PERMUTATION for Online Metric Matching with m Distinct Servers

Alberto Nicoletti[a], Enoch Peserico[b], Michele Scquizzato[a,*]

[a] Dipartimento di Matematica, Università degli Studi di Padova
[b] Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Padova

Abstract

We prove that the $2n-1$ lower bound on the competitive ratio of the PERMUTATION algorithm for online metric matching with n servers holds even if the number m of distinct servers is as low as 4, and even on the line, a simple space where lower competitive ratios are often possible. The same technique yields a $(2m-3) - \frac{2m(m-2)(m-3)}{2n+m(m-2)}$ lower bound on the competitive ratio of PERMUTATION in the special case of n/m servers at each of m evenly-spaced points on the line, improving for all $m > 4$ a recent $m+1$ lower bound and disproving its conjectured optimality.

Keywords: Metric matching, Online algorithms, Competitive analysis

1. Introduction

In *online metric matching* [1, 2] n points of a metric space are designated as *servers*. Then n additional points are designated one by one as *requests*, and each must be immediately matched to a yet-unmatched server at a cost equal to their distance. An online matching algorithm is *c-competitive* if, under any placement of servers and requests, it incurs a total cost of at most c times the minimum total cost attainable *offline*, i.e. if all requests were known beforehand; and its *competitive ratio* is the infimum of all such c [3].

Competitiveness for online metric matching is often expressed as a function of the number of servers n. No deterministic algorithm can be better

*Correspondence regarding this letter should be addressed to:
Michele Scquizzato, Dipartimento di Matematica, Università degli Studi di Padova, via Trieste 63, 35121 Padova (PD), Italy, email: scquizza@math.unipd.it

than $(2n-1)$-competitive, at least on some metric spaces. This competitiveness is achieved on every metric space by the simple algorithm Permutation matching the i-th request to a server that, together with the $i-1$ servers matched to all previous requests, would yield a minimum-cost matching of the first i requests [1, 2]. Better results can be achieved on specific metric spaces, or resorting to randomization.

A different parameter of interest is the number m of *distinct* servers. m significantly smaller than n captures scenarios with a limited number of solutions available for a large number of tasks: for example, matching apparel produced in a small set of colours/sizes to a global clientele, or parking cars in a few high-capacity garages—in this sense online metric matching with a limited number of distinct servers is sometimes called the *online transportation problem* [4].

[4] conjectured the existence of a deterministic algorithm that is $(2m-1)$-competitive on all metric spaces, implying that m is the "right" parameter to study the problem. In fact, this conjecture was proved correct within a factor 4 by a $(8m-5)$-competitive algorithm [5]. However, the algorithm and its analysis are quite complex, even after some recent simplifications [6]; and while [4] stated that the competitive ratio of the much simpler Permutation algorithm cannot be bounded by any function of m, no proof has ever appeared in the literature.

This article fills the gap by showing a (tight) $2n-1$ lower bound on the competitive ratio of Permutation that holds for all $m \geq 4$; and in fact holds even on the line, a simple metric space on which the competitive ratio achievable by deterministic algorithms is polylogarithmic rather than linear in n [7, 8]. Our technique also yields a $(2m-3) - \frac{2m(m-2)(m-3)}{2n+m(m-2)}$ lower bound for the competitiveness of Permutation in the special case of n/m servers at each of m evenly-spaced points on the line, that has received some recent attention [9, 10, 11, 12]; in particular, our result improves for all $m > 4$ the $m+1$ lower bound of [12], disproving its conjectured optimality and providing an even stronger counterexample to the incorrect m upper bound of [9].

2. The competitive ratio of Permutation is $2n-1$ for all $m \geq 4$

Our main result is that, even when the number of distinct servers m is as low as 4, the competitive ratio of Permutation remains $2n-1$, as high as when m is unconstrained [1, 2]. The bound holds even on the line, improving previous lower bounds (m [2] and $m+1$ [12], or n [2] and $n+1$ [12] as a function of n), and showing it is a worst-case metric space for the competitiveness of Permutation.

Theorem 1. *The competitive ratio of* Permutation *is* $2n-1$ *for all* $m \geq 4$.

Proof. Recall that the competitive ratio of Permutation is at most $2n-1$ on every metric space, regardless of the number of distinct servers [1, 2]; we prove that this bound is tight for all $m \geq 4$.

