Online Matching with Convex Delay Costs

Xingwu Liu · Zhida Pan⋆ · Yuyi Wang · Roger Wattenhofer

Received: date / Accepted: date

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, or $O(k\Delta)$ when the metric space has aspect ratio $\Delta$. Moreover, we prove lower bounds for the competitive ratio of deterministic and randomized algorithms, indicating that our deterministic algorithm is optimal. This optimality uncover a substantial difference between

⋆ Corresponding author

A short version appears in the proceedings of ISAAC 2018 under the title Impatient Online Matching. This full version includes all proofs and new lower bounds.

X. Liu
SKL Computer Architecture, ICT, CAS
University of Chinese Academy of Sciences, Beijing, China
E-mail: liuxingwu@ict.ac.cn

Z. Pan
X-Order Lab, Shanghai, China
E-mail: zhidapan@gmail.com

Y. Wang
X-Order Lab, Shanghai, China
E-mail: yywang@xorder.ai

R. Wattenhofer
ETH Zürich
E-mail: wattenhofer@ethz.ch
convex-MPMD and linear-MPMD, since linear-MPMD allows a deterministic algorithm with constant competitive ratio on any uniform metric space.

**Keywords** online algorithm, online matching, convex function, competitive analysis, lower bound

1 Introduction

Online matching has been studied frantically in the last years. Emek et al. [17] 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 involves two kinds of costs or penalties. One is for quality: Requests that are paired up should in some sense be as compatible as possible, so incompatibility leads to penalties. 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 [17,18,4,3,2]. However, existing work in this line only studied linear time cost functions, meaning that the time 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 hungry customer at a restaurant: 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.

[1] https://www.hopkinsmedicine.org/transplant/programs/kidney/incompatible/paired_kidney_exchange.html
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 [4,18,2]). 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 should 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.

Another element of the MPMD problem is a metric space which intuitively captures how compatible the requests are. We focus on uniform metric spaces, i.e., the points in the spaces are equal-distant. This is due to the fundamental role of uniform metrics in the MPMD problem. A common technique in MPMD literature is to embed a general metric into a probabilistic hierarchical separated tree (HST). Since every level of an HST can be considered as a uniform metric, the task is essentially reduced to algorithm design on uniform metrics (or aspect-ratio-bounded metrics which can also be handled by our results). Uniform metrics are known to be tricky, e.g., Emek et al. [18] study linear-MPMD with only two points. Uniform metrics also play an important role in other online problems [21]. For example, the $k$-server problem restricted to uniform metrics is the well-known paging problem.

In this paper, we devise a novel algorithm $A$ for the convex-MPMD problem which is deterministic and solves the problem optimally. More importantly, our results uncover 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 $f$:

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

Theorem 1 can be easily generalized. On an arbitrary metric space with aspect ratio $\Delta$, the competitive ratio of $A$ is $O(k\Delta)$.

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 $k$-point uniform metric space and reaches an $O(1)$ competitive ratio.

However, for a large family of functions $f : \mathbb{R}^+ \rightarrow \mathbb{R}^+$, 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 $\Omega(k)$.

We also get a lower bound for randomized algorithms.

Theorem 4 Suppose that the time cost function $f$ is nondecreasing, unbounded, continuous and satisfies $f(0) = f'(0) = 0$. Then any randomized algorithm for $f$-MPMD on $k$-point uniform metric space has competitive ratio $\Omega(\ln k)$ against oblivious adversaries.

Numerous natural convex functions over the domain of nonnegative real numbers satisfy the conditions of Theorems 3 and 4. Examples include monomials $f(t) = t^\alpha$ 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 is closed under multiplication and linear combination where the coefficients are positive. Hence, the lower bounds are of general significance.

2 Related Work

Matching has become one of the most extensively studied problems in graph theory and computer science since the seminal work of Edmonds [16,15]. Karp et al. [23] studied the matching problem in the context of online computation which inspired a number of different versions of online matching, e.g., [20, 21, 25, 27, 11, 19, 12, 25, 28, 29]. 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 [17], which also introduced the idea that requests are allowed to be matched up to delays that need to be paid as well, so the problem is called Min-cost Perfect Matching with Delays (MPMD). The authors 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 $\Delta$ is the aspect ratio of the metric space. Later, Azar et al. [4] proposed an almost-deterministic algorithm with competitive ratio $O(\log k)$. Ashlagi et al. [2] extended these algorithms to bipartite matching with delays (MBPMD), and achieved a competitive ratio of $O(\log k)$, and also proved a lower bound of $\Omega(\log k / \log \log k)$ for MPMD and a lower bound of $\Omega(\sqrt{\log k / \log \log k})$ for MBPMD. Bienkowski et al. [10] proposed the first deterministic algorithm for MBPMD problems and obtained a competitive ratio of $O(m^{2/46})$ where $m$ is the total number of arriving requests, and this result was improved to $O(m)$ in [9]. Azar et al. [6] proposed an $O(m^{\log(2/3+\epsilon)})$-competitive deterministic algorithm, based on these two previous results, for both MPMD and MBPMD problems. In contrast to our work, all these papers assume that the time cost of a request is linear in its waiting time.

The idea of delaying decisions has been around for a long time in the form of rent-or-buy problems (e.g., ski rental [22,21] and TCP acknowledgment
Online Matching with Convex Delay Costs

[13,14], but [17] is the first to show how to use delays in the context of combinatorial problems such as matching. After this, there have been a number of generalizations of classical online problems, which allow decisions with delays. To name a few, Azar et al. [7] 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 problem essentially different, and there is a universal way to deal with all these functions, in contrast to the online matching problems we consider in the present paper. Other problems with delays include the online bin-packing [5] and the online facility allocation [8].

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 $S = (V, \mu)$ is a set $V$, whose members are called points, equipped with a distance function $\mu : V^2 \rightarrow \mathbb{R}^+$ 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 $x, y \in V$;

Triangle inequality: $\mu(x, y) + \mu(y, z) \geq \mu(x, z)$ for any $x, y, z \in V$.

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

For any finite metric space $S = (V, \mu)$, an online input instance over $S$ is a sequence $R$ of requests. Each request $\rho \in R$ is characterized by its location $\ell(\rho) \in V$ and arrival time $t(\rho) \in \mathbb{R}^+$. Assume that $|R|$ is an even number that is not pre-determined. The goal is to pair up the requests over time without revocation, resulting in a perfect matching of the requests.

Specifically, suppose an algorithm $A$ pairs up $\rho, \rho' \in R$ 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 $A$ on input $R$, denoted by $\text{cost}_s^A(R)$, is the total space cost caused by all the matched pairs, and the time cost $\text{cost}_t^A(R)$ is defined likewise. The objective of the $f$-MPMD is to find an online algorithm $A$ such that $\text{cost}_A(R) = \text{cost}_s^A(R) + \text{cost}_t^A(R)$ is minimized for all $R$.

As usual, the online algorithm $A$ is evaluated by competitive analysis. Let $A^*$ be an optimum offline algorithm\footnote{An offline algorithm knows the whole input instance at the beginning and outputs any pair $\rho, \rho' \in R$ at time $\max\{t(\rho), t(\rho')\}$.} For any finite metric space $S$, if there
are \( a, b \in \mathbb{R}^+ \) such that \( \text{cost}_A(R) \leq a \cdot \text{cost}_A^*(R) + b \) for any online input instance \( R \) over \( S \), then \( A \) is said to be \( a \)-competitive on \( S \). The minimum such \( a \) is called the competitive ratio of \( A \) on \( S \). Note that both \( a \) and \( b \) can depend on \( S \). If no such \( a, b \) exist for some \( S \), we say that \( A \) is not competitive.

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

### 3.2 Notation and Terminology

Any requests \( \rho, \rho' \) that are paired up in the perfect matching is called a match between \( \rho \) and \( \rho' \) 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 \( \rho \), let \( T(\rho) \) be the time when \( \rho \) gets matched; \( \rho \) is said to be pending at any time \( t \in (\ell(\rho), T(\rho)) \) and active at any time \( t \in [\ell(\rho), T(\rho)] \). At any moment \( t \), a point \( v \in V \) is called aligned if the number of pending requests at \( v \) under \( A \) and that under \( A^* \) have the same parity, and misaligned otherwise.

The derivative of any differentiable function \( f : \mathbb{R}^+ \rightarrow \mathbb{R}^+ \) 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 a period \( \theta \)). 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 1.

