Impatient Online Matching

Xingwu Liu
SKL Computer Architecture, ICT, CAS
University of Chinese Academy of Sciences, Beijing, China
liuxingwu@ict.ac.cn

Zhida Pan
Institute of Computing Technology, Chinese Academy of Sciences
University of Chinese Academy of Sciences, Beijing, China
zhidapan@gmail.com

Yuyi Wang
ETH Zurich, Switzerland
yuwang@ethz.ch

Roger Wattenhofer
ETH Zurich, Switzerland
wattenhofer@ethz.ch

Abstract

We investigate the problem of Min-cost Perfect Matching with Delays (MPMD) in which requests are pairwise matched in an online fashion with the objective to minimize the sum of space cost and time cost. Though linear-MPMD (i.e., time cost is linear in delay) has been thoroughly studied in the literature, it does not well model impatient requests that are common in practice. Thus, we propose convex-MPMD where time cost functions are convex, capturing the situation where time cost increases faster and faster. Since the existing algorithms for linear-MPMD are not competitive any more, we devise a new deterministic algorithm for convex-MPMD problems. For a large class of convex time cost functions, our algorithm achieves a competitive ratio of $O(k)$ on any k-point uniform metric space. Moreover, our deterministic algorithm is asymptotically optimal, which uncover a substantial difference between convex-MPMD and linear-MPMD which allows a deterministic algorithm with constant competitive ratio on any uniform metric space.

2012 ACM Subject Classification Theory of computation → Online algorithms

Keywords and phrases online algorithm, online matching, convex function, competitive analysis, lower bound

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2018.62

Acknowledgements This work is partially supported by the National Key Research and Development Program of China (Grant No. 2016YFB1000200), the National Natural Science Foundation of China (61420106013), State Key Laboratory of Computer Architecture Open Fund (CARCH3410).

1 Introduction

Online matching has been studied frantically in the last years. Emek et al. [10] started the renaissance by introducing delays and optimizing the trade-off between timeliness and quality of the matching. This new paradigm leads to the problem of Min-cost Perfect Matching with Delays (MPMD for short), where requests arrive in an online fashion and need to be matched with one another up to delays. Any solution experiences two kinds of costs or penalty. One

29th International Symposium on Algorithms and Computation (ISAAC 2018).
Editors: Wen-Lian Hsu, Der-Tsai Lee, and Chung-Shou Liao; Article No. 62; pp. 62:1–62:12
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

is for quality: Matching two requests of different types incurs cost as such do not match well, while requests of the same type should be matched for free. The other is for timeliness: Delay in matching a request causes a cost that is an increasing function, called the time cost function, of the waiting time. The overall objective is to minimize the sum of the two kinds of costs.

Tractable in theory and fascinating in practice, the MPMD problem has attracted more and more attention and inspired an increasing volume of literature [10, 11, 4, 3, 2]. However, these existing work in this line only studied linear time cost function, meaning that penalty grows at a constant rate no matter how long the delay is. This sharply contrasts to much of our real-life experience. Just imagine a dinner guest: waiting a short time is no problem – but eventually, every additional minute becomes more annoying than ever. The discontentment is experiencing convex growth, an omnipresent concept in biology, physics, engineering, or economics.

Actually, such convex growth of discontentment appears in various real-life scenarios of online matching. For instance, online game platforms often have to match pairs of players before starting a game (consider chess as an example). Players at the same, or at least similar, level of skills should be paired up so as to make a balanced game possible. Then it would be better to delay matching a player in case of no ideal candidate of opponents. Usually it is acceptable that a player waits for a short time, but a long delay may be more and more frustrating and even make players reluctant to join the platform again. Another example appears in organ transplantation: An organ transplantation recipient may be able to wait a bit, but waiting an extended time will heavily affect its health. One may think that organ transplantation would be better modeled by bipartite matching rather than regular matching as considered in this paper; however, organ-recipients and -donors usually come in incompatible pairs that will be matched with other pairs, e.g., two-way kidney exchange[1]. More real-life examples include ride sharing (match two customers), joint lease (match two roommates), just mention a few.

On this ground, we study the convex-MPMD problem, i.e., the MPMD problem with convex time cost functions. To the best of our knowledge, this is the first work on online matching with non-linear time cost.

Convexity of the time cost poses special challenges to the MPMD problem. An important technique in solving linear-MPMD, namely, MPMD with linear time cost function, is to minimize the total costs while sacrifice some requests by possibly delaying them for a long period (see, e.g., the algorithms in [4, 11, 2]). Because the time cost increases at a constant rate, it is the total waiting time, rather than waiting time of individual requests, that is of interest. Hence, keeping a request waiting is not too harmful. The case of convex time costs is completely different, since we cannot afford anymore to delay old unmatched requests, as their time costs grow faster and faster. Instead, early requests must be matched early. For this reason, existing algorithms for the linear-MPMD problem do not work any more for convex-MPMD, as confirmed by examples in Section 4.

In this paper, we devise a novel algorithm $\mathcal{A}$ for the convex-MPMD problem which is deterministic and solves the problem optimally. More importantly, our results disclose a separation: the convex-MPMD problem, even when the cost function is just a little different from linear, is strictly harder than its linear counterpart. Specifically, our main results are as follows, where f-MPMD stands for the MPMD problem with time cost function f:

[1] `https://www.hopkinsmedicine.org/transplant/programs/kidney/incompatible/paired_kidney_exchange.html`

▶ **Theorem 1.** *For any $f(t) = t^\alpha$ with constant $\alpha > 1$, the competitive ratio of $\mathcal{A}$ for f-MPMD on k-point uniform metric space is $O(k)$.*

One may wonder whether the result in Theorem 1 can be further improved because of the known result:

▶ **Theorem 2** ([4, 2])**.** *There exists a deterministic online algorithm that solves linear-MPMD on uniform metrics and reaches an $O(1)$ competitive ratio.*

However, we can show that for a large family of functions $f : \mathbb{R}^+ \to \mathbb{R}^+$, the f-MPMD problem has no deterministic algorithms of competitive ratio $o(k)$.