Consider the line metric, and let us begin with the case $m = 4$. Place one server at -1 and one server at 1; for some $\delta > 0$, and any $n \geq 4$ place $\lceil n/2 \rceil - 1$ servers at $-1-2\delta$ and $\lfloor n/2 \rfloor - 1$ servers at $1+2\delta$. For some positive $\epsilon < \delta/4(n-1)$, the requests $r_1, r_2, \ldots, r_n$ are sequentially placed as follows:

$$\begin{aligned}
r_1 &= 0 - \epsilon \\
r_2 &= -1 - \delta + 2\epsilon \\
r_3 &= 1 + \delta - 4\epsilon \\
r_4 &= -1 - \delta + 6\epsilon \\
&\vdots \\
r_n &= (-1)^n(-1 - \delta + 2(n-1)\epsilon).
\end{aligned}$$

Permutation matches r_1 to the server placed at -1, r_2 to the server placed at 1, r_3 to a server placed at $-1-2\delta$, and the remaining requests alternating between servers placed at the two outermost server sites (see Figure 1). The cost to match r_1 is $1-\epsilon$, and all subsequent requests are matched at a cost greater than $(\delta - 2n\epsilon) + 2 + 2\delta$.

An offline algorithm can match the ℓ-th leftmost request to the ℓ-th leftmost server, for a total cost less than $(1+\epsilon) + (n-1)(\delta + 2n\epsilon)$. Thus, the ratio between the cost incurred by Permutation and that incurred by an optimal offline algorithm on this input is greater than

$$\frac{(1-\epsilon) + (n-1)((\delta - 2n\epsilon) + 2 + 2\delta)}{(1+\epsilon) + (n-1)(\delta + 2n\epsilon)},$$

which for all sufficiently small δ and ϵ is arbitrarily close to $1 + (n-1)2 = 2n-1$.

The construction above uses $m = 4$ distinct servers, but can immediately be extended to any larger value of m by shifting any number of servers a

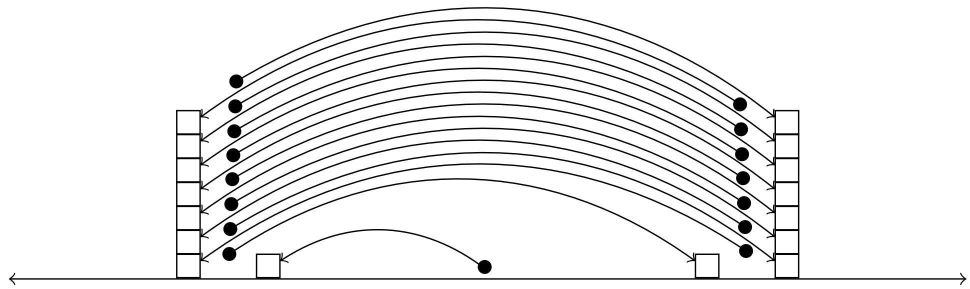

Figure 1: Permutation on a hard input with $n = 16$. Servers are represented by squares, requests by black circles.

sufficiently small distance such that neither Permutation nor any optimal offline algorithm changes its request-server pairings, and such that the cost incurred by each algorithm changes by an arbitrarily small amount. □

3. How does Permutation fare for $m < 4$?

The $2n - 1$ lower bound of [1, 2] immediately yields a $2m - 1$ lower bound on the competitive ratio of *any* deterministic algorithm. In fact, for Permutation this lower bound holds even on the line for $m < 4$, even for $n = m$. This is trivial for $m = 1$. For $m = 2$, consider two servers at -1 and 1, and place a first request at ϵ for some $\epsilon \in]0, 1[$ and a second at 1; Permutation matches them at a total cost of $(1 - \epsilon) + 2 = 3 - \epsilon$, instead of the offline optimal of $(1 + \epsilon) + 0$, yielding a competitive ratio of $3 = 2m-1$ as $\epsilon \to 0$. For $m = 3$, consider 3 servers at -1, 1, $(1+2\delta)$, and the sequence of requests $\epsilon, (1 + \delta - 2\epsilon), -1$, for some $\delta \in]0, 1]$ and some $\epsilon \in]0, \delta]$; Permutation matches them at a total cost of $(1-\epsilon)+(\delta-2\epsilon+2)+(2+2\delta) = 5 + 3\delta - 3\epsilon$, instead of the offline optimal of $(1 - \epsilon) + (\delta + 2\epsilon) + 0 = 1 + \delta + \epsilon$, yielding a competitive ratio of $5 = 2m - 1$ as $\delta \to 0$. We conjecture that this $2m - 1$ lower bound is in fact tight for all n:

Conjecture 1. *The competitive ratio of* Permutation *is $2m - 1$ for all $m < 4$.*

The upper bound that would prove the conjecture is trivial for $m = 1$, and easy to obtain for $m = 2$ as a corollary of Theorem 2.4 in [1], which shows that a variant of Permutation is $(2t-1)$-competitive for online metric matching if the requests arrive in t batches. This variant computes an optimal partial matching after each new batch of requests arrives, and the difference between the sets of servers used in the last two optimal partial matchings is the set of servers matched (optimally) to the new batch. Then, given a generic request sequence $r_1, \ldots, r_n$ serviced by Permutation let r_i be the earliest request before which all remaining unmatched servers coincide, and consider the 2 batches $r_1, \ldots, r_{i-1}$ and $r_i, \ldots, r_n$. If the batched variant of Permutation breaks all ties as Permutation does, both service the first batch identically, matching each request to the closest server; and they also service the second batch identically, since all servers then remaining coincide. Thus they are both 3-competitive.

