Matching on the line admits no $o(\sqrt{\log n})$-competitive algorithm

ENOCH PESERICO, Università degli Studi di Padova, Italy
MICHELE SCQUIZZATO, Università degli Studi di Padova, Italy

We present a simple proof that no randomized online matching algorithm for the line can be $(\sqrt{\log_2(n+1)}/15)$-competitive against an oblivious adversary for any $n = 2^i - 1 : i \in \mathbb{N}$. This is the first super-constant lower bound for the problem, and disproves as a corollary a recent conjecture on the topology-parametrized competitiveness achievable on generic spaces.

CCS Concepts: • **Theory of computation** → **Online algorithms**.

Additional Key Words and Phrases: metric matching, competitive analysis

ACM Reference Format:
Enoch Peserico and Michele Scquizzato. 2023. Matching on the line admits no $o(\sqrt{\log n})$-competitive algorithm. 1, 1 (July 2023), 4 pages. https://doi.org/10.1145/3594873

1 ONLINE MATCHING, ON THE LINE

In *online metric matching* [8, 11] n points of a metric space are designated as *servers*. One by one n additional points are designated as *requests*, and each must be immediately matched to a yet-unmatched server at a cost equal to their distance. Matchings should minimize the ratio between the total cost and the minimum cost attainable *offline*, i.e. if all requests were known beforehand. A randomized matching algorithm is $c(n)$*-competitive* if said ratio does not exceed $c(n)$ in expectation under any placement of servers and requests independent of the algorithm's random choices (the so-called *oblivious adversary* model).

It is widely acknowledged [1, 12, 17] that the line is the most interesting metric space for the problem. Matching on the line models many scenarios, like a shop that must rent to customers skis of approximately their height, where a stream of requests must be serviced with minimally mismatched items from a known store. Despite matching being specifically studied on the line since at least 1996 [9] no tight competitiveness bounds are known.

As for upper bounds, the line is a doubling space and thus admits an $O(\log n)$-competitive randomized algorithm [6]. A sequence of recent results [1, 14, 16] yielded the same ratio without randomization. Better bounds have been obtained only in specific cases, such as when requests are drawn independently from a known probability distribution [5]; or by algorithms with additional capabilities, such as re-assigning past requests [7, 13] or predicting future ones [2], or matching up to two requests to each server [10].

As for lower bounds, the competitive ratio cannot be less than 4.591 for randomized algorithms and 9 for deterministic ones since the *cow-path* problem is a special case of matching on the line [9]. These bounds were conjectured tight [9]

A preliminary version [15] appeared in the proceedings of the 48th International Colloquium on Automata, Languages, and Programming (ICALP 2021).
Authors' addresses: Enoch Peserico, Università degli Studi di Padova, Dipartimento di Ingegneria dell'Informazione, via Gradenigo 6A, 35131 Padova, Italy, enoch@dei.unipd.it; Michele Scquizzato, Università degli Studi di Padova, Dipartimento di Matematica, via Trieste 63, 35121 Padova, Italy, scquizza@math.unipd.it.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2023 Association for Computing Machinery.
Manuscript submitted to ACM

until a complex adversarial strategy yielded a lower bound of 9.001 for deterministic algorithms [4]. Beyond some $\Omega(\log n)$ bounds for restricted classes of algorithms [3, 12, 14] there has been no further progress on the lower-bound side before this work.

2 AN $\Omega(\sqrt{\log n})$ COMPETITIVENESS BOUND

We prove a simple $\Omega(\sqrt{\log n})$ lower bound on the competitive ratio of randomized online matching algorithms for the line.

For any $n = 2^i - 1$ with $i \in \mathbb{N}$ consider the $[0, n{+}1]$ interval. For each positive integer $j \le n$ place a server at j. Then place n requests over $\log_2(n{+}1)$ rounds, as in Figure 1: on the r^{th} round partition the main interval into $(n{+}1)/2^r$ sub-intervals of length 2^r, choose within each uniformly and independently at random an *origin* point, and for each origin (in an arbitrary order) place a request on the closest integer multiple of 2^{-n} breaking ties arbitrarily. "Discretizing" requests instead of directly using the corresponding origins prevents some technical difficulties – see our remark at the end of the section.