▶ **Theorem 3.** *Suppose that the time cost function f is nondecreasing, unbounded, continuous and satisfies $f(0) = f'(0) = 0$. Then any deterministic algorithm for f-MPMD on k-point uniform metric space has competitive ratio $\Omega(k)$.*

Numerous natural convex functions over the domain of nonnegative real numbers satisfy the conditions of Theorem 3. Examples include monomial $f(t) = t^\alpha$ with $\alpha > 1$, $f(t) = e^{\alpha t} - \alpha t - 1$ with $\alpha > 1$, and so on. This, together with Theorem 1, establishes the optimality of our deterministic algorithm. Note that family of functions satisfying the conditions of Theorem 3 is closed under multiplication and linear combination where the coefficients are positive. Hence, Theorem 3 is of general significance.

2 Related Work

Matching has became one of the most extensively studied problems in graph theory and computer science since the seminal work of Edmonds [9, 8]. Karp et al. [15] studied the matching problem in the context of online computation which inspired a number of different versions of online matching, e.g., [13, 16, 18, 19, 6, 12, 1, 7, 17, 20, 21]. In these online matching problems, underlying graphs are assumed bipartite and requests of one side are given in advance.

A matching problem where *all* requests arrive in an online manner was introduced by [10]. This paper also introduced the idea that requests are allowed to be matched with delays that need to be paid as well, so the problem is called Min-cost Perfect Matching with Delays (MPMD). They presented a randomized algorithm with competitive ratio $O(\log^2 k + \log \Delta)$ where k is the size of the underlying metric space known before the execution and Δ is the aspect ratio. Later, Azar et al. [4] proposed an almost-deterministic algorithm with competitive ratio $O(\log k)$. Ashlagi et al. [2] analyzed Emek et al.'s algorithm in a simplified way, and improved its competitive ratio to $O(\log k)$. They also extended these algorithms to bipartite matching with delays (MBPMD). The best known lower bound for MPMD is $\Omega(\log k / \log \log k)$ and MBPMD $\Omega(\sqrt{\log k / \log \log k})$ [2]. In contrast to our work, all these papers assume that the time cost of a request is linear in its waiting time.

In contrast to this previous work, we focus on the uniform metric, i.e., the distance between any two points is the same. While this is only a special case, it is an important one. In the existing linear-MPMD algorithms, a common step is to first embed a general metric to a probabilistic hierarchical separated tree (HST), which is actually an offline approach, and then design an online algorithm on the HST metric. The online algorithms on HST metrics are essentially based on algorithms on uniform metrics (or aspect-ratio-bounded metrics which can also be handled by our results) because every level of an HST can be considered as a uniform metric. Uniform metrics are known to be tricky, e.g., Emek et al. [11] study linear-MPMD with only two points. Uniform metrics also play an important role in the field of online computation [14]. For example, the k-server problem restricted to uniform metrics is the well-known paging problem.

The idea of delaying decisions has been around for a long time in the form of rent-or-buy problems (most prominently: ski rental), but [10] showed how to use delays in the context of combinatorial problems such as matching. In the classical ski rental problem [14], one can also consider the variation that the renting cost rate (to simplify our discussion, let's consider the continuous case) may change over time. If the purchase price is a constant, the renting cost rate function does not change the competitive ratio since a good deterministic online algorithm is always to buy it when the renting fee is equal to the purchase price.

Azar et al. [5] considered online service with delay, which generalizes the k-server problem. As mentioned in their paper, delay penalty functions are not restricted to be linear and even different requests can have different penalty functions. However, different delay penalty functions there do not make the service with delay problem much different, and there is a universal way to deal with these different penalty functions, unlike the online matching problems we consider now.

3 Preliminaries

In this section, we formulate the problem and introduce notations.

3.1 Problem Statement

Let $\mathbb{R}^+$ stands for the set of nonnegative real numbers.

A metric space $\mathcal{S} = (V, \mu)$ is a set V, whose members are called points, equipped with a distance function $\mu : V^2 \to \mathbb{R}^+$ which satisfies the following conditions

Positive definite: $\mu(x, y) \geq 0$ for any $x, y \in V$, and "=" holds if and only if $x = y$;
Symmetric: $\mu(x, y) = \mu(y, x)$ for any $x, y \in V$;
Subadditive or triangle inequality : $\mu(x, y) + \mu(y, z) \geq \mu(x, z)$ for any $x, y, z \in V$.

Given a function $f : \mathbb{R}^+ \to \mathbb{R}^+$, the problem f-MPMD is defined as follows, and f is called the time cost function.

For any finite metric space $\mathcal{S} = (V, \mu)$, an online input instance over $\mathcal{S}$ is a set R of requests, with any $\rho \in R$ characterized by its location $\ell(\rho) \in V$ and arrival time $t(\rho) \in \mathbb{R}^+$. Each request ρ is revealed exactly at time $t(\rho)$. Assume that $|R|$ is an even number. The goal is to construct a perfect matching, i.e. a partition into pairs, of the requests in real time without preemption.

Suppose an algorithm $\mathcal{A}$ matches $\rho, \rho' \in R$ at time T. It pays the space cost $\mu(\ell(\rho), \ell(\rho'))$ and the time cost $f(T - t(\rho)) + f(T - t(\rho'))$. The space cost of $\mathcal{A}$ on input R, denoted by $\text{cost}^s_{\mathcal{A}}(R)$, is the total space cost caused by all the matched pairs, and the time cost $\text{cost}^t_{\mathcal{A}}(R)$ is defined likewise. The objective of the f-MPMD is to find an online algorithm $\mathcal{A}$ such that $\text{cost}_{\mathcal{A}}(R) = \text{cost}^s_{\mathcal{A}}(R) + \text{cost}^t_{\mathcal{A}}(R)$ is minimized for all R.

As usual, the online algorithm $\mathcal{A}$ is evaluated through competitive analysis. Let $\mathcal{A}^*$ be an optimum offline algorithm[2]. For any finite metric space $\mathcal{S}$, if there are $a, b \in \mathbb{R}^+$ such that $\text{cost}_{\mathcal{A}}(R) \leq \text{cost}_{\mathcal{A}^*}(R)a + b$ for any online input instance R over $\mathcal{S}$, then $\mathcal{A}$ is said to be a-competitive on $\mathcal{S}$. The minimum such a is called the competitive ratio of $\mathcal{A}$ on $\mathcal{S}$. Note that both a and b can depend on $\mathcal{S}$.

