Operations Research Letters 47 (2019) 88–91

Contents lists available at ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

Greedy metric minimum online matchings with random arrivals

Martin Gairing [a], Max Klimm [b,*]

[a] University of Liverpool, Department of Computer Science, Ashton Building, Ashton Street, Liverpool, L69 3BX, UK
[b] Humboldt-Universität zu Berlin, Institute of Operations Research, School of Business and Economics, Spandauer Str. 1, 10099 Berlin, Germany

ARTICLE INFO

Article history:
Received 30 September 2018
Received in revised form 31 December 2018
Accepted 6 January 2019
Available online 14 January 2019

Keywords:
Random arrival
Matching
Greedy algorithm
Approximation

ABSTRACT

We consider online metric minimum bipartite matching problems with random arrival order and show that the greedy algorithm assigning each request to the nearest unmatched server is n-competitive, where n is the number of requests. This result is complemented by a lower bound exhibiting that the greedy algorithm has a competitive ratio of at least $n^{(\ln 3-\ln 2)/\ln 4} \approx n^{0.292}$, even when the underlying metric space is the real line.

1. Introduction

In the metric minimum online matching problem we are given a metric space (X, d) containing a set $S = \{s_1, \ldots, s_n\} \subseteq X$ of n servers. A set of requests $R = \{r_1, \ldots, r_n\} \subseteq X$ arrives online in a one-by-one fashion. When a request r_i arrives it has to be matched immediately and irrevocably to a previously unmatched server $s(r_i)$. Matching request r_i to server $s(r_i)$ incurs a cost of $d(r_i, s(r_i))$, and we are interested in minimizing the total cost $\sum_{i=1}^{n} d(r_i, s(r_i))$. A notable special case is the online matching problem on a line where $X = \mathbb{R}$ and $d(x, y) = |x - y|$.

Online matching problems of this kind have many applications, e.g., when assigning cars to parking spots, advertisers to ad slots, or skis to skiers at a rental station. As a consequence, matching problems were among the first for which online algorithms with provable performance guarantee were studied. An important concept for measuring the quality of an algorithm is the competitive ratio defined as the worst-case ratio of the cost of the algorithm and the cost of an optimal algorithm that knows all requests beforehand. When the requests arrive in an adversarial order, there is a deterministic algorithm with competitive ratio $2n - 1$ which is best possible [8,12]. When the requests arrive in a random order, there is a deterministic algorithm with competitive ratio $(2H_n - 1)$ where H_n is the n'th harmonic number. This competitive ratio is best possible for this model [17].

The optimal online algorithms in [8,12,17] have the property that they sometimes match requests to a distant server even though a closer server is still available. This is problematic for several reasons. When the requests come from selfish players that are interested in minimizing the cost of connecting to their matched server, the players may simply choose the best unmatched server (e.g., as in the parking scenario), be dissatisfied when not being offered the best fit (e.g., in the ski rental station), or may strategically misreport their request in order to being matched to a better server.

On the other hand, matching all requests greedily to the nearest server avoids these issues, but leads to an exponential competitive ratio [8,12]. The exponential lower bound on the competitive ratio of the greedy algorithm, however, relies on the requests to arrive in an adversarial order. It has been argued that in many scenarios it makes sense to assume that the requests arrive in a random order [6] and for the optimal deterministic algorithm there is indeed an exponential gap between the optimal competitive ratios for the adversarial and random order setting [8,12,17].

1.1. Our results

We show that for any metric space the natural greedy algorithm that assigns each request to the nearest unmatched server is n-competitive in the random arrival model. This is in contrast to a lower bound of $2^n - 1$ on the competitiveness of the greedy algorithm for adversarial orders and shows that random arrival decreases the competitiveness of the greedy algorithm by an exponential factor. We complement our result with a lower bound establishing that the competitive ratio of the greedy algorithm is at least $n^{(\ln 3-\ln 2)/\ln 4} \approx n^{0.292}$, even on the line.

