



Online minimum matching with uniform metric and random arrivals

Sharmila V.S. Duppala^a, Karthik A. Sankararaman^b, Pan Xu^{c,*}

^a University of Maryland, College Park, USA

^b Meta AI, Menlo Park, USA

^c New Jersey Institute of Technology, Newark, USA

ARTICLE INFO

Article history:

Received 9 December 2021

Accepted 9 December 2021

Available online 14 December 2021

Keywords:

Online minimum matching

Uniform metric space

Random arrival order

ABSTRACT

We consider Online Minimum Bipartite Matching under the uniform metric. We show that Randomized Greedy achieves a competitive ratio equal to $(1 + 1/n)(H_{n+1} - 1)$, which matches the lower bound. Comparing with the fact that RG achieves an optimal ratio of $\Theta(\ln n)$ for the same problem but under the adversarial order, we find that the weaker arrival assumption of random order doesn't offer any extra algorithmic advantage for RG, or make the model strictly more tractable.

© 2021 Elsevier B.V. All rights reserved.

1. Introduction

In the past decade, online-matching based models have seen wide applications in rideshare (e.g., matching drivers and riders in Uber) and crowdsourcing markets (e.g., pairing workers and task in Amazon Mechanical Turk). In many of these real-world applications, we often need to find a matching between two disjoint sets such that the total distance over all matches is minimized. A motivating example can be seen in the online food ordering platform like Grubhub: we often need to find a matching between workers and online orders with the least total distance such that the total waiting time of all users is minimized (here we assume for simplicity that each worker can handle one order only).

Online Minimum Bipartite Matching (OMBM) was proposed by [5] and it has proved a powerful model in aforementioned applications. The basic setting is as follows. Let $[n] = \{1, 2, \dots, n\}$ for any generic integer n . Suppose we have two disjoint sets of points L and R in a metric space with metric $d(\cdot, \cdot)$ and each set consists of n points (i.e., $|L| = |R| = n$). Let us use $i \in L$ and $j \in R$ to index points in the two respective sets, and $d_{ij} \doteq d(i, j)$ for all $i \in L, j \in R$. Points in L are known in advance while points in R arrive sequentially in an online fashion: upon the arrival of each $j \in R$, we have to match it with a point $i \in L$, and it incurs a cost of d_{ij} . Note that each point in L can be matched only once. The goal is to design a matching algorithm such that the total cost is

minimized. Throughout this paper, we use the two terms “cost” and “distance” interchangeably.

Here are several variants in the metric definition and arrival setting. There are three common metrics studied in the literature, **line metric** where all points are required to be in a line; **uniform metric** where $d(\cdot, \cdot)$ takes only binary values, and **general metric** (no special requirements). As for the arrival setting, there are three well-studied assumptions, namely, adversarial order (AO), random order (RO), and known distributions (KD). For AO, both of the set R and the arrival order of R are fixed and unknown to the algorithm. For RO, the set of R is unknown but the arrival order of R is required to be a random permutation over R . For KD, R is known in advance, and during each time a single point from R will be sampled (or arrive) with replacement according to a certain distribution known as prior. A special case of KD is called IID, where arrival distributions over all rounds are known, identical and independent. In the following, we discuss a common measurement, called Competitive Ratio (CR), to evaluate the performance of online algorithms.

Competitive ratio (CR) Consider a given instance $\mathcal{I} = (L, R, d, \mathcal{A})$ of OMBM where d is the metric and $\mathcal{A}$ specifies the arrival setting. Consider an online minimization problem as studied here for example. Let $\text{ALG}(\mathcal{I}) = \mathbb{E}_{S \sim \mathcal{A}}[\text{ALG}(S)]$ denote the expected performance (i.e., the total cost) of ALG on the input $\mathcal{I}$, where the expectation is taken over the potential randomness in the arrival sequence S and that inherent in ALG. Let $\text{OPT}(\mathcal{I}) = \mathbb{E}_{S \sim \mathcal{A}}[\text{OPT}(S)]$ denote the expected *offline optimal* cost, where $\text{OPT}(S)$ refers to the minimum cost of a matching on the bipartite graph (L, S) after observing the full arrival sequence S from R . Then, the competitive ratio of ALG is defined as $\text{CR}(\text{ALG}) = \max_{\mathcal{I}} \frac{\text{ALG}(\mathcal{I})}{\text{OPT}(\mathcal{I})}$.