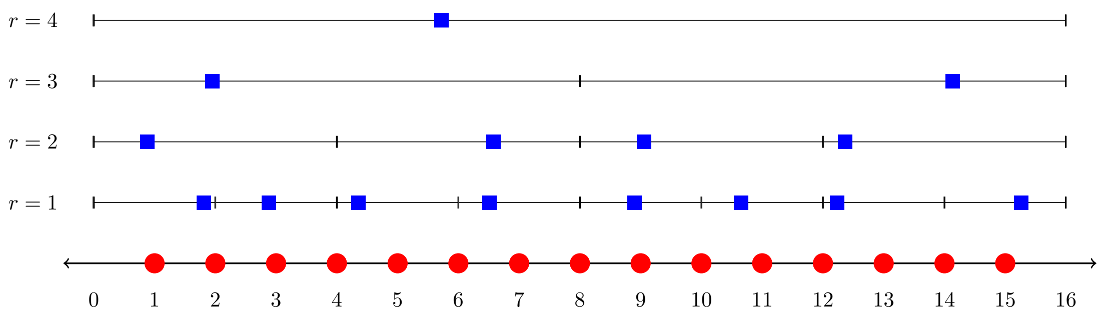

Fig. 1. An input instance for $n = 15$. Servers are red circles, requests/origins blue squares.

We prove in Lemma 2.1 that *any* online matching algorithm ALG incurs an expected cost $\Omega(n)$ in any given round, for a total cost $\Omega(n \log n)$. Conversely, we prove in Lemma 2.2 that the expected distance between the ℓ^{th} leftmost server and the ℓ^{th} leftmost origin is $O(\sqrt{\log n})$, so servers and requests can be matched with an expected offline cost $O(n\sqrt{\log n})$. We combine the two results in Theorem 2.3 to prove that on *some* request sequence ALG incurs $\Omega(\sqrt{\log n})$ times the minimum offline cost.

Lemma 2.1. *Any randomized online matching algorithm incurs an expected cost greater than* $(n{+}1)/12$ *in each round.*

Proof. Consider an origin placed uniformly at random in a sub-interval of size 2^r during the r^{th} round. Assume $m \ge 0$ unmatched servers in the interior points of that sub-interval divide it into $m + 1$ segments of (integer) length $d_0, \dots, d_m$. Then the probability an origin falls inside a segment of length d is $d/2^r$, in which case the expected distance of both the origin and the corresponding request from the segment's closer endpoint is $d/4$. Adding over all the s_r segments in all the round's sub-intervals, applying Jensen's inequality, and noting that s_r does not exceed the number of sub-intervals (i.e. $(n{+}1)/2^r$) plus the total number of unmatched servers (i.e. $(n{+}1)/2^{r-1} - 1$), the expected cost to service all requests in the round is at least

$$\sum_{h=1}^{s_r} \frac{d_h}{4} \cdot \frac{d_h}{2^r} \geq \frac{1}{4 \cdot 2^r} s_r \left(\frac{n+1}{s_r}\right)^2 > \frac{(n+1)^2}{4 \cdot 2^r} \cdot \frac{2^r}{3(n+1)} = \frac{n+1}{12}. \qquad \square$$

Lemma 2.2. *For $1 \leq \ell \leq n$ the expected distance between the ℓ^{th} leftmost server and the ℓ^{th} leftmost origin (breaking ties arbitrarily) is at most $\sqrt{\log_2(n+1)} + 3$.*

Proof. Assume origins and servers are all distinct, since the probability they are not is 0. Let S_ℓ be ℓ^{th} leftmost server, g_ℓ the number of origins to its left, and d_ℓ its distance to the ℓ^{th} leftmost origin O_ℓ.