**Example 1** For any positive integer \( n \) and small real number \( \epsilon > 0 \), construct an online instance as follows. Assume that the metric space consists of two points \( u \) and \( v \) with distance \( \delta \). A request \( \rho_{2i} \) arrives at \( u \) at time \( i \cdot \theta \) for any \( 0 \leq i \leq n \), while a request \( \rho_{2i-1} \) arrives at \( u \) at time \( i \cdot \theta - \epsilon \) for any \( 1 \leq i \leq n \). Point \( v \) gets a request \( \rho' \) at time 0. By Strategy I, as in Figure 1(a), each \( \rho_{2i} \) is matched with \( \rho_{2i+1} \) for any \( 0 \leq i < n \), and \( \rho' \) and \( \rho_{2n} \) are paired up, causing cost \( n \cdot f(\theta - \epsilon) + f(n\theta) + \delta \). 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 \( \epsilon \) approaches 0, \( n \cdot f(\theta - \epsilon) + f(n\theta) + \delta \gg \delta + n \cdot f(\epsilon) \), meaning that Strategy I is not competitive.

Can we improve Strategy I so as to beat Example 1? A plausible way is as follows. At each point \( v \) in the metric space, accumulate from scratch the time costs of all the requests which have arrived at \( v \) since the last external match involving \( v \). An external match between point \( u, v \) is enable only if both have accumulated enough time costs (say, as large as \( \theta \)). Though beating Example
Fig. 1 The input instance of Example 1. 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.

Example 2 Arbitrarily fix an even integer \( n > 0 \) and a small real number \( \epsilon > 0 \). Let \( \tau \in \mathbb{R}^+ \) be such that \( \theta - \epsilon < \frac{n}{2} f(\tau) < \theta \). Again, assume that the metric space consists of two points \( u \) and \( v \) with distance \( \delta \). Suppose that a request \( \rho' \) arrives at \( v \) at time 0, while a request \( \rho_i \) arrives at \( u \) at time \( i\tau \) for any \( 0 \leq i < n \). 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 \( \frac{n}{2} f(\tau) + \delta \). Thus the cost of \( A^* \) is at most \( \frac{n}{2} f(\tau) + \delta \). When \( n \) approaches infinity and \( \epsilon \) 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.

The failure in Example 2 is rooted at the double-enabling mechanism which enables an external match only if the cumulated costs at both points reach some threshold. Hence, we replace that mechanism with a looser one: an external match is enabled if one of the two points has high accumulated cost (say, as high as \( \theta \)). The new algorithm (called Strategy III) defeats both Examples 1 and 2, but the following example shows that it remains not competitive for any monomial time cost function \( f(t) = t^\alpha, \alpha > 1 \).

Example 3 Choose \( \tau \in \mathbb{R}^+ \) and odd integer \( n > 0 \) such that \( f(n\tau) = \theta \). Arbitrarily fix real number \( T_0 > n\tau \). Consider a \( \delta \)-uniform metric space with points \( u, v, w \). Let \( m > 0 \) be an arbitrary integer. Construct an online input
Fig. 3 The input instance of Example 3. 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) The matching produced by Strategy III. (b) An offline solution. Label $T_i$ is (a)

instance $R$ which is the catenation 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 $\rho^u_{0,0}$ at time 0, $\rho^u_{0,i}$ at time $T_0 + (n + i)\tau$ for any $1 \leq i \leq 2n$. $v$ receives a request $\rho^v_{0,i}$ at time $T_0 + i\tau$ for any $1 \leq i \leq 2n$. $w$ receives a request $\rho^w_{0,i}$ at time 0 and a request $\rho^w_{0,n+i}$ at time $T_0 + i\tau$ for any $1 \leq i \leq n$. Let $T_1 = T_0 + (2n + 1)\tau, T_j = T_{j-1} + 3n\tau$ for any $2 \leq j \leq m$.

For any $1 \leq j \leq m$, 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 $1 \leq i \leq 2n$.

Actually, we can slightly perturb the arrival time of some requests so that Strategy III results in exactly the following external matches:

$\langle \rho^u_{0,0}, \rho^u_{0,0} \rangle, \langle \rho^u_{0,n}, \rho^u_{0,0} \rangle, \langle \rho^u_{1,n}, \rho^v_{1,0} \rangle, \langle \rho^v_{1,2n}, \rho^w_{1,1,n} \rangle, \langle \rho^w_{2,2n}, \rho^w_{1,2n} \rangle, \langle \rho^w_{m,2n}, \rho^w_{m,2n} \rangle$, 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 $2f(T_0 + \tau) + \frac{3mn + 5n - 1}{2}f(\tau)$. When $\tau$ approaches zero and $m$ approaches infinity, we have $3m(\delta + \theta) \gg 2f(T_0 + \tau) + \frac{3mn + 5n - 1}{2}f(\tau)$, implying that Strategy III is not competitive.

Let’s look closer at the example. Consider an arbitrary (except the first) external match $\langle p, p' \rangle$ of Strategy III. It is of misaligned-aligned pattern in the sense that $\ell(p)$ and $\ell(p')$ have opposite alignment status when the match occurs. Suppose $\ell(p)$ is misaligned. Then it has accumulated high cost, mainly due to the long delay of $p$. On the contrary, SOL has accumulated little cost at $\ell(p)$, because SOL has no pending request there while $p$ 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 $\Psi \subseteq V$ to store points tend to be misaligned. For every point $v \in V$, we use a timer $z_v \in \mathbb{R}^+$ to accumulate the time cost at $v$. The timer $z_v$ is initialized to be 0, and increases at rate $f'(t - t_0)$ at time $t$ when there is an active request $\rho$ with $l(\rho) = v$ and $t(\rho) = t_0$. The algorithm proceeds round by round, and $\Psi$ 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 produced, 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 rules.

1. Pair up active requests $\rho$ and $\rho'$ if $\ell(\rho) = \ell(\rho')$.
2. For active requests $\rho, \rho'$ with $u \triangleq \ell(\rho) \neq v \triangleq \ell(\rho')$, if there is $x \in \{u, v\}$ satisfying either (a) $\delta \leq z_x < 2\delta$ and $\{u, v\} \cap \Psi = \emptyset$ or (b) $z_x \geq 2\delta$, perform the three actions:
   - Pair up $\rho$ with $\rho'$.
   - Reset $z_u = z_v = 0$.
   - Set $\Psi$ to be $(\Psi \setminus \{u, v\}) \cup \{x\}$ if either $u \notin \Psi$ or $v \notin \Psi$.

In this case, we say that $x$ initiates the match $\langle \rho, \rho' \rangle$.

Priority rule: in applying Operation 2, the requests located outside $\Psi$ 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 $S = (V, \delta)$ of $k$ points, and an arbitrary online input instance $R$ over $S$. 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 might be slightly perturbed and timing in practice is up to errors. Let $A$ stand for our algorithm and $A^*$ for an optimum offline algorithm solving $f$-MPMD. We start competitive analysis by introducing notation.
4.3.1 Notation

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

\[
C_{\text{time}}(\rho, I, A^*) = \int_{(t(\rho), T^*(\rho)] \cap I} f'(t - t(\rho)) dt,
\]

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

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

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

Define

\[
\Gamma = \{ t \in \mathbb{R}^+ : \text{at time } t, A \text{ has a pending request } \rho \text{ with } z_\ell(\rho) > 2\delta \}.
\]

We will analyse time cost of \( A^* \) inside and outside \( \Gamma \) separately.

Our algorithm \( A \) runs round by round as described in Section 4.2. Specifically, the round starting at time \( t_0 \) and ending at time \( t_1 \) corresponds to the time period \( (t_0, t_1] \). Let \( \Pi \) be the set of rounds of \( A \).

For any \( \pi \in \Pi \), define

\[
\text{round cost time}(\pi, A^*) = \sum_{v \in V} C_{\text{time}}(v, \pi \setminus \Gamma, A^*)
\]

which stands for the time cost of \( A^* \) during \( \pi \setminus \Gamma \), and

\[
\text{round cost space}(\pi, A^*) = \sum_{v \in V} C_{\text{space}}(v, \pi, A^*)
\]

which is the space cost of \( A^* \) during \( \pi \).

