Impatient Online Matching

Xingwu Liu
Institute of Computing Technology, Chinese Academy of Sciences.
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 Zürich, Switzerland
yuwang@ethz.ch

Roger Wattenhofer
ETH Zürich, 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

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 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. 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 \( A \) 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 \( f\text{-MPMD} \) stands for the MPMD problem with time cost function \( f \):

\[ \text{Theorem 1.} \quad \text{For any } f(t) = t^\alpha \text{ with } \alpha > 1, \text{ the competitive ratio of } A \text{ for } f\text{-MPMD on } k\text{-point uniform metric space is } O(k). \]

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

\[ 1 \] https://www.hopkinsmedicine.org/transplant/programs/kidney/incompatible/paired_kidney_exchange.html
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 $\Delta$ 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 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, unlike the online matching problems.

3 Preliminaries

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

3.1 Problem Statement

Let $\mathbb{R}^+$ stand 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

Positive definite: $\mu(x, y) \geq 0$ for any $x, y \in V$, and “=” holds if and only if $x = y$;

Symmetrical: $\mu(x, y) = \mu(y, x)$ for any $x, y \in V$;

Subadditive: $\mu(x, y) + \mu(y, z) \geq \mu(x, z)$ 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 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 $\rho$ 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 $A$ 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 $A$ on input $R$, denoted by $\text{cost}_A(R)$, is the total space cost caused by all the matched pairs, and the time cost $\text{cost}_A'(R)$ is defined likewise. The objective of the $f$-MPMD is to find an online algorithm $A$ such that $\text{cost}_A(R) = \text{cost}_A'(R) + \text{cost}'_A(R)$ is minimized for all $R$.

As usual, the online algorithm $A$ is evaluated through competitive analysis. Let $A^*$ be an optimum offline algorithm. For any finite metric space $S$, if there are $a, b \in \mathbb{R}^+$ such that $\text{cost}_A(R) \leq \text{cost}_{A^*}(R) + b$ 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 $S$.

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

3.2 Notations and Terminologies

Any pair of requests $\rho, \rho'$ in the perfect matching is called a match between $\rho$ and $\rho'$ and denoted by $(\rho, \rho')$ or $(\rho', \rho)$ interchangeably. A match $(\rho, \rho')$ is said to be external if $\ell(\rho) \neq \ell(\rho')$. An offline algorithm knows the whole input instance at the beginning and outputs any pair $\rho, \rho' \in R$ at time $\max\{t(\rho), t(\rho')\}$. 

---

2 An offline algorithm knows the whole input instance at the beginning and outputs any pair $\rho, \rho' \in R$ at time $\max\{t(\rho), t(\rho')\}$.
A request \( \rho \) 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 is to prioritize internal matches and to create an external match only if both requests have waited long enough (say, as long as \( \theta \)). 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 4.

**Example 4.** For any positive integer \( n \) and small real number \( \epsilon > 0 \), construct an online instance as follows. 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 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 \( \epsilon \) approaches 0, \( n \cdot f(\theta - \epsilon) + f(n\theta) + \delta \gg \delta + n \cdot f(\epsilon) \), meaning that Strategy I is not competitive.

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 \( \theta \)). 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 \( \delta \). Arbitrarily fix an even integer \( n > 0 \) and a small real number \( \epsilon > 0 \). Arbitrarily choose \( \tau \in \mathbb{R}^+ \) such that \( \theta - \epsilon < \frac{T}{2} f(\tau) < \theta \).

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 \( 0 \leq i \leq n \). 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 \( \frac{\tau}{2} f(\tau) + f(n\tau) + \delta \). 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 \( \frac{\tau}{2} f(\tau) + \delta \). Thus the cost of \( \mathcal{A}^* \) is at most \( \frac{\tau}{2} f(\tau) + \delta \). When \( n \) approaches infinity and \( \epsilon \) approaches 0, we have \( \frac{\tau}{2} f(\tau) + f(n\tau) + \delta \gg \frac{\tau}{2} f(\tau) + \delta \), implying that Strategy II is not competitive.
Since the trouble may be rooted at the double-counter-enabling mechanism, we further improve the strategy by enabling an external match if one of the two points has high accumulated cost (say, as high as $\theta$). 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 $S = (\{u, v, w\}, \delta)$. 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 $\rho_{u,0}$ at time 0, $\rho_{0,0}$ at time $T_0$, and $\rho_{0,i}$ at time $T_0 + (n+i)\tau$ for any $1 \leq i \leq 2n$. $v$ receives a request $\rho_{v,j}$ at time $T_0 + i\tau$ for any $1 \leq i \leq 2n$. $w$ receives a request $\rho_{w,0}$ at time 0 and a request $\rho_{w,n+i}$ at time $T_0 + i\tau$ for any $1 \leq i \leq n$. Let $T_j = T_0 + (2n + 1)\tau$, $T_j = T_j - 3n\tau$ for any $2 \leq j \leq m$.