* Corresponding author at: 4310 Guttenberg Information Technologies Center, New Jersey Institute of Technology, Newark, USA.

E-mail addresses: sduppala@umd.edu (S.V.S. Duppala), karthikabinavs@gmail.com (K.A. Sankararaman), pxu@njit.edu (P. Xu).

Table 1

Summary of related works, where $\mathcal{D}$ and $\mathcal{R}$ denote the collections of all possible deterministic and randomized algorithms, respectively; DG and RG denote deterministic and randomized Greedy, respectively.

	Line metric	Uniform metric	General metric
Adversarial order	$\text{CR}(\text{DG}) = \Omega(2^n)$ [6]	$\text{LB}(\mathcal{D}) = \text{CR}(\text{DG}) = n$ [6]	$\text{LB}(\mathcal{D}) = 2n - 1$ [8,5,4]
	$\text{LB}(\mathcal{D}) = O(\ln n)$ [9]	$\text{LB}(\mathcal{R}) = \text{CR}(\text{RG}) = \Theta(\ln n)$ [6]	$\text{LB}(\mathcal{R}) = O(\ln^3 n)$ [6]
			$\text{LB}(\mathcal{R}) = \tau(n), \text{LB}(\mathcal{R}) = O(\ln^2 n)$ [1]
Random order	$\text{CR}(\text{DG}) = \Omega(n^{0.292})$ [2].	$\text{CR}(\text{RG}) \sim \ln n$ (This paper)	$\text{LB}(\mathcal{R}) \sim 2 \ln n$ [8].
	$\text{CR}(\text{DG}) = O(n)$ [2]	$\text{LB}(\mathcal{R}) \sim \ln n$ (This paper)	

Related work There is a long line of research on OMBM under various settings, see the summary of results in Table 1. Here are a few notations. Let $\mathcal{D}$ and $\mathcal{R}$ be the set of all deterministic and randomized algorithms. Among all algorithms in $\mathcal{D}$ and $\mathcal{R}$, two algorithms are studied most intensively: deterministic greedy (DG) and randomized greedy (RG). Generally, DG always try to assign an arrival point of $j \in R$ to an unmatched point $i \in L$ with the least cost d_{ij} . RG shares the same spirit with DG and the key difference between the two lies in the way of breaking ties when multiple optimal choices are available: DG breaks ties in an arbitrary fixed order while RG always samples an optimal choice uniformly. For either $\mathcal{D}$ and $\mathcal{R}$, let $\text{LB}(\mathcal{D}) = \inf_{\text{ALG} \in \mathcal{D}} \text{CR}(\text{ALG})$ and $\text{LB}(\mathcal{R}) = \inf_{\text{ALG} \in \mathcal{R}} \text{CR}(\text{ALG})$, which denote the respective competitive ratios achieved by an optimal deterministic and randomized algorithm. For two functions f and g over n , we write $f \sim g$ if $f/g \rightarrow 1$ when $n \rightarrow \infty$. In other words, f and g are asymptotically equal including the constants, which is a stronger notion than $\Theta(\cdot)$. Let $H_n = \sum_{k=1}^n 1/k \sim \ln n$.