Arbitrary fix a round \( \pi \) and an external match \( m = (\rho, \rho) \) initiated by \( l(\rho) \in V \). Define the phase of \( m \), denoted by \( \phi(m) \), to be the period \( (t, T(\rho)] \), where \( t < T(\rho) \) is the time closest to \( T(\rho) \) such that \( z_\ell(\rho) = 0 \) at time \( t \).

Define the value of \( m \), denoted by \( \sigma(m) \), to be the value of \( z_\ell(\rho) \) at time \( T(\rho) \).

If \( T(\rho) \in \pi \), we say that \( m \) is a match of round \( \pi \). If \( \phi(m) \subseteq \pi \), \( m \) is called a native match of \( \pi \). \( m \) is called a good match, if the alignment status (namely, being aligned or misaligned) of \( l(\rho) \) never change throughout \( \phi(m) \). If all the external matches of \( \pi \) are good, \( \pi \) is called a good round. Round \( \pi \) is called complete if it has \( 2k \) external matches.
4.3.2 Competitive Ratio of Our Algorithm

Basically, we show that in every round, the increment of the cost of $A$ does not differ too much from that of $A^*$. This is reduced into two tasks. First, if the timers are always small (say, no more than $4\delta$), the cost of $A$ increases by $O(kd)$ in every round, while that of $A^*$ increases by $O(d)$. This is the main task of this subsection and presented in Lemma 8. Second, in case that some timer $z_v$ gets too large, we show that the cost of $A^*$ increases simultaneously and almost proportionately, as claimed in Lemma 9.

Let’s begin with some technical lemmas that will be frequently used.

**Lemma 1** Let $h : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ be an invertible increasing convex function. The inequality $h(h^{-1}(\xi) - h^{-1}(\eta)) + \zeta \geq h(h^{-1}(\xi + \zeta) - h^{-1}(\eta))$ holds for any $\xi, \eta, \zeta \in \mathbb{R}^+$ with $\xi \geq \eta$.

**Proof** Let $x = h^{-1}(\xi), y = h^{-1}(\eta), z = h^{-1}(\xi + \zeta)$. Note that $y \leq x \leq z$. Then $h(z) - h(z - y) = \int_{[z-y,z]} h'(t)dt = \int_{[x-y,x]} h'(t+z-x)dt$. By convexity of $h$, $h'$ is increasing, implying that $h(z) - h(z - y) \geq \int_{[x-y,x]} h'(t)dt = h(x) - h(x - y)$. As a result, $h(x - y) + h(z) - h(x) \geq h(z - y)$, which is exactly the desired inequality. □

**Lemma 2** Suppose that $\rho_1, \cdots, \rho_n \in R$ with $T(\rho_i) < t(\rho_{i+1})$ for any $1 \leq i < n$ are successive pending requests at $v \in V$. Let $\gamma$ and $\lambda$ be the value of $z_v$ at some time $t_i \in (t(\rho_i), T(\rho_i)]$ and $T_n \in (t(\rho_n), T(\rho_n)]$, respectively. Let $t_i = t(\rho_i)$ for $1 \leq i \leq n$ and $T_j = T(\rho_j)$ for $1 \leq j < n$. Then $\sum_{i=1}^n f(T_i - t_i) \geq f(f^{-1}(\lambda) - f^{-1}(\gamma))$.

**Proof** For any $1 \leq i \leq n$, let $c_i$ be the increment of $z_v$ during $I_i = (t_i, T_i]$, i.e. $c_i \triangleq \int_{t_i}^{T_i} f'(t-t(\rho_i))dt$. Then we have $\lambda - \gamma \leq \sum_{i=1}^n c_i$.

When $i > 1$, $c_i = f(T_i - t_i)$ because $t(\rho_i) = t_i$.

Now it comes to $i = 1$. Since $z_v = \gamma$ at time $t_1$, $f(t_1 - t(\rho_1)) = \int_{(t(\rho_1), t_1]} f'(t-t(\rho_1))dt \leq \gamma$. Because $c_1 = \int_{(t_1, T_1]} f'(t-t(\rho_1))dt = f(T_1 - t(\rho_1)) - f(t_1 - t(\rho_1))$, it holds that $T_1 - t_1 = f^{-1}(c_1 + x) - f^{-1}(x)$ where $x = f(t_1 - t(\rho_1))$. By convexity of $f$ and $x \leq \gamma$, we have $f^{-1}(c_1 + x) - f^{-1}(x) \geq f^{-1}(c_1 + \gamma) - f^{-1}(\gamma)$. Then,

$$
\sum_{i=1}^n f(T_i - t_i) \geq f(f^{-1}(c_1 + \gamma) - f^{-1}(\gamma)) + c_2 + \cdots + c_n
$$

$$
\geq f(f^{-1}(\gamma + c_1 + c_2 + \cdots + c_n) - f^{-1}(\gamma))
$$

$$
\geq f(f^{-1}(\lambda) - f^{-1}(\gamma)),
$$

where the second inequality follows from Lemma 1. □

**Lemma 3** If an aligned point $v \in V$ initiates a good native match $m$ of round $\pi$, then $\text{round}_{\text{cost}}\text{time}(\pi, A^*) \geq \min\{\sigma(m), 2\delta\}$.
Lemma 3, the assertion again follows.

Proof Let \( \rho_1, \cdots, \rho_n \in R \) with \( T(\rho_i) < t(\rho_{i+1}) \) for any \( 1 \leq i < n \) be the requests at \( v \) that are successively pending during \( \phi(m) \). We proceed case by case.

Case 1: \( \sigma(m) \leq 2\delta \). Since \( v \) is aligned throughout \( \phi(m) \), \( A^* \) has requests \( \rho_1, \cdots, \rho_n \in R \) at \( v \) with \( t(\rho_i) \leq t(\rho_i') \) and \( T(\rho_i') \geq T(\rho_i) \) for any \( 1 \leq i \leq n \). Then by Lemma 2, \( \text{round}_{\text{cost}}(\pi, A^*) \geq \sum_{i \geq 1} f(T(\rho_i) - t(\rho_i)) = \sigma(m) \).

Case 2: \( \sigma(m) > 2\delta \). Suppose \( z_v \) reaches \( 2\delta \) exactly at time \( \hat{T} \) when \( \rho_m \) is active. Adopting the notation in Case 1, we have \( \text{round}_{\text{cost}}(\pi, A^*) \geq \sum_{i \geq 1} f(T(\rho_i) - t(\rho_i)) + (\hat{T} - t(\rho_m)) = 2\delta \).

Lemma 4 If a misaligned point appears in \( \Psi \) in a good round \( \pi \), then we have \( \text{round}_{\text{cost}}(\pi, A^*) \geq \delta \) or \( \text{round}_{\text{cost}}(\pi, A^*) \geq \delta \).

Proof Let \( v \) be the first misaligned point in \( \Psi \) during the round \( \pi \), namely, during \( \pi \), any point in \( \Psi \) is aligned before \( v \in \Psi \) gets misaligned. Then we proceed case by case.

Case 1: \( v \) is misaligned when it goes into \( \Psi \). By the rule of updating \( \Psi \), \( v \) goes into \( \Psi \) due to an external match \( m \) initiated by \( v \). Hence, before \( m \) occurs, \( v \) is aligned. Then \( \text{round}_{\text{cost}}(\pi, A^*) \geq \delta \) by Lemma 2.

Case 2: \( v \) is aligned when it goes into \( \Psi \), but gets misaligned due to an external match of \( A^* \). Then \( \text{round}_{\text{cost}}(\pi, A^*) \geq \delta \).

Case 3: \( v \) is aligned when it goes into \( \Psi \), but gets misaligned due to an external match \( m \) of \( A \). Then before \( m \) occurs, \( v \) is aligned. Again by the rule of updating \( \Psi \), \( m \) must be initiated either by \( v \) or by another point \( u \in \Psi \). Anyway, the initiating point must be aligned before \( m \) occurs, since \( v \) is the first misaligned point in \( \Psi \) during this round. As a result, \( \text{round}_{\text{cost}}(\pi, A^*) \geq \delta \) by Lemma 3.

Roughly speaking, the next lemma claims that under some condition, even if \( A \) pairs up one request located inside \( \Psi \) with another outside \( \Psi \), the cost of \( A^* \) must increase substantially.