This paper will focus on monomial time cost functions $f(t) = t^\alpha, \alpha > 1$ and uniform metric spaces. A metric space (V, μ) is called δ-uniform if $\mu(u, v) = \delta$ for any $u, v \in V$.

[2] An offline algorithm knows the whole input instance at the beginning and outputs any pair $\rho, \rho' \in R$ at time $\max\{t(\rho), t(\rho')\}$.

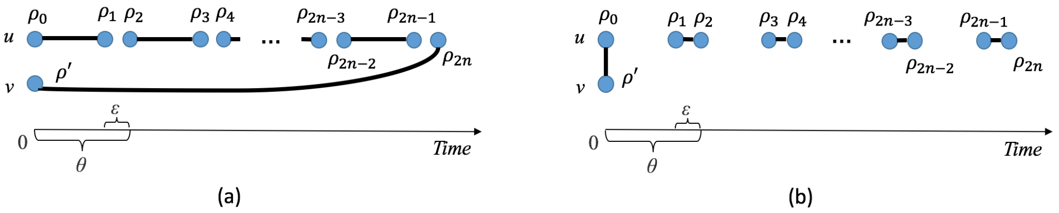

Figure 1 The input instance of Example 4. A blue dot stands for a request, and a thick line or curve for a match. (a) is the matching produced by Strategy I, while (b) is an offline solution.

3.2 Notations and Terminologies

Any pair of requests ρ, ρ' in the perfect matching is called a match between ρ and ρ' and denoted by $\langle \rho, \rho' \rangle$ or $\langle \rho', \rho \rangle$ interchangeably. A match $\langle \rho, \rho' \rangle$ is said to be *external* if $\ell(\rho) \neq \ell(\rho')$, and *internal* otherwise. For any request ρ, let $T(\rho)$ be the time when ρ is matched; ρ is said to be *pending* at any time $t \in (t(\rho), T(\rho))$ and *active* at any time $t \in [t(\rho), T(\rho)]$. At any moment t, a point $v \in V$ is called *aligned* if the number of pending requests at v under $\mathcal{A}$ and that under $\mathcal{A}^*$ have the same parity, and *misaligned* otherwise. The derivative of any differentiable function $f : \mathbb{R}^+ \to \mathbb{R}^+$ is denoted by f'.

4 Algorithm and Analysis

4.1 Basic Ideas

A natural idea to solve f-MPMD on uniform metrics is to prioritize internal matches and to create an external match only if both requests have waited long enough (say, as long as θ). However, for any monomial time cost function $f(t) = t^\alpha, \alpha > 1$, the strategy (called Strategy I) is not competitive, as illustrated in Example 4.

▶ **Example 4.** For any positive integer n and small real number $\epsilon > 0$, construct an online instance as follows. A request ρ_{2i} arrives at u at time $i \cdot \theta$ for any $0 \leq i \leq n$, while a request ρ_{2i-1} arrives at u at time $i \cdot \theta - \epsilon$ for any $1 \leq i \leq n$. Point v gets a request ρ' at time 0. By Strategy I, as in Figure 1(a), each ρ_{2i} is matched with ρ_{2i+1} for any $0 \leq i < n$, and ρ' and ρ_{2n} are matched, causing cost at least $n \cdot f(\theta - \epsilon) + f(n\theta) + \delta$. Consider the offline solution consisting of $\langle \rho', \rho_0 \rangle$ and $\langle \rho_{2i-1}, \rho_{2i} \rangle$ for $1 \leq i \leq n$, , as in Figure 1(b), which has cost $\delta + n \cdot f(\epsilon)$. When n approaches infinity and ϵ approaches 0, $n \cdot f(\theta - \epsilon) + f(n\theta) + \delta \gg \delta + n \cdot f(\epsilon)$, meaning that Strategy I is not competitive.

A plausible way to improve Strategy I is to accumulate the time costs of all the co-located requests which arrive after the last external match involving the point, and to enable an external match if both points have accumulated enough costs (say, as large as θ). Though applicable to the scenario in Example 4, this improvement (called Strategy II) remains not competitive for any time cost function $f(t) = t^\alpha, \alpha > 1$, as shown in the next example.

▶ **Example 5.** Again, consider two points u, v of distance δ. Arbitrarily fix an even integer $n > 0$ and a small real number $\epsilon > 0$. Arbitrarily choose $\tau \in \mathbb{R}^+$ such that $\theta - \epsilon < \frac{n}{2} f(\tau) < \theta$. Suppose that a request ρ' arrives at v at time 0, while a request ρ_i arrives at u at time $i\tau$ for any $0 \leq i \leq n$. Hence there are totally $n + 2$ requests. As illustrated in Figure 2(a), applying Strategy II results in the matches $\langle \rho', \rho_n \rangle$ and $\langle \rho_i, \rho_{i+1} \rangle$ for any even number $0 \leq i < n$, causing cost at least $\frac{n}{2} f(\tau) + f(n\tau) + \delta$. On the other hand, consider the offline solution $\langle \rho', \rho_0 \rangle$ and $\langle \rho_i, \rho_{i+1} \rangle$ for any odd number $0 < i < n$, as shown in Figure 2(b). It has cost

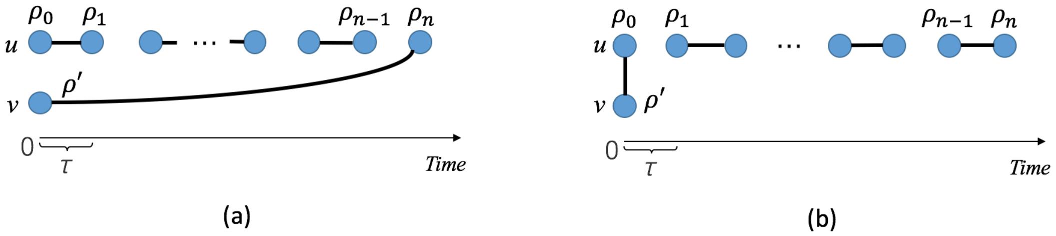

Figure 2 The input instance of Example 5. A blue dot stands for a request, and a thick line or curve for a match. (a) is the matching produced by Strategy II, while (b) is an offline solution.