For any $1 \leq j \leq m$, the part $R_j$ has $6n$ requests as follows: $\rho_{j,0}$ arrives at $u$ at time $T_j + (2n + i - 1)\tau$, $\rho_{j,i}$ arrives at $v$ at time $T_j + (n + i - 1)\tau$, and $\rho_{j,i}$ arrives at $w$ at time $T_j + (i - 1)\tau$, for every $1 \leq i \leq 2n$.

Actually, we can very slightly perturb the arrival time of some requests so that Strategy III results in exactly the following external matches: $(\rho_{u,0}, \rho_{w,0}), (\rho_{u,0}, \rho_{v,0}), (\rho_{u,2n}, \rho_{v,2n}), (\rho_{v,2n}, \rho_{w,2n})$ for $1 \leq j \leq m$, $(\rho_{v,2n}, \rho_{v,1,n})$ and $(\rho_{v,2n}, \rho_{v,1,n})$ for $1 \leq i < m$, and $(\rho_{m,2n}, \rho_{m,2n})$, 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

$$f(T_0) + \frac{6mn + 5n - 1}{2} f(\tau).$$

When $\tau$ 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 $\rho$. On the contrary, SOL has accumulated little cost at $\ell(\rho)$, because SOL has no pending request there while $\rho$ 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 $z_\rho, \rho \in \mathbb{R}^+$, which is initially set to 0, for every point $v \in V$. 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 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 $z_\rho$ increases at rate $f'(t - t_0)$ if there is an active request $\rho$ at $v$ with $t(\rho) = t_0$.
2. Match any pair of active requests $\rho$ and $\rho'$ if $\ell(\rho) = \ell(\rho')$.
3. For any pair of active requests $\rho, \rho'$ 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. $z_x \geq 2\delta$, or
   b. $\delta \leq z_x < 2\delta$ and $\{u, v\} \cap \Psi = \emptyset$.

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

Priority rule: in applying Operation 3, 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 can be arbitrarily perturbed and timing in practice is up to errors. Let $A$ stands for our algorithm and $A^*$ for an optimum offline algorithm solving $f$-MPMD. We start competitive analysis by introducing notation.
4.3.1 Notations

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

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

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

$$C_{time}(v, I, A^*) = \sum_{r \in R, t(r) = v} C_{time}(r, I, A^*).$$

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

Define $\Gamma = \{t \in \mathbb{R}^+ : \text{at time } t, A \text{ has a pending request } r \text{ with } z_t > 2\delta\}$. We will analyze time cost of $A^*$ inside and outside $\Gamma$ separately.

Our algorithm $A$ 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 $\Pi$ be the set of rounds of $A$.

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

For any $v \in V$, we divide time into phases based on $A$’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 $\Phi_v$ 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 $\phi$, denoted by $\sigma(\phi)$, to be the value of $z_v$ at the end of $\phi$. For an external match $m$ of $A$ initiated by $v$, the phase of $v$ ending with $m$ is called the phase of $m$, denoted by $\phi_m$. For any round $\pi \in \Pi$, let $\Phi_\pi$ be the set of phases ending in $\pi$. For any round $\pi \in \Pi$, define phase_cost_time($\pi, A^*$) = $\sum_{v \in V} \sum_{\phi \in \Phi_\pi} C_{time}(v, \phi \setminus \Gamma, A^*)$, and phase_cost_space($\pi, A^*$) = $\sum_{v \in V} \sum_{\phi \in \Phi_\pi} C_{space}(v, \phi, A^*)$.

We say that a phase of $v$ is good, if the alignment status of $v$ does not change during the phase. Furthermore, a round $\pi$ is good if all the phases in $\Phi_\pi$ 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 $A$, while a round is complete if $A$ 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 $A$ and that of $A^*$ do not differ too much. This is reduced to two tasks. First, if all the counters are always small (say, no more than $4\delta$), the incremental cost of $A$ in every round is $O(\delta d)$, so it suffices to show that the cost of $A^*$ 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 $A^*$ 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 $A$, the cost of the optimum offline algorithm $A^*$ is not small.

Lemma 7. In every good complete round $\pi$, we have either round_cost_time($\pi, A^*$) $\geq f(f^{-1}(2\delta) - f^{-1}(\delta))$, or round_cost_space($\pi, A^*$) $\geq \delta$, or phase_cost_time($\pi, A^*$) $\geq \delta$. 
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.