Lemma 5 Suppose in a good round \( \pi \), \( v \notin \Psi \) initiates a native match \( m \) between a request at \( v \) and another at \( v' \in \Psi \). Then one of the following two claims is true:

\[
\begin{align*}
- \text{round}_{\text{cost}}(\pi, A^*) \geq f(f^{-1}(2\delta) - f^{-1}(\delta)), \\
- \text{round}_{\text{cost}}(\pi, A^*) \geq \delta.
\end{align*}
\]

Proof If there exists a misaligned point in \( \Psi \) during \( \pi \), according to Lemma 4, the assertion follows. If \( v \) is aligned throughout the period \( \phi(m) \), according to Lemma 3, the assertion again follows.
Now we focus on the other case, namely, all points in Ψ are aligned, and \( v \) is misaligned in \( \phi(m) \).

Let \( \rho_1, \ldots, \rho_n \) with \( T(\rho_i) \leq t(\rho_{i+1}) \) for each \( i \) be the pending requests at \( v \) that cause \( z_v \) to increase from \( \delta \) to \( 2\delta \). Choose \( t(\rho_1) \leq a_1 < T(\rho_1) \) and \( t(\rho_n) < b_n \leq T(\rho_n) \) such that \( z_v = \delta \) at time \( a_1 \) and \( z_v = 2\delta \) at time \( b_n \). Let \( a_i = t(\rho_i) \) for any \( 1 \leq i \leq n \), \( b_i = T(\rho_i) \) for any \( 1 \leq i \leq n \), and \( I_i = (a_i, b_i) \) for any \( 1 \leq i \leq n \). Then \( \sum_{i=1}^n \int_{I_i} f'(t - t(\rho_i)) dt = 2\delta - \delta = \delta \).

Now we have three observations.

1. **During each time interval \( I_i \), no request is pending outside \( \Psi \cup \{v\} \).**
   For contradiction, suppose there is a pending request \( \rho' \) in \( I_i \) with \( t(\rho') \notin \Psi \cup \{v\} \). Since \( z_v \geq \delta \) and there is a pending request \( \rho \) at \( v \) during \( I_i \), \( A \) should match \( \rho \) with \( \rho' \), contradictory to the assumption that \( m \) is between requests at \( \Psi \) and \( V \setminus \Psi \).

2. **During each time interval \( I_i \), no requests arrive at any point outside \( \Psi \cup \{v\} \).**
   Suppose on the contrary that a request \( \rho' \) arrives at \( u \notin \Psi \cup \{v\} \) during \( I_i \). By Observation \( \text{II} \) only \( v \) has a pending request outside \( \Psi \), which must get matched with \( \rho' \) due to the priority rule. This means that \( m \) occurs outside \( \Psi \), again contradictory to the assumption of the lemma.

3. **During each time interval \( I_i \), \( \Psi \) remains unchanged.**
   First, we argue that no point is added to \( \Psi \). Suppose on the contrary that some \( u \) is added to \( \Psi \) during \( I_i \). This means that an external match \( m' = (\rho, \rho') \) initiated by \( u \) occurs during \( I_i \). Without loss of generality, assume \( u = \ell(\rho), w = \ell(\rho') \). Since at any moment at most one request arrives, either \( \rho \) or \( \rho' \) arrives before \( m' \) occurs. By Observation \( \text{II} \) the only possibility is that \( \rho' \) arrives before \( m' \) occurs and \( w \notin \Psi \), which contradicts the priority rule of \( A \).

   Second, we show that no point is removed from \( \Psi \). Suppose on the contrary that some \( u \) is removed from \( \Psi \) during \( I_i \). Since no point is added to \( \Psi \) during \( I_i \), the size of \( \Psi \) decreases by one when \( u \) is removed, which is contradictory to the rule of updating \( \Psi \).

Since the number of misaligned points is even, at any moment during \( \bigcup_{i=1}^n I_i \) when \( v \) is misaligned, there must be a misaligned point outside \( \Psi \cup \{v\} \).

By the above observations and the definition of alignment status, for any \( 1 \leq i \leq n \), \( A^* \) must have a request \( \rho'_i \) that is pending throughout \( I_i \). Let \( u_i = \ell(\rho'_i) \) for any \( 1 \leq i \leq n \).

Since each \( \rho'_i \) is pending throughout \( I_i \) and \( f' \) is increasing,

\[
C_{time}(u_i, I_i, A^*) \geq \int_{I_i} f'(t - t(\rho'_i)) dt \geq \int_{I_i} f'(t - a_i) dt = f(b_i - a_i).
\]
As a result,

\[ \text{round}_{\text{cost}}(\pi, A^*) \geq \sum_{i=1}^{n} C_{\text{time}}(u_i, I_i, A^*) \]

\[ \geq \sum_{i=1}^{n} f(b_i - a_i) \]

\[ \geq f(f^{-1}(2\delta) - f^{-1}(\delta)), \]

where the last inequality follows from Lemma 2.

\[ \square \]

It is time to prove the following key lemma, stating that in every good complete round of $A$, the cost of the optimum offline algorithm $A^*$ is not small.

**Lemma 6** In any good complete round $\pi$, one of the following three claims must be true:

- $\text{round}_{\text{cost}}(\pi, A^*) \geq f(f^{-1}(2\delta) - f^{-1}(\delta))$.
- $\text{round}_{\text{cost}}(\pi, A^*) \geq \delta$.

**Proof** Let $M$ be the set of external matches $A$ outputs during $\pi$. By definition, $|M| = 2k$. Let $M' = \{m \in M : m \text{ causes } |\Psi| \text{ to increase by one}\}$ and $M'' = M \setminus M'$. Since any $m \in M''$ does not change $|\Psi|$ and $|\Psi| \leq k - 1$, we have $|M''| \leq k - 1$, which in turn implies $|M''| \geq k + 1$. There must be a point $v \in V$ which initiates at least two external matches in $M''$. Let $m \in M''$ be the second external match initiated by $v$. Obviously, $m$ is a native match of $\pi$.

Now we proceed case by case.

**Case 1:** $v \in \Psi$ during $\phi(m)$. If $v$ is aligned in $\phi(m)$, by Lemma 5 we have $\text{round}_{\text{cost}}(\pi, A^*) \geq \delta$. Otherwise, either $\text{round}_{\text{cost}}(\pi, A^*) \geq \delta$ or $\text{round}_{\text{cost}}(\pi, A^*) \geq \delta$ by Lemma 4.

**Case 2:** $v \notin \Psi$ during $\phi(m)$. Assume $m = (\rho, \rho')$ and $v = \ell(\rho), u = \ell(\rho')$. Since $m \in M''$, it hold that $u \in \Psi$ when $m$ occurs. Applying Lemma 5 we finish the proof. \[ \square \]

Up to now, we have focused on good rounds. The next lemma indicates that the cost of $A^*$ in bad rounds can be ignored in some sense. Here, a match/round is called bad if it is not good.

**Lemma 7** The number of bad rounds of $A$ is at most twice the number of external matches of $A^*$.

**Proof** The only event that causes a match of $A$ to be bad is an external match of $A^*$. An external match of $A^*$ changes the alignment status of at most two points, hence leading to at most two bad match of $A$, which in turn incurs at most two bad rounds. \[ \square \]
For any $v \in V$ and $t \in \mathbb{R}^+$, define $z_{v,t}$ to be the value of $z_v$ at time $t$. Then we begin to handle the cost of $A^*$ when $z_{v,t}$ is too big, namely, on the time set $\Gamma = \{ t : A \text{ has a pending request } \rho \text{ at time } t \text{ and } z_{t(\rho),t} > 2\delta \}$. More notation is needed.

For any $v \in V$ and $t \in \mathbb{R}^+$, we say that $t$ is critical time of $v$ if $t$ is the time when the last match involving $v$ occurs or when an external match involving $v$ occurs. Let $A = \{(v,t) \in V \times \mathbb{R}^+ : t \text{ is critical time of } v \}$. For any $(v,t) \in A$, define $\Gamma_{v,t} = \emptyset$ be the emptyset if $z_{v,t} < 2\delta$, otherwise $\Gamma_{v,t} = \{ t' < t : v \text{ has a pending request at time } t' \text{ and } z_{v,s} > 2\delta \text{ for } t' \leq s \leq t \}$.

We have the following easy observations:

- All the $\Gamma_{v,t}$'s are pairwise disjoint and $\Gamma = \bigcup_{(v,t) \in A} \Gamma_{v,t}$.
- The size of $\Lambda$ is at most $k$ plus twice the number of external matches of $A$.
- Since the timers $z_v$ are reset to 0 if and only external matches involving $v$ occurs, cost($A$) = $\sum_{(v,t) \in A} z_{v,t}$.

