal, N.; Buchbinder, N.; Gupta, A.; Naor, J.S. A randomized $O(\log^2 k)$-competitive algorithm for metric bipartite matching. *Algorithmica* **2014**, *68*, 390–403. [CrossRef]
5. Gupta, A.; Lewi, K. The online metric matching problem for doubling metrics. In Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP), Warwick, UK, 9–13 July 2012; pp. 424–435.
6. Fuchs, B.; Hochstattler, W.; Kern, W. Online matching on a line. *Theor. Comput. Sci.* **2005**, *332*, 251–264. [CrossRef]
7. Antoniadis, A.; Barcelo, N.; Nugent, M.; Pruhs, K.; Scquizzato, M. A $o(n)$-competitive deterministic algorithm for online matching on a line. *Algorithmica* **2019**, *81*, 2917–2933. [CrossRef]
8. Nayyar, K.; Raghvendra, S. An input sensitive online algorithm for the metric bipartite matching problem. In Proceedings of the IEEE 58th Annual Symposium on Foundations of Computer Science, Berkeley, CA, USA, 15–17 October 2017; pp. 505–515.
9. Raghvendra, S. A robust and optimal online algorithm for minimum metric bipartite matching. In Proceedings of the International Conference on Approximation, Randomization, and Combinatorial Optimization—Algorithms and Techniques (APPROX/RANDOM 2016), Seattle, WA, USA, 16–18 August 2016; p. 18.
10. Raghvendra, S. Optimal analysis of an online algorithm for the bipartite matching problem on a line. In Proceedings of the 34th International Symposium on Computational Geometry, College Park, MD, USA, 14–17 July 2017; p. 67.

11. Peserico, E.; Scquizzato, M. Matching on the line admits no $o(\sqrt{\log n})$-competitive algorithm. In Proceedings of the 48th International Colloquium on Automata, Languages, and Programming (ICALP), Glasgow, Scotland, 12–16 July 2021; p. 103.
12. Idury, R.; Schaffer, A.A. A Better Lower Bound for On-Line Bottleneck Matching. Manuscript. 1992. Available online: http://www.ncbi.nlm.nih.gov/core/assets/cbb/files/Firehouse.pdf (accessed on 29 September 2022).
13. Anthony, B.M.; Chung, C. Online bottleneck matching. *J. Comb. Optim.* **2014**, *27*, 100–114. [CrossRef]
14. Kalyanasundaram, B.; Pruhs, K. The online transportation problem. *SIAM J. Discret. Math.* **2000**, *13*, 370–383. [CrossRef]
15. Ahmed, A.R.; Rahman, M.S.; Kobourov, S. Online facility assignment. *Theor. Comput. Sci.* **2020**, *806*, 455–467. [CrossRef]
16. Itoh, T.; Miyazaki, S.; Satake, M. Competitive analysis for two variants of online metric matching problem. *Discret. Math. Algorithms Appl.* **2021**, *13*, 2150156. [CrossRef]
17. Xiao, M.; Li, W.D. Online semi-matching problem with two heterogeneous sensors in a metric space. In Proceedings of the 28th International Computing and Combinatorics Conference (COCOON), Shenzhen, China, 22–24 October 2022.
18. Xiao, M.; Zhao, S.; Li, W.D.; Yang, J.H. Online Bottleneck Semi-matching. In Proceedings of the 15th Annual International Conference on Combinatorial Optimization and Applications (COCOA), Tianjin, China, 17–19 December 2021; pp. 445–455.