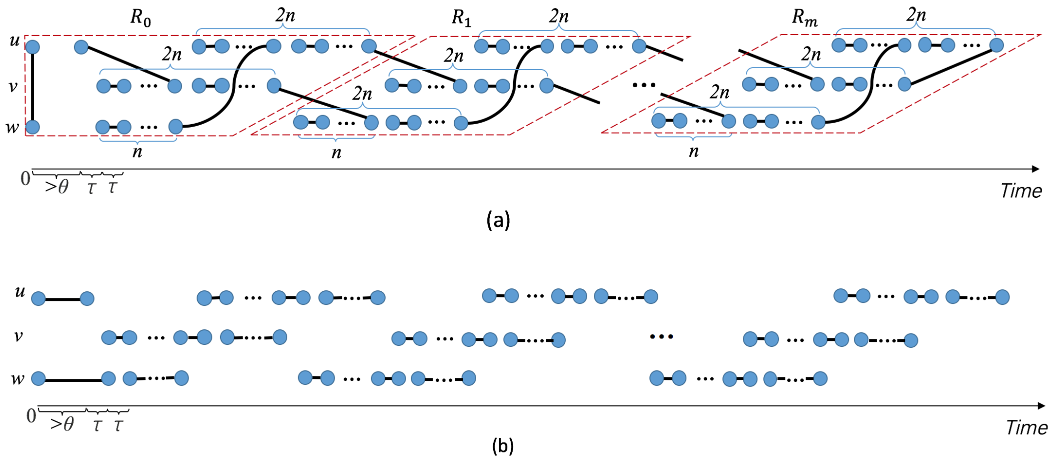

Figure 3 The input instance of Example 6. A blue dot stands for a request, an area surrounded by dash lines stands for a part of the instance, and a thick line or curve for a match. (a) is the matching produced by Strategy III, while (b) is an offline solution.

$\frac{n}{2}f(\tau)+\delta$. Thus the cost of $\mathcal{A}^*$ is at most $\frac{n}{2}f(\tau)+\delta$. When n approaches infinity and ϵ approaches 0, we have $\frac{n}{2}f(\tau)+f(n\tau)+\delta \gg \frac{n}{2}f(\tau)+\delta$, implying that Strategy II is not competitive.

Since the trouble may be rooted at the double-counter-enabling mechanism which enables an external match when two counters both reach some threshold, we further improve the strategy by enabling an external match if one of the two points has high accumulated cost (say, as high as θ). This improvement (called Strategy III) defeats both Examples 4 and 5, but the following example shows that it remains not competitive for any monomial time cost function $f(t)=t^{\alpha}, \alpha>1$.

▶ **Example 6.** Choose $\tau \in \mathbb{R}^+$ and odd integer $n>0$ such that $f(n\tau)=\theta$. Arbitrarily choose real number $T_0>f^{-1}(\theta)$. Consider a uniform metric space $\mathcal{S}=(\{u,v,w\},\delta)$. Let $m>0$ be an arbitrary integer. Construct an online input instance R which is the union of $m+1$ parts $R_0,\cdots,R_m$, as illustrated in Figure 3.

The part R_0 has $5n+3$ requests. Specifically, u receives a request $\rho^u_{0,-1}$ at time 0, $\rho^u_{0,0}$ at time T_0, and $\rho^u_{0,i}$ at time $T_0+(n+i)\tau$ for any $1\le i\le 2n$. v receives a request $\rho^v_{0,i}$ at time $T_0+i\tau$ for any $1\le i\le 2n$. w receives a request $\rho^w_{0,-1}$ at time 0 and a request $\rho^w_{0,n+i}$ at time $T_0+i\tau$ for any $1\le i\le n$. Let $T_1=T_0+(2n+1)\tau, T_j=T_{j-1}+3n\tau$ for any $2\le j\le m$.

For any $1\le j\le m$, the part R_j has $6n$ requests as follows: $\rho^u_{j,i}$ arrives at u at time $T_j+(2n+i-1)\tau$, $\rho^v_{j,i}$ arrives at v at time $T_j+(n+i-1)\tau$, and $\rho^w_{j,i}$ arrives at w at time $T_j+(i-1)\tau$, for every $1\le i\le 2n$.

Actually, we can very slightly perturb the arrival time of some requests so that Strategy III results in exactly the following external matches: $\langle \rho^u_{0,-1}, \rho^w_{0,-1} \rangle$, $\langle \rho^u_{0,0}, \rho^v_{0,n} \rangle$, $\langle \rho^u_{j,n}, \rho^w_{j,2n} \rangle$ for $1 \le j \le m$, $\langle \rho^u_{i,2n}, \rho^v_{i+1,n} \rangle$ and $\langle \rho^v_{i,2n}, \rho^w_{i+1,n} \rangle$ for $1 \le i < m$, and $\langle \rho^u_{m,2n}, \rho^v_{m,2n} \rangle$, as illustrated in Figure 3(a). The cost of Strategy III is at least $3m(\delta + \theta)$. On the other hand, consider the offline solution SOL which has no external matches, as indicated in Figure 3(b). It has cost at most $2f(T_0 + \tau) + \frac{6mn+5n-1}{2} f(\tau)$. When τ approaches zero and m approaches infinity, we have $3m(\delta + \theta) \gg 2f(T_0 + \tau) + \frac{6mn+5n-1}{2} f(\tau)$, implying that Strategy III is not competitive.

Let's look closer at the example. Consider an arbitrary (except the first) external match $\langle \rho, \rho' \rangle$ of Strategy III. It is of misaligned-aligned pattern in the sense that $\ell(\rho)$ and $\ell(\rho')$ have opposite alignment status when the match occurs. Suppose $\ell(\rho)$ is misaligned. Then it has accumulated high cost, mainly due to the long delay of ρ. On the contrary, SOL has accumulated little cost at $\ell(\rho)$, because SOL has no pending request there while ρ is pending. Hence, a match of misaligned-aligned pattern can significantly enlarge the gap between online/offline costs. To be worse, such a match does not change the number of aligned/misaligned points, making it possible that this pattern appears again and again, enlarging the gap infinitely. As a result, we establish a set which consists of points that are likely to be misaligned, and prioritize matching those requests that are located outside the set. The algorithm is described in detail as follows.