For any $x \in \mathbb{R}^+$, define its truncated value with respect to $y \geq 0$ to be

$$\text{trun}(x; y) = \begin{cases} 0 & \text{if } x \leq y \\ f(f^{-1}(x) - f^{-1}(y)) & \text{otherwise} \end{cases}$$

Now we derive a lower bound of the time cost of $A^*$ on every $\Gamma_{v,t}$.

**Lemma 8** For any $(v,s) \in \Lambda$,

$$\sum_{u \in V} C_{\text{time}}(u, \Gamma_{v,s}, A^*) \geq \text{trun}(z_{v,s}; 2\delta).$$

**Proof** Basically, the proof is similar to that of Lemma 5. It is enough to consider the case where $\Gamma_{v,s} \neq \emptyset$.

Suppose that $\Gamma_{v,s}$ consists of disjoint intervals $I_i = (a_i, b_i]$ for $1 \leq i \leq n$, and $b_i < a_{i+1}$ for $1 \leq i < n$. Then there are pending requests $\rho_1, \ldots, \rho_n$ at point $v$ such that

- $T(\rho_i) = b_i$ for $1 \leq i \leq n$, $t(\rho_i) \leq a_i$, $t(\rho_i) = a_i$ for $1 < i < n$,
- $\sum_{i=1}^{n} c_i = z_{v,s} - 2\delta$, where $c_i = \int_{a_i}^{b_i} f'(t - t(\rho_i))dt$ for $1 \leq i \leq n$.

At any time $t \in I_i$, $A$ has no pending requests at points other than $v$, meaning that totally an odd number of requests have arrived by time $t$. Since a match consumes two requests, $A^*$ must also have pending requests throughout each time interval $I_i$. Furthermore, note that no requests arrive during $I_i$. Hence, for each $1 \leq i \leq n$, $A^*$ has a request $\rho_i'$ at some $u_i$ that is pending throughout $I_i$. Considering that $f'$ is increasing, we have

$$C_{\text{time}}(u_i, I_i, A^*) \geq \int_{I_i} f'(t - t(\rho_i'))dt \geq \int_{I_i} f'(t - a_i)dt = f(b_i - a_i).$$
Therefore,
\[
\sum_{u \in V} C_{\text{time}}(u, \Gamma_{v,j}, A^*) \geq \sum_{i=1}^{n} C_{\text{time}}(u_i, I_i, A^*)
\geq \sum_{i=1}^{n} f(b_i - a_i)
\geq \text{trun}(z_{v,s}; 2\delta). \quad \text{(by Lemma 2)}
\]

The lemma thus holds. \(\square\)

We are ready to obtain a lower bound of the time cost of \(A^*\).

**Lemma 9** cost\(_{A^*}(R) \geq \sum_{\pi \in \Pi} \text{round}_{\text{cost}}(\pi, A^*) + \sum_{(v,s) \in A} \text{trun}(z_{v,s}; 2\delta).\)**

**Proof** It is easy to see that
\[
\text{cost}_{A^*}(R) \geq \sum_{\pi \in \Pi} \sum_{u \in V} C_{\text{time}}(u, \pi, A^*)
\geq \sum_{\pi \in \Pi} \sum_{u \in V} \left( C_{\text{time}}(u, \pi \setminus \Gamma, A^*) + C_{\text{time}}(u, \pi \cap \Gamma, A^*) \right)
\geq \sum_{\pi \in \Pi} \text{round}_{\text{cost}}(\pi, A^*) + \sum_{u \in V} C_{\text{time}}(u, \Gamma, A^*)
\]
Furthermore,
\[
\sum_{u \in V} (C_{\text{time}}(u, \Gamma, A^*)) = \sum_{u \in V} C_{\text{time}}(u, \bigcup_{(v,s) \in A} \Gamma_{v,s}, A^*)
= \sum_{u \in V} \sum_{(v,s) \in A} C_{\text{time}}(u, \Gamma_{v,s}, A^*)
= \sum_{(v,s) \in A} \sum_{u \in V} C_{\text{time}}(u, \Gamma_{v,s}, A^*)
\geq \text{trun}(z_{v,s}; 2\delta) \quad \text{(by Lemma 5)}
\]

\(\square\)

The following technical lemmas will be needed.

**Lemma 10** Given \(c, c_0, c_1, \ldots, c_n \in \mathbb{R}^+\) with \(c_i \geq c_0 > c\) for any \(1 \leq i \leq n\), we have
\[
\frac{\sum_{j=1}^{n} (c_j - c)}{\sum_{j=1}^{n} \text{trun}(c_j; c)} \leq \frac{c_0 - c}{\text{trun}(c_0; c)}.
\]

**Proof** Recall that \(f(t) = t^\alpha\) and \(\alpha > 1\). It suffices to prove that \(\frac{a-b}{(\sqrt{\pi} - \sqrt{b})^\alpha (x-y)^{\alpha-1}}\) decreases with \(a\) when \(a > b\). This is equivalent to showing \(g(x) = \frac{a-b}{(\sqrt{\pi} - \sqrt{b})^\alpha (x-y)^{\alpha-1}}\) decrease with \(x\) when \(x > y\). The claim holds since \(g'(x) = \alpha \cdot \frac{(x-y)^{\alpha-1}}{(x-y)^{\alpha+1}} \leq 0\).
\(\square\)
Lemma 11 If $\mathcal{A}$ has only one round on input instance $R$, $\frac{\text{cost}_A(R)}{\text{cost}_{A^*}(R)} = O(k)$.

Proof Denote the round by $\pi$. We proceed case by case.

Case 1: Both $\mathcal{A}$ and $\mathcal{A}^*$ have no external matches. Then they must behave in the same way on $R$. Hence $\text{cost}_A(R)/\text{cost}_{A^*}(R) = 1$.

Case 2: Either $\mathcal{A}$ or $\mathcal{A}^*$ has an external match. If $\mathcal{A}^*$ has no external matches, the first external match of $\mathcal{A}$ must be a good native match, so $\text{round}_{\text{cost}_{\text{time}}}(\pi, \mathcal{A}^*) \geq \delta$ by Lemma 3. If $\mathcal{A}^*$ has an external match, $\text{cost}_{A^*}(R) \geq \delta$. So, we always have $\text{cost}_A(R) + \text{round}_{\text{cost}_{\text{time}}}(\pi, \mathcal{A}^*) \geq \delta$.

On the one hand, $\mathcal{A}$ has at most $2k$ external matches in a round, so $\text{cost}_A(R) \leq 2k\delta + \sum_{(v,s) \in \Lambda} z_{v,s}$. Let $\Lambda' = \{(v,s) \in A : z_{v,s} > 4\delta\}$. Because $|\Phi| \leq 5k$, it holds that $\text{cost}_A(R) \leq 22k\delta + \sum_{(v,s) \in \Lambda'} (z_{v,s} - 2\delta)$.

On the other hand, as to the cost of $\mathcal{A}^*$, we have $\text{cost}_{A^*}(R) = \text{cost}_{s_{A^*}}(R) + \text{cost}_{t_{A^*}}(R) \geq \delta + \sum_{(v,s) \in \Lambda'} \text{trun}(z_{v,s}; 2\delta)$.

By Lemma 10, $\frac{\text{cost}_A(R)}{\text{cost}_{A^*}(R)} \leq \frac{22k\delta + \sum_{(v,s) \in \Lambda'} (z_{v,s} - 2\delta)}{\delta + \sum_{(v,s) \in \Lambda'} \text{trun}(z_{v,s}; 2\delta)} = O(k)$.

Now we are ready to prove the first 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 $m > 1$ without loss of generality.