* Corresponding author.
E-mail addresses: M.Gairing@liverpool.ac.uk (M. Gairing), max.klimm@hu-berlin.de (M. Klimm).

https://doi.org/10.1016/j.orl.2019.01.002
0167-6377/

1.2. Related work

Karp et al. [11] studied a maximum online bipartite matching problem where the vertices on the right hand side are known in advance while the vertices on the left hand side arrive in an online fashion. Upon arrival of a vertex, the incident edges are revealed and a previously unmatched vertex of the right hand side has to be assigned immediately and irrevocably. The goal is to maximize the number of matched vertices. In this setting any algorithm that constructs a maximal matching is clearly 1/2-competitive and Karp et al. showed that no deterministic algorithm can beat this bound. They also proposed a randomized algorithm – termed ranking algorithm – that fixes a random permutation of the vertices on the right hand side beforehand and matches each vertex of the left hand side after arrival to the highest ranked vertex on the right hand side that is unmatched. They showed that this algorithm is $(1-1/e)$-competitive which is best possible. To show this result, Karp et al. proved that the performance of the ranking algorithm for adversarial input is equal to the performance of the greedy algorithm in the random permutation model where in the latter the order of arrival of the vertices of the left hand side is chosen uniformly at random.

Khuller et al. [12] and Kalyanasundaram and Pruhs [8] considered an online bipartite matching problem with the goal to minimize the cost of the matching. They showed that for non-metric spaces the competitive ratio of any algorithm is unbounded. For metric spaces, they gave an online algorithm with competitive ratio of $2n-1$ which is best possible for deterministic algorithms. Kalyanasundaram and Pruhs [8] also showed that the natural greedy algorithm that matches each vertex to the closest unmatched vertex is not better than (2^n-1)-competitive, even on the line. A recent algorithm by Raghvendra [17] matches the best possible competitive ratio of $2n-1$ when the requests arrive in an adversarial order. Interestingly, if the order is chosen randomly, than the competitive ratio of this algorithm reduces to $\mathcal{O}(\log n)$, which they also show to be best possible.

For metric spaces also *randomized* algorithms have been considered. In this regard, Meyerson et al. [15] and, independently, Csaba and Pluhár [4] obtained a randomized $\mathcal{O}(\log^3 n)$-competitive algorithm using randomized tree embeddings. Their approach was subsequently improved by Bansal et al. [3] to obtain a randomized $\mathcal{O}(\log^2 n)$-competitive algorithm. They also showed that there is no randomized algorithm with a competitive ratio better than $\ln n$.

An online problem related to the online matching problem on the line is the so-called cow-path problem. In the cow path problem a single server is located at point zero and needs to find an unknown point on the line while minimizing the distance travelled. For this problem, the optimal competitive ratio achievable by deterministic algorithms is 9 and is achieved by a standard doubling technique, see Baeza-Yates et al. [2]. For randomized algorithms the tight competitive ratio is about 4.591, see Kao et al. [10]. Kalyanasundaram and Pruhs [9] observed that the cow-path problem is a special case of the online matching problem on the line and conjectured that there is also a 9-competitive algorithm for the latter problem. This conjecture was refuted by Fuchs et al. [5] who gave a lower bound of 9.001 on the competitive ratio of any deterministic algorithm for the online matching problem on the line. Kalyanasundaram and Pruhs further conjectured that the work function algorithm of Koutsoupias and Papadimitriou [14] is constant competitive for online matching on the line, but Koutsoupias and Nanavati [13] showed that it has competitive ratio $\Omega(\log n)$. Gupta and Lewi [7] gave a randomized algorithm for online matching on the line with competitive ratio $\mathcal{O}(\log n)$. For deterministic algorithms, Antoniadis et al. [1] gave an algorithm with sublinear competitive ratio. Nayyar and Raghvendra [16] gave an algorithm with competitive ratio $\mathcal{O}(\log^2 n)$. Raghvendra [18] provided a tight analysis of the letter algorithm showing that it has a competitive ratio of $\Theta(\log n)$.

2. Upper bound on the performance of the greedy algorithm

In the random order model, the requests are revealed according to a permutation $\pi[n] \to [n]$ chosen uniformly at random among all permutations of n elements, i.e., the requests are revealed in order $r_{\pi(1)}, \dots, r_{\pi(n)}$. We are interested particularly in the performance of the natural greedy algorithm that assigns each request to the nearest available server, i.e., $s(r_i) \in \arg\min_{s \in S \setminus \bigcup_{j=1}^{i-1} s(r_j)} d(r_i, s)$.

We first show that the greedy algorithm has a competitive ratio of at most n, where n is the number of requests.

Theorem 1. *In the random order model the greedy algorithm has a competitive ratio of at most n.*

Proof. Let c_i be the expected cost of the i'th request assigned by the greedy algorithm, where the expectation is taken over all permutations of requests, and let c_i^* denote the expected cost of the same request (arriving on position i in the random order) in an optimum solution. As each request appears in each position of the random order with probability $1/n$, we have $c_i^* = \mathsf{OPT}/n$ for all requests positions $i \in [n]$.