4.2 Algorithm Description

Our algorithm maintains a subset $\Psi \subseteq V$ and a counter $z_v \in \mathbb{R}^+$, which is initially set to 0, for every point $v \in V$. The algorithm proceeds round by round, and Ψ is reset to be the empty set $\emptyset$ at the beginning of each round. The first round begins when the algorithm starts. Let $k = |V|$. Whenever $2k$ external matches are output, the present round ends immediately and the next one begins. At any time t, the following operations are performed exhaustively, i.e., until there is no possible matching according to the following rules.

1. Every z_v increases at rate $f'(t - t_0)$ if there is an active request ρ at v with $t(\rho) = t_0$.
2. Match any pair of active requests ρ and ρ' if $\ell(\rho) = \ell(\rho')$.
3. For any pair of active requests ρ, ρ' with $u \triangleq \ell(\rho) \neq v \triangleq \ell(\rho')$, match them and reset $z_u = z_v = 0$ if there is $x \in \{u, v\}$ satisfying
   a. $z_x \ge 2\delta$, or
   b. $\delta \le z_x < 2\delta$ and $\{u, v\} \bigcap \Psi = \emptyset$.

   Arbitrarily choose such an $x \in \{u, v\}$, and we say that x initiates this match. Reset Ψ to be $(\Psi \setminus \{u, v\}) \bigcup \{x\}$ if either $u \notin \Psi$ or $v \notin \Psi$.

Priority rule: in applying Operation 3, the requests located outside Ψ are prioritized.

4.3 Competitive Analysis

Throughout this subsection, arbitrarily fix a time cost function $f(t) = t^\alpha$ with $\alpha > 1$, a uniform metric space $\mathcal{S} = (V, \delta)$ of k points, and an arbitrary online input instance R over $\mathcal{S}$. For ease of presentation, we assume that the arrival times of the requests are pairwise different. This assumption does not lose generality since the arrival times can be arbitrarily perturbed and timing in practice is up to errors. Let $\mathcal{A}$ stand for our algorithm and $\mathcal{A}^*$ for an optimum offline algorithm solving f-MPMD. We start competitive analysis by introducing notation.

4.3.1 Notations

For any request $\rho \in R$ and subset $I \subseteq \mathbb{R}^+$ of time, the time cost of $\mathcal{A}^*$ incurred by ρ during I is defined to be

$$C_{time}(\rho, I, \mathcal{A}^*) = \int_{(t(\rho), T^*(\rho)] \bigcap I} f'(t - t(\rho))dt,$$

where $T^*(\rho)$ is the time when ρ gets matched by $\mathcal{A}^*$. For any $v \in V$, define

$$C_{time}(v, I, \mathcal{A}^*) = \sum_{\rho \in R, \ell(\rho) = v} C_{time}(\rho, I, \mathcal{A}^*).$$

Let $C_{space}(v, I, \mathcal{A}^*)$ be $\frac{\delta}{2}$ times the number of requests at v that are externally matched by $\mathcal{A}^*$ during I.

Define $\Gamma = \{t \in \mathbb{R}^+ :$ at time $t, \mathcal{A}$ has a pending request ρ with $z_{\ell(\rho)} > 2\delta\}$. We will analyze time cost of $\mathcal{A}^*$ inside and outside Γ separately.

Our algorithm $\mathcal{A}$ runs round by round. Specifically, the *round* starting at time t_0 and ending at time t_1 is referred to as the time period $(t_0, t_1]$. Let Π be the set of rounds of $\mathcal{A}$.

For any $\pi \in \Pi$, define $round\_cost_{time}(\pi, \mathcal{A}^*) = \sum_{v \in V} C_{time}(v, \pi \setminus \Gamma, \mathcal{A}^*)$ which stands for the time cost of $\mathcal{A}^*$ during $\pi \setminus \Gamma$, and $round\_cost_{space}(\pi, \mathcal{A}^*) = \sum_{v \in V} C_{space}(v, \pi, \mathcal{A}^*)$ which is the space cost of $\mathcal{A}^*$ during π.

For any $v \in V$, we divide time into *phases* based on $\mathcal{A}$'s behavior as follows. The first phase begins at time $t = 0$. Whenever an external match involving v occurs, the current phase of v ends and the next phase of v begins. Specifically, the phase of v starting at time t_0 and ending at time t_1 is referred to as the period $(t_0, t_1]$ spent by v. For any $v \in V$, let Φ_v be the set of phases of v, and $\Phi = \bigcup_{v \in V} \Phi_v$. For any $\phi \in \Phi_v$, define the value of ϕ, denoted by $\sigma(\phi)$, to be the value of z_v at the end of ϕ. For an external match $\mathfrak{m}$ of $\mathcal{A}$ initiated by v, the phase of v ending with $\mathfrak{m}$ is called the phase of $\mathfrak{m}$, denoted by $\phi_{\mathfrak{m}}$. For any round $\pi \in \Pi$, let Φ_π be the set of phases ending in π. For any round $\pi \in \Pi$, define $phase\_cost_{time}(\pi, \mathcal{A}^*) = \sum_{v \in V} \sum_{\phi \in \Phi_\pi \bigcap \Phi_v} C_{time}(v, \phi \setminus \Gamma, \mathcal{A}^*)$, and $phase\_cost_{space}(\pi, \mathcal{A}^*) = \sum_{v \in V} \sum_{\phi \in \Phi_\pi \bigcap \Phi_v} C_{space}(v, \phi, \mathcal{A}^*)$.

We say that a phase of v is *good*, if the alignment status of v does not change during the phase. Furthermore, a round π is *good* if all the phases in Φ_π are good. A phase or a round is said to be bad if it is not good.

A phase is called *complete* if it ends with an external match of $\mathcal{A}$, while a round is *complete* if $\mathcal{A}$ outputs $2k$ external matches during it. Obviously, any round other than the final one is complete.