Because $\mathcal{A}$ has at most $2mk$ external matches, we have $|\Lambda| \leq k + 2 \times 2mk < 5mk$. Let $\Lambda' = \{(v,s) \in A : z_{v,s} > 4\delta\}$. Then it holds that

$$\text{cost}_A(R) = \text{cost}_{s_{A}}(R) + \text{cost}_{t_{A}}(R) \leq 2km\delta + \sum_{(v,s) \in \Lambda} z_{v,s} \leq 22km\delta + \sum_{(v,s) \in \Lambda'} (z_{v,s} - 4\delta) \leq 22km\delta + \sum_{(v,s) \in \Lambda'} (z_{v,s} - 2\delta).$$
On the other hand, as for the cost of $A^*$, we have by Lemma 9 that
\[
\text{cost}_{A^*}(R) = \text{cost}_{A^*}^s(R) + \text{cost}_{A^*}^t(R)
\]
\[
\geq \text{cost}_{A^*}^s(R) + \sum_{\pi \in \Pi} \text{round cost}_{\text{time}}(\pi, A^*) + \sum_{(v,s) \in \Lambda} \text{trun}(z_{v,s}; 2\delta)
\]
\[
= \sum_{\pi \in \Pi} (\text{round cost}_{\text{space}}(\pi, A^*) + \text{round cost}_{\text{time}}(\pi, A^*))
\]
\[
+ \sum_{(v,s) \in \Lambda'} \text{trun}(z_{v,s}; 2\delta).
\]

Let $\Pi'$ 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 7, $A^*$ has at least $m''$ external matches, meaning that $\text{cost}_{A^*}(R) \geq \text{cost}_{A^*}^s(R) \geq m'' 2\delta$. As a result,
\[
2\text{cost}_{A^*}(R) \geq \sum_{\pi \in \Pi} (\text{round cost}_{\text{space}}(\pi, A^*) + \text{round cost}_{\text{time}}(\pi, A^*))
\]
\[
+ \sum_{(v,s) \in \Lambda'} \text{trun}(z_{v,s}) + \frac{m''}{2}\delta
\]
\[
\geq f(f^{-1}(2\delta) - f^{-1}(\delta))m' + \frac{m''}{2}\delta + \sum_{(v,s) \in \Lambda'} \text{trun}(z_{v,s}; 2\delta)
\]
\[
\geq \frac{m - 1}{2} (\sqrt{2} - 1)\alpha\delta + \sum_{(v,s) \in \Lambda'} \text{trun}(z_{v,s}; 2\delta)
\]
where the second inequality is due to Lemma 6.

By Lemma 10,
\[
\frac{\text{cost}_{A^*}(R)}{\text{cost}_{A^*}(R)} \leq \frac{22km\delta + \sum_{(v,s) \in \Lambda'} (z_{v,s} - 2\delta)}{\frac{m}{2} (\sqrt{2} - 1)\alpha\delta + \frac{1}{2} \sum_{(v,s) \in \Lambda'} \text{trun}(z_{v,s}; 2\delta)} = O(k).
\]

\[\Box\]

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 $f : \mathbb{R}^+ \to \mathbb{R}^+$ be a non-decreasing, unbounded, continuous function satisfying $f(0) = f'(0) = 0$. Let $S = (V, \delta)$ be a uniform metric space with $V = \{v_0, v_1, \ldots, v_k\}$. Suppose that $A$ is an arbitrary deterministic online algorithm for the $f$-MPMD problem. Let $T \in \mathbb{R}^+$ be such that $f(T) = k\delta$. Arbitrarily fix a real number $\tau > 0$ such that $n = \frac{T}{\tau}$ is an even number.

We construct an instance $R$ of online input to $A$ and show that the competitive ratio of $A$ is at least $O(k)$. The instance $R$ is determined in an online fashion: Roughly speaking, based on the up-to-now behavior of $A$, we decide
the time and locations of the following requests so that $\mathcal{A}$ has to take many external matches.

Specifically, $R$ is constructed 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 two 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 fix $1 \leq r \leq m$, and we describe the requests that arrive during the $r$th round, namely in the interval $[T_r, T_{r+1})$. Define $G_0 = (V, \emptyset)$ to be the edgeless graph on $V$, and let $C_0 = \{v_0\}$, $h = 1$. At time $T_r$, a request $\rho_{00}$ arrives at $v_0$. The other requests are specified as follows.

For any integers $(h-1)n < j \leq hn$ and $1 \leq i \leq k$ such that $v_i \in V \setminus C_{h-1}$, a request $\rho_{ij}$ arrives at point $v_i$ at time $T_r + j\tau$. When $\rho_{i,hn}$ arrives, namely, at time $T_r + hT$, construct an undirected graph $G_h$ on $V$. It has an edge between any pair of vertices $v_i \neq v_j$ if and only if by time $T_r + hT$, $\mathcal{A}$ has matched one request at $v_i$ and another at $v_j$ both arriving in the period $[T_r, T_r + hT]$. Let $C_h$ be the set of the vertices in the connected component of $G_h$ that contains $v_0$. If $C_{h-1} \neq C_h \neq V$, increase $h$ by 1 and iterate the process described in this paragraph. Otherwise, arbitrarily choose $i$ such that $v_i \in V \setminus C_{h-1}$, let one more request $\rho_{i,hn+1}$ arrive at point $v_i$ at time $T_r + hT + \tau$, and finish the construction of round $r$; denote the final $h$ by $h_r$.

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 in round $r$ arrives.
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 of this section.

Lemma 12 $\text{cost}_A(R_r) \leq (\delta + \frac{k^2n}{2} f(\tau) + f(\tau))r$.

Proof It suffices to show that the cost that $\mathcal{A}^*$ pays for any round is at most $\delta + \frac{k^2n}{2} f(\tau) + f(\tau)$. Without loss of generality, we prove this for the first round and assume that the last request of this round is located at $v_k$. By Facts 2 and 3, the requests of this round can be paired up in this way: $\langle \rho_{00}, \rho_{k1} \rangle$, $\langle \rho_{ij}, \rho_{i,j+1} \rangle$ for $1 \leq i \leq k-1$ and odd number $j \geq 1$, and $\langle \rho_{kj}, \rho_{k,j+1} \rangle$ for even numbers $j \geq 2$. Since $\mathcal{A}^*$ is an optimum offline algorithm, its cost is at most the cost of this matching. \hfill $\square$

Lemma 13 $\text{cost}_A(R_r) \geq k\delta r$.

Proof By Fact 4, it is equivalent to show that the cost that $\mathcal{A}$ pays for requests in $R_r \setminus R_{r-1}$ is at least $k\delta$. We proceed case by case.
Case 1: $C_{h_r} = C_{h_r-1}$. On the one hand, assume Case 2 in this round does happen. We have three observations:

- After time $T_r + (h_r - 1)T$, no request arrives at any $v \in C_{h_r}$.
- The total number of requests that have arrived at points in $C_{h_r}$ is an odd number. Hence, there must be a request $\rho$ such that (1) $\ell(\rho) \in C_{h_r}$ and (2) $A$ eventually matches $\rho$ with another request $\rho'$ satisfying $\ell(\rho') \notin C_{h_r}$.
- Since no external match occurs between $C_{h_r}$ and $V \setminus C_{h_r}$ in period $(T_r + (h_r - 1)T, T_r + hT)$, the request $\rho$ must be pending throughout the period, incurring time cost at least $f(T) = k\delta$.

Case 2: $C_{h_r} = V$. Then $A$ has at least $k$ external matches in this round. Altogether, the cost $A$ pays for this round is at least $k\delta$. □

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 $m \geq 1$, $\text{cost}_A(R) \leq a \cdot \text{cost}_A(R) + b$. Fix $k$ and $\delta$. Dividing both sides of inequality by $m$ and letting $m$ approach infinity, by Lemmas 12 and 13 we get $f(n\tau) \leq (\delta + k^2 n f(\tau) + f(\tau))a$, which means that $a \geq \frac{f(n\tau)}{\delta + k^2 n f(\tau) + f(\tau)} = \frac{k\delta + \frac{1}{2} f(n\tau)}{\delta + k^2 n f(\tau) + f(\tau)}$.

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

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

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

6 Lower Bound for Randomized Algorithms

As in Section 5, arbitrarily fix a nondecreasing, unbounded, continuous function $f : \mathbb{R}^+ \to \mathbb{R}^+$ which satisfies $f(0) = f'(0) = 0$. Let $S = (V, \delta)$ be a uniform metric space with $V = \{v_0, v_1, ..., v_k\}$. This section is devoted to show that against oblivious adversaries, the competitive ratio of any randomized algorithm for $f$-MPMD is lower bounded by $\Omega(\ln k)$.

By Yao’s principle, it is enough to show that there is a randomized input instance on $S$ such that any deterministic $f$-MPMD algorithm must have expected competitive ratio $\Omega(\ln k)$.

To construct the random instance, let $T \in \mathbb{R}^+$ be such that $f(T) = k\delta \ln k$.