Here are elaborations on the results shown in Table 1. We first focus on studies under the general metric. The work of [5] and [4] are among the first studies for AO: each independently gave an optimal deterministic $2n - 1$ -competitive algorithm. Recently, Raghvendra [8] presented a primal-dual-based deterministic algorithm that achieves almost the same ratio ($2n - 1 + o(1)$) for AO. What's more, they showed the same algorithm achieves an optimal competitive ratio of $2H_n - 1 - o(1)$ under RO. For AO, Meyerson et al. [6] gave the first greedy-based randomized algorithm, which has a sublinear competitive ratio of $O(\ln^3 n)$, which was then improved to $O(\ln^2 n)$ by [1] who also gave a lower bound of $O(\ln n)$. More recently, Gupta et al. [3] gave a $O((\ln \ln \ln n)^2)$ -competitive algorithm for KIID. Now we survey some results for the line metric. For AO, Raghvendra [9] showed that the deterministic Robust Matching algorithm achieves a ratio of $\Theta(\ln n)$. Pesarico and Squizzato [7] gave a lower bound of $\Omega(\sqrt{\ln n})$ for any randomized algorithm. Gairing and Klimm [2] studied the setting of RO and showed that the competitive ratio of DG is $O(n)$ and $\Omega(n^{0.292})$.

1.1. Main contributions

Our main contributions include a tight competitive analysis of the randomized greedy (RG) for OMBM under the uniform metric and RO and a proof of a lower bound suggesting the optimality of RG. We state our main theorems as follows. For a generic positive integer $n \in \mathbb{N}$, let $\tau(n) := (1 + 1/n)(H_{n+1} - 1)$ with H_{n+1} being the $(n + 1)$ th harmonic number. Observe that $\tau(n) = \Theta(\ln n)$.

Theorem 1.1. *Randomized Greedy (RG) achieves a competitive ratio equal to $\tau(n)$ for OMBM under the uniform metric and random order. Our analysis is tight.*

Theorem 1.2. *For OMBM under the uniform metric and random order, any algorithm (deterministic or randomized) will have a competitive ratio at least $\tau(n)$.*

Note that Meyerson et al. [6] analyzed both DG and RG on the uniform metric but under AO. The main results are: (1) DG achieves a competitive ratio of n , which is optimal among all deterministic online algorithms. (2) RG achieves a competitive ratio of $H_n \sim \Theta(\ln n)$, which is optimal among all randomized algorithms. By comparing the result in (2) with that stated in our main theorem, we see that for OMBM with the uniform metric, RG achieves the same competitive ratio in RO as that of AO. This suggests that when applying RG to OMBM with the uniform metric, the assumption of RO does not offer any extra algorithmic power compared with that of AO.

2. Proof of the Main Theorem 1.1

We split the whole proof into the below two lemmas. Throughout this section, we focus on OMBM under the uniform metric and RO. Let $\text{CR}(\text{RG})$ denote the competitive ratio achieved by the randomized greedy.

Lemma 2.1. $\text{CR}(\text{RG}) \geq \tau(n)$.

Lemma 2.2. $\text{CR}(\text{RG}) \leq \tau(n)$.

2.1. Proof of Lemma 2.1

Observe that though Theorem 1.2 implies Lemma 2.1, we present the proof for the completeness, and we believe the proof below offers insights into that of Theorem 1.2 and can serve as a good warmup.

Example 2.1. Consider the following instance. Recall that $|L| = |R| = n$ and $d_{ij} = d(i, j)$. Set $d_{ij} = 0$ iff $i = j \in [n - 1]$ and $d_{ij} = 1$ otherwise. In other words, the first $n - 1$ points in L colocate with the first $n - 1$ points in R , and there are essentially only $n + 1$ points in $L \cup R$.

From the nature of RG, we see that on the above example: (1) during each time when $j = n$ arrives, we will uniformly sample one available $i \in L$ since all costs are 1; (2) during each time when some $j < n$ arrives, we first check if $i = j$ is free. If so, then match it; otherwise uniformly sample one available $i \in L$. Observe that offline optimal is $\text{OPT} = 1$. Let $F(n)$ be the expected cost of RG on Example 2.1.

Lemma 2.3. $F(n) = \tau(n)$.