We claim that

$$c_j \le c_j^* + \frac{1}{n-j+1}\sum_{i=1}^{j-1} c_i + \frac{1}{n-j+1}\sum_{i=1}^{j-1} c_i^*$$

for all $j \in [n]$.

For the proof of the claim, let $j \in [n]$ be arbitrary and let us fix the assignment of requests to servers up to request r_{j-1}. For $i \in [j-1]$, let $s(r_i)$ denote the server assigned to request r_i. Moreover, for $i \in [n]$ let $s^*(r_i)$ denote the server assigned to request r_i in a fixed optimal assignment OPT.

Consider the auxiliary directed graph G in which the set of vertices corresponds to requests, and there is a directed edge (r_i, r_j) if $s(r_i) = s^*(r_j)$, i.e., request r_i is assigned to the server that is assigned to r_j in OPT. (We also allow for self-loops when requests are matched to the same server in the greedy and optimal solution.) When request r_j arrives, G has exactly $j-1$ edges, each vertex is the endpoint of at most one edge, and there is no edge leaving the vertex that corresponds to request r_j. Conditioned on the event that r_j is not the endpoint of an edge, we have $c_j \le c_j^*$ as $s^*(r_j)$ has not been assigned to a request $r_i, i \in \{1, \dots, j-1\}$, and, thus, the cost of matching r_j is bounded by the cost in OPT. If, on the other hand, r_j is the endpoint of an edge, let $P = (i_1, \dots, i_l = r_j)$ with $l \le j-1$ be the maximal directed path in G that ends in r_j. Then, by the maximality of P, the server $s^*(r_{i_1})$ is available and, thus, the expected cost of the j'th request can be bounded from above by

$$c_j \le c_j^* + \sum_{k=1}^{\ell} (c_{i_k} + c_{i_k}^*), \tag{1}$$

by the triangle inequality, see also Fig. 1.

Finally, note that for a request $r_i, i \in [j-1]$, the term $(c_i + c_i^*)$ appears in the upper bound on c_j in (1) if and only if request r_j is the endpoint of the unique path that uses the edge originating in r_i. With other words, among the remaining $n-j+1$ requests, there is exactly one for which the term $(c_i + c_i^*)$ appears in the calculation of c. As each of the remaining $n-j+1$ requests is selected with probability $\frac{1}{n-j+1}$, we obtain

$$c_j \le c_j^* + \frac{1}{n-j+1}\sum_{i=1}^{j-1} (c_i + c_i^*), \tag{2}$$

as claimed. To finish the proof, note that (2) implies

$$\sum_{i=1}^{j} c_i \le c_j^* + \left(1 + \frac{1}{n-j+1}\right)\sum_{i=1}^{j-1} c_i + \frac{1}{n-j+1}\sum_{i=1}^{j-1} c_i^*$$

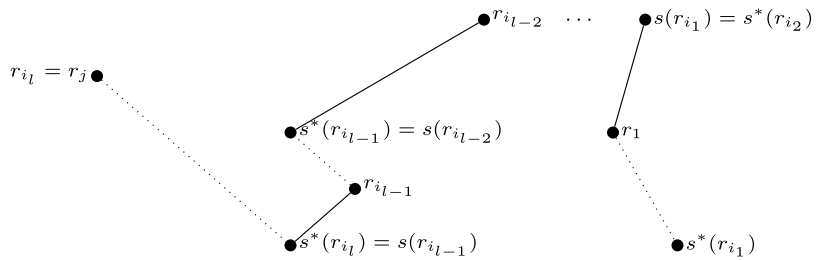

Fig. 1. Servers available to request r_j in the proof of Theorem 1. Solid lines correspond to costs that occur in the greedy solution while dotted lines correspond to costs that occur in the optimal solution. As costs are metric, the cost $d(r_j, s^*(r_{i_1}))$ is bounded by the length of the path.

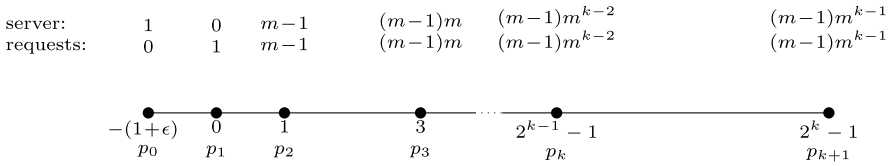

Fig. 2. Lower bound.