 Arbitrarily choose real number $\tau > 0$ such that $n = \frac{T}{\tau}$ is an even number. Let $R$ be an instance of online input over $S$ such that

- Point $v_0$ has one request, which arrives at time $t = 0$;
- For $1 \leq i \leq k - 1$, each point $v_i$ has $ni$ requests, which arrive at time $j\tau$, $1 \leq j \leq ni$, respectively;
- Point $v_k$ has $kn + 1$ requests, which arrive at time $j\tau$, $1 \leq j \leq kn + 1$, respectively.

An example of $R$ with $k = 3$ is illustrated in Figure 4(a).

Let $\Xi_k$ be the set of permutations on $\{0, 1, ..., k\}$. For any permutation $\sigma \in \Xi_k$, define $R_\sigma$ to be the online input instance in which each point $v_{\sigma(i)}$ has the same requests as $v_i$ does in $R$, $0 \leq i \leq k$. An example of $R_\sigma$ is illustrated in Figure 4(b).

Let $\sigma$ be a random permutation uniformly distributed on $\Xi_k$. Then $R_\sigma$ is uniformly distributed on the set $\{R_\sigma : \sigma \in \Xi_k\}$.

Arbitrarily fix a deterministic algorithm $A$ solving $f$-MPMD problem. Let $A^*$ be the optimum offline algorithm. The main task is to compare the expected cost of $A$ with that of $A^*$ on the input instance $R_\sigma$. Let’s begin with some lemmas, where $E[\cdot]$ means expectation.

**Lemma 14** When $k$ is fixed, $\lim_{\tau \to 0} E[\text{cost}_{A^*}(R_\sigma)] \leq \delta$.

**Proof** Consider the following offline matching of $R_\sigma$: one external match occurs between the request at point $v_{\sigma(0)}$ and the first request at $v_{\sigma(k)}$, and the other matches are internal and occur between requests that arrive consecutively. The cost is $\delta + \left(\frac{n(k+1)}{4} + 1\right)f(\tau) \leq \delta + n(k + 1)^2f(\tau)$. Because $A^*$ is an optimum offline algorithm, it must hold that $\text{cost}_{A^*}(R_\sigma) \leq \delta + n(k + 1)^2f(\tau)$.

Furthermore, since $k$ is fixed,

$$\lim_{\tau \to 0} nf(\tau) = \lim_{\tau \to 0} n\frac{f(\tau)}{\tau} = f^{-1}(k\delta \ln k)f'(0) = 0,$$

which immediately leads to the lemma. \qed

Now we consider the behavior of $A$ on the random input instance $R_\sigma$.

**Lemma 15** If with nonzero probability $A$ keeps some request pending no shorter than $T$, $E[\text{cost}_{A}(R_\sigma)] \geq \delta \ln k$.

**Proof** Let $\sigma' \in \Xi_k$ be such that on the input instance $R_{\sigma'}$, $A$ keeps a request pending for a period at least $T$. The time cost incurred by this delay is at least $f(T) = k\delta \ln k$, so $\text{cost}_{A}(R_{\sigma'}) \geq k\delta \ln k$. Since $\sigma$ is uniformly distributed on $\Xi_k$, we have $\Pr(\sigma = \sigma') = \frac{1}{k!}$, which implies that $E[\text{cost}_{A}(R_{\sigma})] \geq \Pr(\sigma = \sigma')\text{cost}_{A}(R_{\sigma'}) \geq \delta \ln k$. \qed
Actually Lemma 15 remains true even if no request is delayed too long, as claimed in the next lemma.

**Lemma 16** If $A$ doesn’t keep any request pending longer than $T$, $E[\text{cost}_A(R_{\sigma})] \geq \delta \Theta(\ln k)$.

**Proof** The basic idea of the proof is to show that in expectation, $A$ has $\ln k$ external matches. The number of external matches can be estimated by sampling the points where an externally matched request is located. This sampling is done sequentially while $A$ is running. The detail is described below.

Suppose that we have already sampled $u_1, \ldots, u_{h-1} \in V$ for some $h \geq 1$. If $h = 1$, let $j_0 = 0$. If $h > 0$, wait until no more requests arrive at $u_{h-1}$, and by definition of $R_{\sigma}$, we can determine both $j_{h-1}$ such that $u_{h-1} = v_{\sigma(j_{h-1})}$ and $\sigma(j)$ for $j \leq j_{h-1}$. Then we proceed case by case.

**Case 1:** $j_{h-1} = k$. The sampling process ends and output $\{u_1, \ldots, u_{h-1}\}$ as the final sample. Let $h_{\sigma} = h - 1$.

**Case 2:** $j_{h-1} < k$. At time $(j_{h-1} + 1)T$, let $V_h = \{v_{\sigma(j)} : j \leq j_{h-1}\}$ and
\[
U_h = \{u \in V \setminus V_h : \text{there is a match } (\rho, \rho') \text{ with } l(\rho) = u, l(\rho') \in V_h\}.
\]
Define $R_h = \{\rho \in R_{\sigma} : f(\rho) \in V_h\}$. Note that all the requests in $R_h$ have arrived by time $j_{h-1}T$, hence being matched by time $(j_{h-1} + 1)T$ according to the assumption that $A$ doesn’t keep any request pending longer than $T$. Because $j_{h-1} < k$, $R_h$ has an odd number of requests, implying that by time $(j_{h-1} + 1)T$, $A$ must have matched one request in $R_h$ with another located outside $V_h$. As a result, $U_h \not= \emptyset$. Arbitrarily choose $u_h \in U_h$, and iterate the process with already-sampled points $u_1, \ldots, u_h$.

The above sampling process finally produce a sample $\{u_1, \ldots, u_{h_{\sigma}}\}$. Recall the indices $j_i$ satisfies $v_{\sigma(j_i)} = u_i$, $1 \leq i \leq h_{\sigma}$. Due to the randomness of $\sigma$, all the indices $j_i$ and $h_{\sigma}$ are random variables. It is easy to observe two facts.

**Fact 1:** for any $1 \leq h \leq h_{\sigma}$, given $j_{h-1}$, $j_h$ is uniformly distributed on the set $\{j_{h-1} + 1, \ldots, k\}$. This follows from the uniformly distribution of $R_{\sigma}$.

**Fact 2:** $A$ has at least $h_{\sigma}$ external matches. This obviously holds.

Now we estimate $E[h_{\sigma}]$. For any integer $1 \leq i \leq k$, let $S_i$ stands for the set $\{i, \ldots, k\}$, and define a random variable $X_i$ by induction as follows:

- Uniformly randomly sample $c_1$ from $S_1$.
- Suppose $c_{i-1}$ has been sampled. If $c_{i-1} < k$, uniformly randomly sample $c_i$ from $S_{c_{i-1}+1}$, and iterate this step with $h$ increased by 1. Otherwise, the induction ends and let $X_i = h_{\sigma} - 1$.

We can see that for any $1 \leq i \leq k$, $E[X_i] = 1 + \frac{1}{i+1} \sum_{j=i+1}^{k} E[X_j]$. Considering that $E[X_k] = 1$, we have $E[X_1] = \sum_{j=1}^{k} \frac{1}{j}$. By Fact 1 mentioned above, $h_{\sigma}$ and $X_1$ are identically distributed. As a result, $E[h_{\sigma}] = E[X_1] = \Theta(\ln k)$.

Since an external match incurs space cost $\delta$, the proof ends due to Fact 2. \qed
Online Matching with Convex Delay Costs

For any $t \in \mathbb{R}^+$, define an online input instance $R^{(t)}$ which is derived from $R$ by deferring the arrival of each request for a period of length $t$. Likewise define $R^{(t)}_{\sigma}$ for any $\sigma \in \Xi_k$. Arbitrarily fix a positive integer $m$. For any $1 \leq r \leq m$, let $\sigma_1, \ldots, \sigma_m$ be independent random variables that are uniformly distributed on $\Xi_k$. Our ultimate random online input instance is $R = \bigcup_{r=1}^m R^{(T_r)}_{\sigma_r}$, where $T_1 = 0$ and for any $1 \leq r \leq m$, $T_r$ is arbitrary time before which all requests in $\bigcup_{i=1}^{r-1} R^{(T_i)}_{\sigma_i}$ have been matched. The instance $R$ is illustrated in Figure 5.