Proof. First, we can verify that $F(1) = 1$. When $n = 1$, we have essentially one point each in L and R with a distance 1. Thus, the expected cost of RG is 1. Now we try to devise a recursive formula on $F(n)$. Consider the case when $j = n$ arrives at some time $t \in [n - 1]$. Note that in our instance, $d_{i,n} = 1$ for all $i \in L$. We observe that (1) the current total cost for RG is 0 so far before matching $j = n$; (2) if we match $j = n$ to $i = n$, then the total final cost of RG will be 1; (3) if we match $j = n$ to some available

$i^* \in L$ with $i^* < n$, then the total cost should be $1 + F(n - t)$. In this scenario the remaining case can be reduced to the exact same problem with size $n - t$ where $j = i^*$ will play the role of $j = n$ before (we are sure $j = i^*$ never arrives before $j = n$ otherwise i^* will not be available). Thus, summarizing the above analysis, we have that

$$F(n) = 1 + \frac{1}{n} \sum_{t=1}^{n-1} \left(1 - \frac{1}{n-t+1}\right) F(n-t)$$

Together with the initial value $F(1) = 1$, we can solve that $F(n) = (1 + 1/n) \sum_{t=1}^n \frac{1}{t+1} = \tau(n)$. $\square$

Proof of Lemma 2.1

Proof. Note that on Example 2.1, the offline optimal $\text{OPT} = 1$ and RG has an expected total cost of $F(n)$. This suggests that the competitive ratio of RG on Example 2.1 is equal to $F(n)$. By definition of the competitive ratio, we claim $\text{CR}(\text{RG}) \geq F(n) = \tau(n)$. $\square$

2.2. Proof of Lemma 2.2

Consider a general instance of OMBM with $|L| = |R| = n$. Suppose the offline optimal is $\text{OPT} = n - k$ with $k \leq n$. Since the offline optimal is $n - k$, it suggests that there are at least k zero-cost pairs of points in $L \times R$. WLOG assume that $d_{ij} = 0$ for all $i = j \in [k]$ and all the rest $d_{ij} = 1$. Let $F(k, n)$ be the expected cost returned by RG. According to the nature of RG, we have the following two cases.

- With probability k/n , some $j \in [k]$ will arrive first. In this case, we will match j with $i = j$ with cost zero and the remaining case is reduced to $F(k-1, n-1)$.
- With probability $1 - k/n$, some $k+1 \leq j \leq n$ will arrive first. In this case, we will have a unit matching cost for j . If we match j with some $i \leq k$, then the resulting case is reduced to $F(k-1, n-1)$ (occurring with probability k/n); otherwise, the resulting case is reduced to $F(k, n-1)$ (occurring with probability $1 - k/n$).

Summarizing the above analysis, we have that

$$\begin{aligned} F(k, n) &= \frac{k}{n} F(k-1, n-1) \\ &\quad + \left(1 - \frac{k}{n}\right) \left(1 + \frac{k}{n} F(k-1, n-1)\right) \\ &\quad + \left(1 - \frac{k}{n}\right) \left(\left(1 - \frac{k}{n}\right) F(k, n-1)\right) \end{aligned}$$

Simplifying the above expression, we have $F(k, n) = (1 - k/n) + A(k, n) + B(k, n)$, where

$$A(k, n) = \left(1 - \frac{k}{n}\right)^2 F(k, n-1) \quad (1)$$

$$B(k, n) = \frac{k}{n} \left(2 - \frac{k}{n}\right) F(k-1, n-1) \quad (2)$$

Lemma 2.4.

$$F(k, n) \leq (n-k) \left(1 + \frac{1}{n}\right) (H_{n+1} - 1) \quad (3)$$

Note that Lemma 2.4 immediately implies Lemma 2.2 since the offline optimal is equal to $n - k$ and the instance is arbitrarily selected. We prove Lemma 2.4 by induction on n .