The $2m-1$ upper bound on the competitive ratio of Permutation holds when $m = 3$ too, if $n = m$ [1, 2]; proving it holds if $n > m$ would then prove the conjecture.

4. The Special Case of Even Spacing on the Line

The proof technique used in Section 2 can be readily adapted to address the special case of n servers placed in groups of n/m at m evenly-spaced points on the line, yielding a lower bound stronger than the $m + 1$ lower bound of [12] for $m \geq 5$ and n/m as low as 5.

Theorem 2. *The competitive ratio of* Permutation *on the line with $m \geq 4$ evenly-spaced distinct servers, each coinciding with $n/m - 1$ additional ones, is at least $(2m - 3) - \frac{2m(m-2)(m-3)}{2n+m(m-2)}$.*

Proof. Begin by placing the 4 outermost servers at $(-1-2\delta)$, -1, 1, $(1+2\delta)$ as in Theorem 1. Then place $m - 4$ additional servers so as to partition the $[-1, 1]$ interval into $m-3$ subintervals of length $2/(m-3)$. If $\delta = 1/(m-3)$ all m servers are now evenly spaced. Finally, place $n/m-1$ additional servers at each of the m points.

Now let $u = 2(n/m + 1)$, i.e. $u/2 - 1 = n/m$. Place $u/2 - 2$ requests at -1, $u/2 - 2$ at 1, and $u/2 - 1$ at each point coinciding with a server between -1 and 1. These requests are obviously matched at 0 cost by both Permutation and an optimal offline algorithm, and recreate exactly the same scenario of the proof of Theorem 1, except that there are now u rather than n unmatched servers: $u/2 - 1$ at each of $1 - 2\delta$ and $1 + 2\delta$, and 1 at each of -1 and 1. The same request sequence and analysis of Theorem 1, simply replacing n with u in all formulas, then yield a lower bound on the competitive ratio equal to

$$\frac{(1-\epsilon)+(u-1)((\delta-2u\epsilon)+2+2\delta)}{(1+\epsilon)+(u-1)(\delta+2u\epsilon)},$$

which for all sufficiently small ϵ (note that the equispacing constraint does not allow us to push δ to 0) is arbitrarily close to

$$\begin{aligned}\frac{1+(u-1)(2+3\delta)}{1+(u-1)\delta} &= \frac{2+3\delta}{\delta} - \frac{2+2\delta}{\delta(1+(u-1)\delta)} \\ &= (2m-3) - \frac{2m(m-2)(m-3)}{2n+m(m-2)}. \qquad \square\end{aligned}$$

Acknowledgements

The authors thank Kirk Pruhs for the central idea in the construction yielding Theorem 1. Michele Scquizzato was supported in part by the Italian National Center for HPC, Big Data, and Quantum Computing.

References

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

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

[3] A. Borodin, R. El-Yaniv, Online Computation and Competitive Analysis, Cambridge University Press, 1998.

[4] B. Kalyanasundaram, K. Pruhs, Online network optimization problems, in: Online Algorithms: The State of the Art, Springer-Verlag, 1998, pp. 268–280, from the Dagstuhl Seminar on Online Algorithms, 1996.

[5] T. Harada, T. Itoh, A nearly optimal deterministic algorithm for online transportation problem, in: Proceedings of the 52nd International Colloquium on Automata, Languages, and Programming (ICALP), 2025, pp. 94:1–94:17.

[6] S. Arndt, B. Moseley, K. Pruhs, M. Uetz, Competitive online transportation simplified, in: Proceedings of the 2026 SIAM Symposium on Simplicity in Algorithms (SOSA), 2026, pp. 463–471.

[7] S. Raghvendra, 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), 2018, pp. 67:1–67:14.

[8] E. Peserico, M. Scquizzato, Matching on the line admits no $o(\sqrt{\log n})$-competitive algorithm, ACM Trans. Algorithms 19 (3) (2023) 28:1–28:4.

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

[10] T. Itoh, S. Miyazaki, M. Satake, Competitive analysis for two variants of online metric matching problem, Discret. Math. Algorithms Appl. 13 (6) (2021) 2150156:1–2150156:16.

[11] T. Harada, T. Itoh, S. Miyazaki, Capacity-insensitive algorithms for online facility assignment problems on a line, Discret. Math. Algorithms Appl. 16 (5) (2024) 2350057:1–2350057:39.

[12] T. Harada, A lower bound on the competitive ratio of the permutation algorithm for online facility assignment on a line (2024). arXiv:2402.12734, URL: https://arxiv.org/abs/2402.12734.