By the definition of $R$, we know that any pair of requests paired up by $A$ must belong to the same $R^{(T_r)}_{\sigma_r}$, for some $1 \leq r \leq m$. Hence, $\text{cost}_A(R) \leq \sum_{r=1}^m \text{cost}_A(R^{(T_r)}_{\sigma_r})$ and $\text{cost}_A^*(R) = \sum_{r=1}^m \text{cost}_A^*(R^{(T_r)}_{\sigma_r})$. This, together with the trivial observation that Lemmas 14-16 remain true for any instance $R^{(T_r)}_{\sigma_r}$, immediately leads to the following lemma.

**Lemma 17** \[E[\text{cost}_A(R)] \geq m\delta \Theta(\ln k), \text{ and } \lim_{\tau \to 0} E[\text{cost}_A^*(R)] \leq m\delta \text{ when } k \text{ is fixed.}\]

Now we are ready to prove the main result of this section.

**Theorem 5** Suppose that the time cost function $f$ is nondecreasing, unbounded, continuous and satisfies $f(0) = f'(0) = 0$. Then any randomized algorithm for $f$-MPMD on $k$-point uniform metric space has competitive ratio $\Omega(\ln k)$ against oblivious adversaries.

**Proof** Arbitrarily choose a deterministic online algorithm $A$ for the $f$-MPMD problem. Let $A^*$ be an optimum offline algorithm for the $f$-MPMD problem. Consider the random online input instance $R$ constructed above.

Suppose there are $a = a(k, \delta)$ and $b = b(k, \delta)$ such that for any $m \geq 1$,

\[E[\text{cost}_A(R)] \leq a \cdot E[\text{cost}_A^*(R)] + b.\]

Fix $k$ and $\delta$. Letting $\tau$ approach zero, by Lemma 17 we have $m\delta \Omega(\ln k) \leq a \cdot m\delta + b$. Dividing both sides of the inequality by $m$ and letting $m$ approach infinity, we get $a = \Omega(\ln k)$. The lemma is proven due to Yao’s principle. \qed

7 Conclusion

We have formulated the convex-MPMD problem and have designed a deterministic online algorithm to solve this problem. For any monomial function
If \( f(t) = t^\alpha \) with \( \alpha > 1 \), this algorithm has competitive ratio \( O(k) \) if requests are from a \( k \)-point uniform metric space. Actually, this result can be generalized to any convex function that satisfies (a) \( f(0) = f'(0) = 0 \), (b) \( \ln f(t) \) is a concave function, and (c) for any \( \lambda_1 < \lambda_2 \in \mathbb{R}^+ \), \( \sup_{x \in \mathbb{R}^+} \frac{x}{f^{-1}(\lambda_2 x) - f^{-1}(\lambda_1 x)} < \infty \). Under these conditions, all the lemmas remain true and the proof of Theorem 1 remains valid.

Our algorithm can be adapted to work on general metric spaces. Given a metric space \( S \), let \( d_{\text{max}} \) (respectively, \( d_{\text{min}} \)) stand for the maximum (respectively, minimum) pairwise distance among \( S \). To handle requests from \( S \), replaced \( \delta \) in the algorithm by \( d_{\text{max}} \). All the lemmas in Section 4 remain true up to the slight modifications: \( \delta \) in the lower bounds of the space cost of \( \mathcal{A}^* \) is replaced by \( d_{\text{min}} \), and \( \delta \) in the upper bounds of of the space cost \( \mathcal{A} \) is replaced by \( d_{\text{max}} \). As a result, Theorem 1 still holds, if the upper bound \( O(k) \) is relaxed to \( O(k \Delta) \), where \( \Delta \triangleq \frac{d_{\text{max}}}{d_{\text{min}}} \).

We have also proved lower bound \( \Omega(k) \) (respectively, \( \Omega(\log k) \)) of the competitive ratio of any deterministic (respectively, randomized) algorithm solving convex-MPMD. This means that our deterministic algorithm is the optimum while optimum randomized algorithms remain open.

Hence, 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 \( O(\log k) \). 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 concave cost functions, but to design a good one, i.e., with a very small competitive ratio, seems still challenging.

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).

References

1. G. Aggarwal, G. Goel, C. Karande, and A. 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. I. Ashlagi, Y. Azar, M. Charikar, A. Chiplunkar, O. Geri, H. Kaplan, R. Makhljani, Y. Wang, and R. 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. Y. Azar, A. Chiplunkar, and H. Kaplan. Polylogarithmic Bounds on the Competitiveness of Min-cost Perfect Matching with Delays. *abs/1610.05155*, 2016.
4. Y. Azar, A. Chiplunkar, and H. 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. Y. Azar, Y. Emek, R. van Stee, and D. Vainstein. The price of clustering in bin-packing with applications to bin-packing with delays. In The 31st ACM Symposium on Parallelism in Algorithms and Architectures, pages 1–10, 2019.
6. Y. Azar and A. J. Fanani. Deterministic min-cost matching with delays. In International Workshop on Approximation and Online Algorithms, pages 21–35. Springer, 2018.
7. Y. Azar, A. Ganesh, R. Ge, and D. Panigrahi. Online service with delay. In Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, pages 551–563. ACM, 2017.
8. Y. Azar and N. Touitou. General framework for metric optimization problems with delay or with deadlines. In 2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS), pages 60–71. IEEE, 2019.
9. M. Bienkowski, A. Kraska, H.-H. Liu, and P. Schmidt. A primal-dual online deterministic algorithm for matching with delays. In International Workshop on Approximation and Online Algorithms, pages 51–68. Springer, 2018.
10. M. Bienkowski, A. Kraska, and P. Schmidt. A match in time saves nine: Deterministic online matching with delays. In International Workshop on Approximation and Online Algorithms, pages 132–146. Springer, 2017.
11. B. E. Birnbaum and C. Mathieu. On-line bipartite matching made simple. SIGACT News, 39(1):80–87, 2008.
12. N. R. Devanur, K. Jain, and R. 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.
13. D. R. Dooley, S. A. Goldman, and S. D. Scott. Tcp dynamic acknowledgment delay (extended abstract) theory and practice. In Proceedings of the thirteenth annual ACM symposium on Theory of computing, pages 389–398, 1998.
14. D. R. Dooley, S. A. Goldman, and S. D. Scott. On-line analysis of the tcp acknowledgment delay problem. Journal of the ACM (JACM), 48(2):243–273, 2001.
15. J. Edmonds. Maximum matching and a polyhedron with 0, 1-vertices. Journal of Research of the National Bureau of Standards B, 69:125–130, 1965.
16. J. Edmonds. Paths, trees, and flowers. Canadian Journal of Mathematics, 17:449–467, 1965.
17. Y. Emek, S. Kutten, and R. Wattenhofer. Online Matching: Haste makes Waste! In 48th Annual Symposium on Theory of Computing (STOC), June 2016.
18. Y. Emek, Y. Shapiro, and Y. Wang. Minimum Cost Perfect Matching with Delays for Two Sources. In 10th International Conference on Algorithms and Complexity (CIAC), Athens, Greece, May 2017.
19. G. Goel and A. 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.
20. B. Kalyanasundaram and K. Pruhs. Online Weighted Matching. J. Algorithms, 14(3):478–488, 1993.
21. A. R. Karlin, M. S. Manasse, L. A. McGeoch, and S. Owicki. Competitive randomized algorithms for nonuniform problems. Algorithmica, 11(6):542–571, 1994.
22. A. R. Karlin, M. S. Manasse, L. Rudolph, and D. D. Sleator. Competitive snoopy caching. Algorithmica, 3(1-4):79–119, 1988.
23. R. M. Karp, U. V. Vazirani, and V. 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.
24. S. Khuller, S. G. Mitchell, and V. V. Vazirani. On-Line Algorithms for Weighted Bipartite Matching and Stable Marriages. Theor. Comput. Sci., 127(2):255–267, 1994.
25. A. Mehta. Online Matching and Ad Allocation. Foundations and Trends in Theoretical Computer Science, 8(4):265–368, 2013.
26. A. Mehta, A. Saberi, U. V. Vazirani, and V. V. Vazirani. AdWords and Generalized Online Matching. In 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS), pages 264–273, 2005.
27. A. Meyerson, A. Nanavati, and L. J. Poplawski. Randomized online algorithms for minimum metric bipartite matching. In Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, 2006.
28. S. Miyazaki. On the advice complexity of online bipartite matching and online stable marriage. Inf. Process. Lett., 114(12):714–717, 2014.
29. J. Naor and D. 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.