Proof. Consider the basic case $n = 1$. We can verify that $F(0, 1) = 1$ and $F(1, 1) = 0$, which satisfies Inequality (3). Assume the inequality is valid for all $k \leq n \leq N$ for some integer N . Now we show the case of $N + 1$. Consider any given $k \leq n \leq N + 1$. If $k = n$, we have that $F(k, n) = 0$, then we are done. Now assume $k < n \leq N + 1$. Recall that $F(k, n) = (1 - k/n) + A(k, n) + B(k, n)$. By inductive assumptions, we have

$$A(k, n) \leq \left(1 - \frac{k}{n}\right)^2 (n-k-1) \left(1 + \frac{1}{n-1}\right) (H_n - 1)$$

$$B(k, n) \leq \frac{k}{n} \left(2 - \frac{k}{n}\right) (n-k) \left(1 + \frac{1}{n-1}\right) (H_n - 1)$$

Plugging the above two inequalities to $F(k, n) = (1 - k/n) + A(k, n) + B(k, n)$, we have

$$F(k, n) \leq \left(1 - \frac{k}{n}\right) + \frac{(n-k)(n^2 - n + k)}{n(n-1)} (H_n - 1) \doteq G(k, n)$$

Notice that

$$\begin{aligned} G(k, n) &\leq (n-k) \left(1 + \frac{1}{n}\right) (H_{n+1} - 1) \\ &\Leftrightarrow \left(1 - \frac{k}{n}\right) + \frac{(n-k)(n^2 - n + k)}{n(n-1)} (H_n - 1) \\ &\leq (n-k) \left(1 + \frac{1}{n}\right) (H_{n+1} - 1) \\ &\Leftrightarrow \frac{1}{n} + \frac{(n^2 - n + k)}{n(n-1)} (H_n - 1) \leq \left(1 + \frac{1}{n}\right) (H_{n+1} - 1) \\ &\Leftrightarrow \frac{1}{n} + \frac{(n^2 - n + k)}{n(n-1)} (H_n - 1) \leq \left(1 + \frac{1}{n}\right) (H_n - 1) + \frac{1}{n} \\ &\Leftrightarrow \frac{H_n - 1}{n(n-1)} (n^2 - n + k - (n-1)(n+1)) \leq 0 \\ &\Leftrightarrow -(n-k-1) \leq 0 \end{aligned}$$

Now, we have proved that for any $k \leq n \leq N + 1$,

$$F(k, n) \leq G(k, n) \leq (n-k) \left(1 + \frac{1}{n}\right) (H_{n+1} - 1). \quad \square$$

3. Proof of the Main Theorem 1.2

In this section, we describe an approach to show that any randomized algorithm has a competitive ratio of at least $\tau(n) = (1 + 1/n) \sum_{t=1}^n 1/(t+1)$ for OMBM under the random order arrival. The high-level idea is as follows. We construct a family of instances $\mathcal{F}$ and a probability distribution p over these instances. We show that any deterministic online algorithm D has an expected cost of $\tau(n)$ for an instance f chosen randomly from $\mathcal{F}$ using p and then choosing a random arrival order σ of the vertices on the randomly chosen instance f . In other words $\mathbb{E}_{f \sim p} [\mathbb{E}_{\sigma} [D(f, \sigma)]] \geq \tau(n)$. Formally, we have the lemma below.

Lemma 3.1. *There exists a family of instances $\mathcal{F}$ of OMBM and a distribution p such that any deterministic online algorithm D has an expected cost of $\tau(n)$, where the expectation is taken over both the distribution of p and the random arrival order of vertices.*

We first show how the above lemma implies Theorem 1.2.