For any $h \geq 0$, if $g_\ell = \ell + h$ or $g_\ell = (\ell - 1) - h$, then h origins lie between S_ℓ and O_ℓ; and since the first round placed one origin in every sub-interval of size 2, then $d_\ell < 3 + 2h$. In other words, $\forall \epsilon \in]0, 1[,\ d_\ell < 3 + 2\lfloor |g_\ell - (\ell - \epsilon)| \rfloor$. One can thus bound d_ℓ by bounding $\gamma_\ell = |g_\ell - (\ell - \frac{\ell}{n+1})| = |g_\ell - \frac{n}{n+1}\ell|$, which is the absolute deviation from the mean of g_ℓ since the expected density of origins equals $\frac{n}{n+1}$ at every non-integer point of the $[0, n+1]$ interval.

g_ℓ is the sum of n independent indicator random variables, each denoting whether a given origin was placed to the left of S_ℓ. At most one such variable in a given round has variance greater than 0 (the variable corresponding to the origin placed in a sub-interval with S_ℓ strictly in its interior) and none greater than $1/4$; thus the variance $E[\gamma_\ell^2]$ of their sum is at most $\log_2(n+1)/4$. By Jensen's inequality $E[\gamma_\ell] \leq \sqrt{E[\gamma_\ell^2]}$, so $E[d_\ell] \leq 2E[\gamma_\ell] + 3 \leq \sqrt{\log_2(n+1)} + 3$. $\square$

Theorem 2.3. *No randomized online matching algorithm for the line can be $(\sqrt{\log_2(n+1)}/15)$-competitive against an oblivious adversary for any $n = 2^i - 1 : i \in \mathbb{N}$.*

Proof. Let $C_A(\sigma)$ be the expected cost incurred by a randomized online matching algorithm ALG on a request sequence σ, and $C_O(\sigma)$ the minimum offline cost; and let p_σ be the probability of generating σ through the origin-request process described earlier. Since $(\sum_i a_i)/(\sum_i b_i)$ is for all positive a_i, b_i a convex linear combination of the individual ratios a_i/b_i, focusing on the case $\sqrt{\log_2(n+1)}/15 \geq 1$ in which $\sqrt{\log_2(n+1)} + 3 + 2^{-n} < (5/4)\sqrt{\log_2(n+1)}$ we have

$$\max_{\sigma:p_\sigma \neq 0} \frac{C_A(\sigma)}{C_O(\sigma)} \geq \frac{\sum_{\sigma:p_\sigma \neq 0} C_A(\sigma) p_\sigma}{\sum_{\sigma:p_\sigma \neq 0} C_O(\sigma) p_\sigma} > \frac{(n+1)\log_2(n+1)/12}{n(\sqrt{\log_2(n+1)} + 3 + 2^{-n})} > \frac{\sqrt{\log_2(n+1)}}{15}. \qquad \square$$

Remark. Without discretized requests the term $\sum_{\sigma:p_\sigma \neq 0} C_A(\sigma) p_\sigma$ in the theorem's proof would have been an integral, potentially ill-defined: for example, if ALG serviced requests for rational points in an interval with one server and for irrational points with another.

3 CONCLUSIONS

The "major open question [of] whether there exists an $O(1)$-competitive online algorithm (deterministic or randomized) on the line" [13] for the metric matching problem has a negative answer – via a simple, self-contained proof.

A related question on online matching algorithms defined on all metric spaces then has a negative answer, too. Consider a space with a non-zero metric. For any set X of n points, let d_X be its diameter and c_X the length of its shortest circuit; and let $\mu = \sup_{X:d_X \neq 0}(c_X/d_X)$. Recent work proved no deterministic algorithm can be $o(\mu)$-competitive on *any* space; developed a deterministic algorithm $O(\mu \log^2 n)$-competitive on *every* space; and asked whether there might be one with a similarly universal $O(\mu)$ competitive ratio [14]. An $\Omega(\sqrt{\log n})$ competitiveness bound on the line, where $\mu = 2$, implies this is not the case. Note that in any space with only 2 (distinct) points $\mu = 2$ and the greedy algorithm's competitive ratio is $1 = O(\mu) = o(\mu\sqrt{\log n})$; so μ might not be the "right" parameter to capture the competitive ratio achievable on generic spaces.