4.3.2 Competitive Ratio of Our Algorithm

Basically, we show that in every round, the incremental cost of $\mathcal{A}$ and that of $\mathcal{A}^*$ do not differ too much. This is reduced to two tasks. First, if all the counters are always small (say, no more than 4δ), the incremental cost of $\mathcal{A}$ in every round is $O(kd)$, so it suffices to show that the cost of $\mathcal{A}^*$ increases by $\Omega(d)$. This is the main task of this subsection and presented in Lemma 8. Second, to deal with the case that some counter z_v is large, we have to show that the accumulated cost of $\mathcal{A}^*$ in the phase increases nearly proportionately with z_v, as claimed in Lemma 9.

The following is a key lemma, stating that in every good complete round of $\mathcal{A}$, the cost of the optimum offline algorithm $\mathcal{A}^*$ is not small.

▶ **Lemma 7.** *In every good complete round π, we have either $round\_cost_{time}(\pi, \mathcal{A}^*) \geq f(f^{-1}(2\delta) - f^{-1}(\delta))$, or $round\_cost_{space}(\pi, \mathcal{A}^*) \geq \delta$, or $phase\_cost_{time}(\pi, \mathcal{A}^*) \geq \delta$.*

Up to now, we have focused on good rounds. The next lemma indicates that the cost of $\mathcal{A}^*$ in bad rounds can be *ignored* in some sense.

▶ **Lemma 8.** *The number of bad rounds of $\mathcal{A}$ is at most twice the number of external matches of $\mathcal{A}^*$.*

For any phase $\phi \in \Phi$, define its truncated value to be

$$\sigma'(\phi) = \begin{cases} 0 & \text{if } \sigma(\phi) \leq 2\delta \\ f(f^{-1}(\sigma(\phi)) - f^{-1}(2\delta)) & \text{otherwise} \end{cases}.$$

We will use truncated phase values to give a lower bound of the time cost of $\mathcal{A}^*$.

▶ **Lemma 9.** $\text{cost}^t_{\mathcal{A}^*}(R) \geq \sum_{\pi \in \Pi} phase\_cost_{time}(\pi, \mathcal{A}^*) + \sum_{\phi \in \Phi} \sigma'(\phi)$.

The following technical lemmas will be needed.

▶ **Lemma 10.** *For any $c_1, \cdots, c_n \geq c_0 > c > 0$ and $\alpha > 1$, we have*

$$\frac{\sum_{j=1}^n (c_j - c)}{\sum_{j=1}^n (\sqrt[\alpha]{c_j} - \sqrt[\alpha]{c})^\alpha} \leq \frac{c_0 - c}{(\sqrt[\alpha]{c_0} - \sqrt[\alpha]{c})^\alpha}.$$

▶ **Lemma 11.** *If $\mathcal{A}$ has only one round on the instance R, $\text{cost}_{\mathcal{A}}(R)/\text{cost}_{\mathcal{A}^*}(R) = O(k)$.*

Now we are ready to prove the main result.

▶ **Theorem 1.** *For any $f(t) = t^\alpha$ with constant $\alpha > 1$, the competitive ratio of $\mathcal{A}$ for f-MPMD on k-point uniform metric space is $O(k)$.*

Proof. Suppose that $\mathcal{A}$ has m rounds on the online input instance R, namely $|\Pi| = m$. By Lemma 11, we assume that $m > 1$.

In every round, there are at most $2k$ external matches and each of them ends two complete phases. So, there are altogether at most $4km$ complete phases. Considering that there are totally at most k incomplete phases, $|\Phi| \leq (4m+1)k \leq 5mk$. Let $\Phi' = \{\phi \in \Phi : \sigma(\phi) \geq 4\delta\}$. It holds that $\text{cost}_{\mathcal{A}}(R) = \text{cost}^s_{\mathcal{A}}(R) + \text{cost}^t_{\mathcal{A}}(R) \leq 2km\delta + \sum_{\phi \in \Phi} \sigma(\phi) \leq 22km\delta + \sum_{\phi \in \Phi'} (\sigma(\phi) - 4\delta) \leq 22km\delta + \sum_{\phi \in \Phi'} (\sigma(\phi) - 2\delta)$.

On the other hand, as to the cost of $\mathcal{A}^*$, we have $\text{cost}_{\mathcal{A}^*}(R) = \text{cost}^s_{\mathcal{A}^*}(R) + \text{cost}^t_{\mathcal{A}^*}(R) \geq \text{cost}^s_{\mathcal{A}^*}(R) + \sum_{\pi \in \Pi} phase\_cost_{time}(\pi, \mathcal{A}^*) + \sum_{\phi \in \Phi} \sigma'(\phi)$ by Lemma 9. Trivially we also have $\text{cost}_{\mathcal{A}^*}(R) \geq \sum_{\pi \in \Pi} [round\_cost_{time}(\pi, \mathcal{A}^*) + round\_cost_{space}(\pi, \mathcal{A}^*)]$. Let Π' be the set of good complete rounds and $m' = |\Pi'|$. Let m'' be the number of bad rounds. An easy observation is that $m' + m'' \geq m - 1$. By Lemma 8, $\mathcal{A}^*$ has at least $\frac{m''}{2}$ external matches:

$$\begin{aligned} 2\text{cost}_{\mathcal{A}^*}(R) \quad \geq \quad & \text{cost}^s_{\mathcal{A}^*}(R) + \textstyle\sum_{\pi \in \Pi} phase\_cost_{time}(\pi, \mathcal{A}^*) + \sum_{\phi \in \Phi} \sigma'(\phi) \\ & + \textstyle\sum_{\pi \in \Pi} [round\_cost_{time}(\pi, \mathcal{A}^*) + round\_cost_{space}(\pi, \mathcal{A}^*)] \\ \geq \quad & \tfrac{m''}{2}\delta + \textstyle\sum_{\phi \in \Phi} \sigma'(\phi) + \sum_{\pi \in \Pi'} [phase\_cost_{time}(\pi, \mathcal{A}^*) \\ & + round\_cost_{time}(\pi, \mathcal{A}^*) + round\_cost_{space}(\pi, \mathcal{A}^*)] \\ \geq \quad & \tfrac{m''}{2}\delta + \textstyle\sum_{\phi \in \Phi} \sigma'(\phi) + f(f^{-1}(2\delta) - f^{-1}(\delta))m' \\ \geq \quad & \tfrac{m-1}{2}(\sqrt[\alpha]{2} - 1)^\alpha \delta + \textstyle\sum_{\phi \in \Phi'} \sigma'(\phi) \end{aligned}$$

where the third equality is due to Lemma 7.