Proof. Assuming that $(\mathcal{F}, p)$ satisfies the property stated in Lemma 3.1 for any deterministic algorithm. First, note that for any given n , the number of deterministic algorithms is a finite number. Indeed, there are at most n rounds, and in each round the deterministic algorithm can choose to try and match an arriving vertex

to one of the U vertices by processing them in some fixed order for this round. Thus, for each arrival there are $n!$ different number of deterministic possible sequences in which the vertices in U can be processed. In other words, a deterministic algorithm is a map from the index of the round and the vertex in V to the sequence with which the vertices in U will be processed. This implies that the total number of deterministic algorithms is upper-bounded by the quantity $K := n^2 \cdot n!$. Let $\mathcal{D} = \{D_1, D_2, \dots, D_K\}$ be the collection of all deterministic algorithms. Let $D_k(f, \sigma)$ denote the cost incurred when running D_k on a given instance $f \in \mathcal{F}$ and a given arrival order σ of the RHS vertices of f . By the property in Lemma 3.1, we see that

$$\mathbb{E}_{f \sim p, \mathcal{F}, \sigma} [D_k(f, \sigma)] \geq \tau(n), \forall D_k \in \mathcal{D}.$$

This suggests that for any vector $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_K) \in [0, 1]^K$ with $\sum_{k=1}^K \alpha_k = 1$,

$$\mathbb{E}_{f \sim p, \mathcal{F}, \sigma} \left[\sum_{k=1}^K \alpha_k \cdot D_k(f, \sigma) \right] \geq \tau(n). \quad (4)$$

Note that for any online randomized algorithm R , there exists a unique vector $\alpha^R = [0, 1]^K$ with $\sum_{k=1}^K \alpha_k^R = 1$ such that R can be viewed as running the deterministic algorithm D_k with probability α_k for $k = 1, 2, \dots, K$. Thus, Inequality (4) implies that for any randomized algorithm R ,

$$\mathbb{E}_{f \sim p, \mathcal{F}, \sigma} [\mathbb{E}_R[R(f, \sigma)]] \geq \tau(n).$$

Therefore, we claim the existence of an instance $\hat{f}$ such that $\mathbb{E}_{R, \sigma} [R(\hat{f}, \sigma)] \geq \tau(n)$ for all R . $\square$

3.1. Proof of Lemma 3.1

Now we show how to construct the family $\mathcal{F}$ and the distribution p such that any deterministic algorithm incurs a regret of at least $\tau(n)$.

We have a complete bipartite graph (U, V, E) with $U = \{u_1, u_2, \dots, u_n\}$ and $V = \{v_1, v_2, \dots, v_n\}$. Let instance $\mathcal{I}_\pi$ denote the following instance. Choose a random permutation π over U . For all edges of the form $(\pi(u_i), v_i)$ with $i \in [n-1]$ we add an edge of cost 0. All other edges have a cost 1. The distribution p is the uniform distribution.

We will now make an observation which will simplify the description of the proof. We have that for every instance in the family $\mathcal{F}$ and arrival sequence, any deterministic algorithm $\tilde{D}$ that ignores a 0-cost neighbor, when available, is dominated by another deterministic algorithm D that chooses the 0-cost neighbor at that step. This implies that it suffices to consider only those deterministic algorithms D which chooses the 0-cost neighbor when available and uses a fixed ordering of vertices in U to break the ties among the 1-cost edges.

Now we will prove that any deterministic algorithm incurs an expected cost of at least $\tau(n)$. Consider a deterministic algorithm D . This algorithm can be viewed as a $n \times n$ matrix M_D where the i^{th} row represents the order in which the algorithm chooses a neighbor for vertex v_i . Note from the observation above we consider only those matrices where in the first $n-1$ rows we have that the first element is the 0-cost neighbor. Consider the sequence of arrivals σ . Since the matrix is fixed, on a random input from this family (f, σ) the action of this algorithm is equivalent to the following randomized algorithm on the instance where the 0-cost edges are between the pairs $(u_1, v_1), (u_2, v_2), \dots, (u_{n-1}, v_{n-1})$. The algorithm chooses a permutation π over the labels of the vertices in U . For each vertex v_i , the randomized algorithm checks

the vertices in π according to the order $M_D(i)$. We will now prove via induction that the expected cost incurred by this algorithm is given by the recurrence,

$$F(n) = 1 + \frac{1}{n} \sum_{t=1}^{n-1} \left(1 - \frac{1}{n-t+1} \right) F(n-t),$$

with a base case of $F(1) = 1$.