Using $c_i^* = \mathsf{OPT}/n$, we obtain

$$\sum_{i=1}^{j} c_i \le \frac{n-j+2}{n-j+1}\sum_{i=1}^{j-1} c_i + \left(\frac{j-1}{n-j+1}+1\right)\frac{\mathsf{OPT}}{n} = \frac{n-j+2}{n-j+1}\sum_{i=1}^{j-1} c_i + \frac{\mathsf{OPT}}{n-j+1}.$$

Recursively, we obtain

$$\sum_{i=1}^{j} c_i \le \frac{n}{n-j+1}c_1 + \frac{j-1}{n-j+1}\mathsf{OPT} = \frac{1}{n-j+1}\mathsf{OPT} + \frac{j-1}{n-j+1}\mathsf{OPT} = \frac{j}{n-j+1}\mathsf{OPT}.$$

For $j = n$ this implies the claimed result. □

3. Lower bound on the performance of the greedy algorithm

In this section, we show that the competitive ratio of the greedy algorithm is at least $n^{0.292}$, where n is the number of requests.

Theorem 2. *In the random order model the greedy algorithm has a competitive ratio of at least* $n^{\frac{\ln 3-\ln 2}{\ln 4}} \approx n^{0.292}$*, even on the line.*

Proof. Let $\epsilon > 0$ and consider the line with $k+2$ points $p_0, p_1, \ldots, p_{k+1}$ where

$$p_i = \begin{cases} -(1+\epsilon) & \text{if } i = 0, \\ 2^{i-1}-1 & \text{else}. \end{cases}$$

We then have $d(p_0, p_1) = 1+\epsilon$ and $d(p_i, p_{i+1}) = 2^{i-1}$ for all $i > 0$. We denote by δ_i the number of requests at point p_i and by σ_i the number of servers at point p_i. For an integer parameter m (which will be optimized to $m = 4$ later), let

$$\sigma_i = \begin{cases} 1 & \text{if } i = 0, \\ 0 & \text{if } i = 1, \\ (m-1)m^{i-2} & \text{else}, \end{cases} \qquad \delta_i = \begin{cases} 0 & \text{if } i = 0, \\ 1 & \text{if } i = 1, \\ (m-1)m^{i-2} & \text{else}. \end{cases}$$

See also Fig. 2 for an illustration.

First observe that the optimum allocation assigns the request at p_1 to the server at p_0, and all other requests to a server at the same point which yields a total cost of $\mathsf{OPT} = 1+\varepsilon$. Assume that the requests arrive in a random order Π drawn uniformly from all permutations. For $i \in \{1, \ldots, k\}$, let Π_i denote the order Π restricted to the requests from points $p_0, p_1, \ldots, p_{i+1}$. Let E_i be the event that the last request in Π_i is for point p_{i+1}. Let $q_i = \mathbf{Pr}(E_i)$ be the probability that E_i happens. Observe that for all $i \in [1, k]$,

$$q_i = \frac{\delta_{i+1}}{\sum_{j=1}^{i+1}\delta_j} = \frac{m-1}{m}.$$

Denote E the event that all events $(E_i)_{i\in[1,k]}$ happen simultaneously. Observe that for any pair $i, j \in [1, k]$, $i \neq j$, the events E_i and E_j are independent. So the probability that E happens is

$$q := \mathbf{Pr}(E) = q_1 \cdot q_2 \cdot \ldots \cdot q_k = \left(\frac{m-1}{m}\right)^k.$$

A simple inductive claim shows that if E happens then the last request in Π will be assigned to the server at p_0 even though the request is for p_{k+1}. Thus, the expected cost for the last request in Π is at least

$$q(p_{k+1} - p_0) = \left(\frac{m-1}{m}\right)^k (2^k + \varepsilon) > \left(\frac{2(m-1)}{m}\right)^k.$$

Since the number of requests $n = \sum_{i=0}^{k+1}\delta_k = m^k$ we obtain $k = \ln(n)/\ln(m)$. It follows that for any integer parameter m, the expected total cost of the greedy algorithm is at least

$$\left(\left(\frac{2(m-1)}{m}\right)^{\frac{1}{\ln m}}\right)^{\ln(n)}$$

which is maximized for $m = 4$ and yields a lower bound of

$$\left(\left(\frac{3}{2}\right)^{\frac{1}{\ln(4)}}\right)^{\ln(n)} = n^{\ln\left(\left(\frac{3}{2}\right)^{\frac{1}{\ln(4)}}\right)} = n^{\frac{\ln(3)-\ln(2)}{\ln(4)}} \approx n^{0.292}.$$