In fact, the $\Omega(\sqrt{\log n})$ competitiveness bound applies even to online algorithms with significant extra power in terms of lookahead or request re-assignment. In particular, Lemma 2.1 holds even if an algorithm can observe all requests in a given round before servicing any, or equivalently if it is allowed to re-assign any of the current round's requests. Also, it holds asymptotically even if one ignores the cost of all requests save those in an arbitrary fixed sub-interval of length $\Omega(n)$. Finally, $\Omega(\log n)$ rounds are sufficient to translate Lemma 2.1 into the main $\Omega(\sqrt{\log n})$ competitiveness bound. The bound then applies asymptotically for any $\epsilon = \Omega(1)$ even to online algorithms with advance knowledge of the next $n^{1-\epsilon}$ requests and the power to re-assign the most recent $n^{1-\epsilon}$; and/or with advance knowledge of, and the power to re-assign, all requests save the final n^{ϵ}.

ACKNOWLEDGMENTS

We thank Kirk Pruhs and the anonymous reviewers for their constructive criticism and insightful observations. This work was partially supported by the Italian National Center for HPC, Big Data, and Quantum Computing, and by Univ. Padova grant BIRD197859/19.

REFERENCES

[1] Antonios Antoniadis, Neal Barcelo, Michael Nugent, Kirk Pruhs, and Michele Scquizzato. 2019. A $o(n)$-competitive deterministic algorithm for online matching on a line. *Algorithmica* 81, 7 (2019), 2917–2933.

[2] Antonios Antoniadis, Christian Coester, Marek Eliáš, Adam Polak, and Bertrand Simon. 2023. Online metric algorithms with untrusted predictions. *ACM Trans. Algorithms* 19, 2, Article 19 (2023).

[3] Antonios Antoniadis, Carsten Fischer, and Andreas Tönnis. 2018. A collection of lower bounds for online matching on the line. In *Proceedings of the 13th Latin American Theoretical Informatics Symposium (LATIN)*. 52–65.

[4] Bernhard Fuchs, Winfried Hochstättler, and Walter Kern. 2005. Online matching on a line. *Theor. Comput. Sci.* 332, 1-3 (2005), 251–264.

[5] Anupam Gupta, Guru Guruganesh, Binghui Peng, and David Wajc. 2019. Stochastic online metric matching. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP)*. 67:1–67:14.

[6] Anupam Gupta and Kevin Lewi. 2012. The online metric matching problem for doubling metrics. In *Proceedings of the 39th International Colloquium on Automata, Languages, and Programming (ICALP)*. 424–435.

[7] Varun Gupta, Ravishankar Krishnaswamy, and Sai Sandeep. 2020. Permutation strikes back: the power of recourse in online metric matching. In *Proceedings of the 23rd International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*. 40:1–40:20.

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

[9] Bala Kalyanasundaram and Kirk Pruhs. 1998. Online network optimization problems. In *Online Algorithms: The State of the Art*. Springer-Verlag, 268–280. From the Dagstuhl Seminar on Online Algorithms, 1996.

[10] Bala Kalyanasundaram and Kirk Pruhs. 2000. The online transportation problem. *SIAM J. Discrete Math.* 13, 3 (2000), 370–383.

[11] Samir Khuller, Stephen G. Mitchell, and Vijay V. Vazirani. 1994. On-line algorithms for weighted bipartite matching and stable marriages. *Theor. Comput. Sci.* 127, 2 (1994), 255–267.

[12] Elias Koutsoupias and Akash Nanavati. 2003. The online matching problem on a line. In *Proceedings of the 1st International Workshop on Approximation and Online Algorithms (WAOA)*. 179–191.

[13] Nicole Megow and Lukas Nölke. 2020. Online minimum cost matching with recourse on the line. In *Proceedings of the 23rd International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*. 37:1–37:16.

[14] Krati Nayyar and Sharath Raghvendra. 2017. An input sensitive online algorithm for the metric bipartite matching problem. In *Proceedings of the 58th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*. 505–515.

[15] Enoch Peserico and Michele Scquizzato. 2021. 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)*. 103:1–103:3.

[16] Sharath Raghvendra. 2018. Optimal analysis of an online algorithm for the bipartite matching problem on a line. In *Proceedings of the 34th International Symposium on Computational Geometry (SoCG)*. 67:1–67:14.

[17] Rob van Stee. 2016. SIGACT News online algorithms column 27: online matching on the line, part 1. *SIGACT News* 47, 1 (2016), 99–110.