This evaluates to the expression $F(n) = \tau(n)$ and therefore shows that D incurs an expected cost of at least $\tau(n)$.

We prove the following inductive hypothesis, which will complete the proof.

Inductive Hypothesis. For every graph with $n \geq 2$ and every probing order $\psi_1, \psi_2, \dots, \psi_n$ over the random permutation for vertices $v_1, v_2, \dots, v_n$ respectively, the performance is given by the recurrence above.

The base case is when $n = 2$. In this case note that both v_1 and v_2 have a unique probing order over the 1-cost edges. We can verify that the performance is given by the above recurrence in this case.

Now suppose for a given k , we have proved the inductive hypothesis for all $2 \leq n \leq k-1$ and all probing sequences $\psi_1, \psi_2, \dots, \psi_n$. Consider the case when $n = k$ and an arbitrary probing sequences $\psi_1, \psi_2, \dots, \psi_n$. Consider some $1 \leq t \leq n-1$ when the vertex v_n comes. In the randomized algorithm, the probability that v_n chooses a the “blocking” neighbor is given by $1 - \frac{1}{n-t+1}$. Also note that only the $n-t$ vertices in V have their relative ordering in π fixed. Therefore the reduced instance is now on the $n-t$ vertices with the relative ordering of *none* of the vertices in π fixed. Since we have proved by induction that for a graph of size $2, 3, \dots, n-1$ and any arbitrary order of probing $\psi_1, \psi_2, \dots, \psi_{n-1}$, the recurrence holds, therefore the reduced instance has a performance $F(n-t)$.

Note that this above proof implies that after choosing a permutation π , we can wlog assume that every vertex v_i chooses the tie-breaking in the same order according to π , for this instance.

4. Conclusion

We study the randomized greedy for OMBM problem under the uniform metric and RO. In particular, we give an exact formula for the competitive ratio achieved by RG and prove its optimality. We find that when applying RG to OMBM with the uniform metric, the assumption of RO does not offer any extra algorithmic power compared with that of AO.

Acknowledgement

The third author Pan Xu is partially supported by NSF CRII Award IIS-1948157.

References

- [1] N. Bansal, N. Buchbinder, A. Gupta, J. Naor, A randomized $O(\log^2 k)$ -competitive algorithm for metric bipartite matching, *Algorithmica* 68 (2014) 390–403, <http://dblp.uni-trier.de/db/journals/algorithmica/algorithmica68.html#BansalBGN14>.
- [2] M. Gairing, M. Klimm, Greedy metric minimum online matchings with random arrivals, *Oper. Res. Lett.* 47 (2019) 88–91.
- [3] A. Gupta, G. Guruganesh, B. Peng, D. Wajc, Stochastic online metric matching, *arXiv preprint, arXiv:1904.09284*, 2019.
- [4] B. Kalyanasundaram, K. Pruhs, Online weighted matching, *J. Algorithms* 14 (1993) 478–488.
- [5] S. Khuller, S.G. Mitchell, V.V. Vazirani, On-line algorithms for weighted matching and stable marriages, Technical Report, Cornell University, 1990.
- [6] A. Meyerson, A. Nanavati, L. Poplawski, Randomized online algorithms for minimum metric bipartite matching, in: *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm, Society for Industrial and Applied Mathematics*, 2006, pp. 954–959.

- [7] E. Peserico, M. Scquizzato, Matching on the line admits no $o(\sqrt{\log n})$ -competitive algorithm, in: 48th International Colloquium on Automata, Languages, and Programming (ICALP 2021), Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.
- [8] S. Raghvendra, A robust and optimal online algorithm for minimum metric bipartite matching, in: Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2016), Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.
- [9] S. Raghvendra, Optimal analysis of an online algorithm for the bipartite matching problem on a line, arXiv preprint, arXiv:1803.07206, 2018.