Altogether, $\frac{\text{cost}_{\mathcal{A}}(R)}{\text{cost}_{\mathcal{A}^*}(R)} \leq \frac{22km\delta + \sum_{\phi \in \Phi'} (\sigma(\phi) - 2\delta)}{\frac{m-1}{4}(\sqrt[\alpha]{2} - 1)^\alpha \delta + \frac{1}{2} \sum_{\phi \in \Phi'} \sigma'(\phi)}$, which is $O(k)$ by Lemma 10. ◀

5 Lower Bound for Deterministic Algorithms

This section is devoted to showing that any deterministic algorithm for the convex-MPMD problem on k-point uniform metric space must have competitive ratio $\Omega(k)$, meaning that our algorithm is optimum, up to a constant factor. Let's begin with a convention of notation. Let $f : \mathbb{R}^+ \mapsto \mathbb{R}^+$ be a nondecreasing, unbounded, continuous function satisfying $f(0) = f'(0) = 0$. Let $\mathcal{S} = (V, \delta)$ be a uniform metric space with $V = \{v_0, v_1, ...v_k\}$. Suppose that $\mathcal{A}$ is an arbitrary deterministic online algorithm for the f-MPMD problem. Let $T \in \mathbb{R}^+$ be such that $f(T) = k\delta$. Arbitrarily choose a real number $\tau > 0$ such that $n = \frac{T}{\tau}$ is an even number.

We construct an instance R of online input to $\mathcal{A}$ and show that the competitive ratio of $\mathcal{A}$ is at least $\Omega(k)$. The instance R is determined in an online fashion: Roughly speaking, based on the up-to-now behavior of $\mathcal{A}$, we choose when and where to input next requests so as to force $\mathcal{A}$ to have many external matches.

Specifically, R is determined in m rounds, where m is an arbitrary positive integer. The first round begins at time $T_1 = 0$. Some requests arrive in the manner as described in the next four paragraphs. At arbitrary time T_2 after these requests are all matched, finish the first round and start the second round. Repeat this process until we have finished m rounds. All the requests form the instance R.

Now we describe the requests that arrive during the rth round, namely in the interval $[T_r, T_{r+1})$, for any $1 \leq r \leq m$. Basically, at v_0 there is just one request, denoted by ρ_{00}, which arrives at time T_r, while a request ρ_{ij} arrives at every point v_i at time $T_r + j\tau$, for any integers $1 \leq i \leq k$ and $j \geq 1$. We will iteratively specify when requests should stop arriving at the points other than v_0.

Define $G_0 = (V, \emptyset)$ to be the graph on V with no edges. Let $C_0 = \{v_0\}$. Starting with $h = 1$, iterate the following process until no more requests will arrive. At time $T_r + hT$, construct an undirected graph G_h on V. It has an edge between any pair of vertices $v_i \neq v_{i'}$ if and only if by time $T_r + hT$, $\mathcal{A}$ has matched one request at v_i and another at $v_{i'}$ both of which arrived during the period $[T_r, T_r + hT]$. Let C_h be the set of the vertices in the connected component of G_h containing v_0. We proceed case by case:

Case 1: $C_{h-1} \neq C_h = V$. Then no more requests except $\rho_{i,hn+1}$ will arrive, where i is arbitrarily chosen such that $v_i \in C_h \setminus C_{h-1}$. Denote this h by h_r.

Case 2: $C_{h-1} = C_h$. Then no more requests except $\rho_{i,hn+1}$ will arrive, where i is arbitrarily chosen such that $v_i \in V \setminus C_h$. Denote this h by h_r.

Case 3: otherwise. Then no more requests will arrive at any $v_i \in C_h$, while requests continue arriving at points in $V \setminus C_h$. Increase h by 1 and iterate.

Arbitrarily fix $1 \leq r \leq m$ in the rest of this section. Let R_r be the set of requests that arrive in the first r rounds, and N_r be the number of requests in $R_r \setminus R_{r-1}$, where $R_0 = \emptyset$. Let $R = R_m$. It is easy to see four facts:

Fact 1: $N_r \leq k^2n + 2$.

Fact 2: $R_r \setminus R_{r-1}$ has exactly one request at v_0, and has an odd number of requests at the point where the last request arrives, respectively.

Fact 3: $R_r \setminus R_{r-1}$ has an even number of requests at any other point.

Fact 4: No match occurs between requests of different rounds.

Some lemmas are needed for proving the main result.

▶ **Lemma 12.** $\text{cost}_{\mathcal{A}^*}(R_r) \leq (\delta + \frac{k^2n}{2}f(\tau) + f(\tau))r$.

▶ **Lemma 13.** $\text{cost}_{\mathcal{A}}(R_r) \geq k\delta r$.

▶ **Theorem 3.** *Suppose that the time cost function f is nondecreasing, unbounded, continuous and satisfies $f(0) = f'(0) = 0$. Then any deterministic algorithm for f-MPMD on k-point uniform metric space has competitive ratio $\Omega(k)$.*

Proof. Suppose there are $a = a(k, \delta)$ and $b = b(k, \delta)$ such that for any $m \geq 1$, $\text{cost}_{\mathcal{A}}(R) \leq a \cdot \text{cost}_{\mathcal{A}^*}(R) + b$. Fix k and δ. Dividing both sides of inequality by m and letting m approach infinity, by Lemmas 12 and 13, we get $f(n\tau) \leq (\delta + \frac{k^2 n}{2} f(\tau) + f(\tau))a$, which means that $a \geq \frac{f(n\tau)}{\delta + \frac{k^2 n}{2} f(\tau) + f(\tau)} = \frac{\frac{k\delta}{2} + \frac{1}{2} f(n\tau)}{\delta + \frac{k^2 n}{2} f(\tau) + f(\tau)}$.