The claim follows since $\mathsf{OPT} = 1+\varepsilon$. □

Acknowledgements

The authors thank the area editor and anonymous reviewer. The research of the first author is supported by EPSRC grants EP/L011018/1 and EP/J019399/1. The research of the second author was carried out in the framework of MATHEON supported by Einstein Foundation Berlin.

References

[1] A. Antoniadis, N. Barcelo, M. Nugent, K. Pruhs, M. Scquizzato, A $o(n)$-competitive deterministic algorithm for online matching on a line, in: Proc. 12th Internat. Workshop Approximation and Online Algorithms, in: WAOA, 2014, pp. 11–22.

[2] R.A. Baeza-Yates, J.C. Culberson, G.J.E. Rawlins, Searching in the plane, Inform. and Comput. 106 (2) (1993) 234–252.

[3] N. Bansal, N. Buchbinder, A. Gupta, J. Naor, A randomized $O(\log^2 k)$-competitive algorithm for metric bipartite matching, Algorithmica 68 (2) (2014) 390–403.

[4] B. Csaba, A. Pluhár, A randomized algorithm for the on-line weighted bipartite matching problem, J. Sched. 11 (6) (2008) 449–455.

[5] B. Fuchs, W. Hochstätter, W. Kern, Online matching on a line, Theoret. Comput. Sci. 332 (1–3) (2005) 251–264.

[6] G. Goel, A. Mehta, Online budgeted matching in random input models with applications to adwords, in: Proc. 19th Annual ACM-SIAM Sympos. Discrete Algorithms, in: SODA, 2008, pp. 982–991.

[7] A. Gupta, K. Lewi, The online metric matching problem for doubling metrics, in: Proc. 39th Internat. Colloquium Automata, Languages, and Programming, in: ICALP, 2012, pp. 424–435.

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

[9] B. Kalyanasundaram, K. Pruhs, On-line network optimization problems, in: A. Fiat, G.J. Woeginger (Eds.), Online Algorithms, in: LNCS, vol. 1442, Springer, 1998, pp. 268–280.

[10] M.-Y. Kao, J.H. Reif, S.R. Tate, Searching in an unknown environment: An optimal randomized algorithm for the cow-path problem, Inform. and Comput. 131 (1) (1996) 63–79.

[11] R.M. Karp, U.V. Vazirani, V.V. Vazirani, An optimal algorithm for on-line bipartite matching, in: Proc. 22nd Annual ACM Sympos. Theory Comput., in: STOC, 1990, pp. 352–358.

[12] S. Khuller, S.G. Mitchell, V.V. Vazirani, On-line algorithms for weighted bipartite matching and stable marriages, Theoret. Comput. Sci. 127 (2) (1994) 255–267.

[13] E. Koutsoupias, A. Nanavati, The online matching problem on a line, in: Proc. 1st Internat. Workshop Approximation and Online Algorithms, in: WAOA, 2003, pp. 179–191.

[14] E. Koutsoupias, C.H. Papadimitriou, On the k-server conjecture, J. ACM 42 (5) (1995) 971–983.

[15] A. Meyerson, A. Nanavati, L. Poplawski, Randomized online algorithms for minimum metric bipartite matching, in: Proc. 17th Annual ACM-SIAM Sympos. Discrete Algorithms, in: SODA, 2006, pp. 954–959.

[16] K. Nayyar, S. Raghvendra, An input sensitive online algorithm for the metric bipartite matching problem, in: Proc. 58th IEEE Annual Sympos. Foundation Comput. Sci., in: FOCS, 2017, pp. 505–515.

[17] S. Raghvendra, A robust and optimal online algorithm for minimum metric bipartite matching, in: Proc. 19th Internat. Workshop Approximation Algorithms for Combinatorial Optim. Problems, in: APPROX, 2016, pp. 18:1–18:16.

[18] S. Raghvendra, Optimal analysis of an online algorithm for the bipartite matching problem on a line, in: Proc. 34th Internat. Sympos. Computational Geometry, in: SOCG, 2018, pp. 67:1–67:14.