Let τ approach zero. One has $\lim_{\tau \to 0} f(\tau) = 0$, and

$$\lim_{\tau \to 0} \frac{f(n\tau)}{k^2 n f(\tau)} = \lim_{\tau \to 0} \frac{1}{k^2} \frac{f(n\tau)}{n\tau} \frac{\tau}{f(\tau)} = \lim_{\tau \to 0} \frac{1}{k^2} \frac{f(T)}{T} \frac{\tau}{f(\tau)} = +\infty \quad \text{since } f'(0) = 0$$

This means $\lim_{\tau \to 0} k^2 n f(\tau) = 0$, since $f(n\tau) = k\delta$ is a constant when k and δ are fixed. As a result, $a = \lim_{\tau \to 0} a \geq \lim_{\tau \to 0} \frac{\frac{k\delta}{2} + \frac{1}{2} f(n\tau)}{\delta + \frac{k^2 n}{2} f(\tau) + f(\tau)} = \frac{k\delta}{\delta} = k$. ◀

6 Conclusion

We have designed an optimum deterministic online algorithm that solves f-MPMD for any monomial function $f(t) = t^\alpha$ with $\alpha > 1$. It is remarkable that the algorithm remains optimum if only $f : \mathbb{R}^+ \mapsto \mathbb{R}^+$ is an increasing and convex polynomial function with $f(0) = 0$. Actually, following Subsection 4.3.2, one can easily see that the competitive ratio is at most $\max \left\{ \frac{120k\delta}{f(f^{-1}(2\delta) - f^{-1}(\delta))}, \sup_{c \geq 4\delta} \frac{c - 2\delta}{f(f^{-1}(c) - f^{-1}(2\delta))} \right\}$, which is $O(k)$ by elementary calculus, when f is fixed.

An interesting future direction is to design a randomized algorithm for convex-MPMD. A randomized algorithm is usually more competitive than a deterministic one when considering oblivious adversaries. We conjecture that there is a randomized algorithm for convex-MPMD with competitive ratio $O(\log k)$ but no such algorithm with competitive ratio $O(1)$. If this turns out true, there is still a clear separation between linear-MPMD and convex-MPMD in the context of randomized algorithms.

In contrast to convex functions, concave functions may model the fact that in some applications the delay cost grows slower and slower, which encourages matching two new requests instead of matching old requests. It seems not difficult to design an algorithm with bounded competitive ratio for these concave cost functions, but to design a good one, i.e., with a very small competitive ratio, seems still challenging.

References

1 Gagan Aggarwal, Gagan Goel, Chinmay Karande, and Aranyak Mehta. Online Vertex-Weighted Bipartite Matching and Single-bid Budgeted Allocations. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, page 1253–1264, 2011.

2 Itai Ashlagi, Yossi Azar, Moses Charikar, Ashish Chiplunkar, Ofir Geri, Haim Kaplan, Rahul Makhijani, Yuyi Wang, and Roger Wattenhofer. Min-cost Bipartite Perfect Matching with Delays. In *20th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX), Berkeley, California, USA*, August 2017.

3 Yossi Azar, Ashish Chiplunkar, and Haim Kaplan. Polylogarithmic Bounds on the Competitiveness of Min-cost Perfect Matching with Delays. *arXiv*, 2016. `arXiv:1610.05155`.

4 Yossi Azar, Ashish Chiplunkar, and Haim Kaplan. Polylogarithmic Bounds on the Competitiveness of Min-cost Perfect Matching with Delays. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, page 1051–1061, 2017.

5 Yossi Azar, Arun Ganesh, Rong Ge, and Debmalya Panigrahi. Online service with delay. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 551–563. ACM, 2017.

6 Benjamin E. Birnbaum and Claire Mathieu. On-line bipartite matching made simple. *SIGACT News*, 39(1):80–87, 2008.

7 Nikhil R. Devanur, Kamal Jain, and Robert D. Kleinberg. Randomized Primal-Dual analysis of RANKING for Online BiPartite Matching. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, page 101–107, 2013.

8 Jack Edmonds. Maximum matching and a polyhedron with 0, 1-vertices. *Journal of Research of the National Bureau of Standards B*, 69:125–130, 1965.

9 Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

10 Yuval Emek, Shay Kutten, and Roger Wattenhofer. Online Matching: Haste makes Waste! In *48th Annual Symposium on Theory of Computing (STOC)*, June 2016.

11 Yuval Emek, Yaacov Shapiro, and Yuyi Wang. Minimum Cost Perfect Matching with Delays for Two Sources. In *10th International Conference on Algorithms and Complexity (CIAC), Athens, Greece*, May 2017.

12 Gagan Goel and Aranyak Mehta. Online budgeted matching in random input models with applications to Adwords. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, page 982–991, 2008.

13 Bala Kalyanasundaram and Kirk Pruhs. Online Weighted Matching. *J. Algorithms*, 14(3):478–488, 1993.

14 Anna R. Karlin, Mark S. Manasse, Lyle A. McGeoch, and Susan Owicki. Competitive randomized algorithms for nonuniform problems. *Algorithmica*, 11(6):542–571, 1994.

15 Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An Optimal Algorithm for On-line Bipartite Matching. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, page 352–358, 1990.

16 Samir Khuller, Stephen G. Mitchell, and Vijay V. Vazirani. On-Line Algorithms for Weighted Bipartite Matching and Stable Marriages. *Theor. Comput. Sci.*, 127(2):255–267, 1994.

17 Aranyak Mehta. Online Matching and Ad Allocation. *Foundations and Trends in Theoretical Computer Science*, 8(4):265–368, 2013.

18 Aranyak Mehta, Amin Saberi, Umesh V. Vazirani, and Vijay V. Vazirani. AdWords and Generalized On-line Matching. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, page 264–273, 2005.

19 Adam Meyerson, Akash Nanavati, and Laura J. Poplawski. Randomized online algorithms for minimum metric bipartite matching. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, 2006.

20 Shuichi Miyazaki. On the advice complexity of online bipartite matching and online stable marriage. *Inf. Process. Lett.*, 114(12):714–717, 2014.

21 Joseph Naor and David Wajc. Near-Optimum Online Ad Allocation for Targeted Advertising. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC*, page 131–148, 2015.