**Lemma 8.** The number of bad rounds of $A$ is at most twice the number of external matches of $A^*$.

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 $A^*$.

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

The following technical lemmas will be needed.

**Lemma 10.** For any $c_1, \ldots, c_n \geq 0$ and $\alpha > 1$, we have

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

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

Now we are ready to prove the main result.

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

**Proof.** Suppose that $A$ 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 $\text{cost}_A(R) = \text{cost}_{A^*}(R) + \text{cost}_{A^*}(R) \leq 2kmd + \sum_{\phi \in \Phi} \sigma(\phi) \leq 22kmd + \sum_{\phi \in \Phi'} (\sigma(\phi) - 4\delta) \leq 22kmd + \sum_{\phi \in \Phi'} (\sigma(\phi) - 2\delta)$.

On the other hand, as to the cost of $A^*$, we have $\text{cost}_{A^*}(R) = \text{cost}_{A^*}(R) + \text{cost}_{A^*}(R) \geq \text{cost}_{A^*}(R) + \sum_{\pi \in \Pi} \text{phase}_{\text{cost time}}(\pi, A^*) + \sum_{\phi \in \Phi} \sigma'(\phi)$ by Lemma 9. Trivially we also have $\text{cost}_{A^*}(R) \geq \sum_{\pi \in \Pi} [\text{round}_{\text{cost time}}(\pi, A^*) + \text{round}_{\text{cost space}}(\pi, A^*)]$. 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 8, $A^*$ has at least $\frac{m''}{a''}$ external matches.

Hence,

$$
2\text{cost}_{A^*}(R) \geq \text{cost}_{A^*}(R) + \sum_{\pi \in \Pi} \text{phase}_{\text{cost time}}(\pi, A^*) + \sum_{\phi \in \Phi} \sigma'(\phi) \\
\geq \frac{m'}{a'} \delta + \sum_{\phi \in \Phi} \sigma'(\phi) \\
\geq \frac{m'}{a'} (\sqrt[2]{1 - \delta})^a \delta + \sum_{\phi \in \Phi} \sigma'(\phi)
$$

where the third equality is due to Lemma 7.

Altogether, $\frac{\text{cost}_{A^*}(R)}{\text{cost}_{A^*}(R)} \leq \frac{22kmd + \sum_{\phi \in \Phi'} (\sigma(\phi) - 2\delta)}{\frac{m'}{a'} (\sqrt[2]{1 - \delta})^a \delta + \sum_{\phi \in \Phi'} \sigma'(\phi)}$ 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 $m$ round, 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 $r$th 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 $\rho_{00}$, which arrives at time $T_r$, while a request $\rho_{ij}$ 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 $T_r + h\tau$, 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 $T_r + h\tau$, $\mathcal{A}$ has matched one request at $v_i$ and another at $v_{i'}$ both of which arrived during the period $[T_r, T_r + h\tau]$. 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. $\text{cost}_\mathcal{A}(R_r) \leq (\delta + \frac{k^2n}{\tau}f(\tau) + f(\tau))r$.

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 $\delta$. Dividing both sides of inequality by $m$ and letting $m$ approach infinity, by Lemmas 12 and 13, we get $f(n\tau) \leq (\delta + \frac{k^2}{2} f(\tau))a$, which means that $a \geq \frac{f(n\tau)}{\delta + \frac{k^2}{2} f(\tau)}$. Let $\tau$ approach zero. One has $\lim_{\tau \to 0} f(\tau) = 0$, and

$$\lim_{\tau \to 0} \frac{f(n \tau)}{\delta + \frac{k^2}{2} f(\tau)} = \lim_{\tau \to 0} \frac{1}{\delta + \frac{k^2}{2}} = +\infty \quad \text{since } f'(0) = 0.$$

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

### 6 Conclusion

We have designed an optimum deterministic online algorithm that solves $f$-MPMD for any monomial function $f(t) = t^a$ with $a > 1$. 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 $\max \left\{ \frac{120k^4}{f(1)^{228} - f(1)^{228}} \left\{ \frac{c - 25}{f(c)^{225} - f(1)^{225}} \right\}^{24}, c \leq 48 \right\}$, 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 concise cost functions, but to design a good one, i.e., with a very small competitive ratio, seems still challenging.

### References


A
Omitted Proofs in Section 4

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

**Lemma 14.** 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 \( h(x) - h(z - y) = \int_{(x,y,z)} h'(t)dt = \int_{(x,y,z)} h(t + z - x)dt \). By convexity of \( h \), \( h' \) is increasing, implying that \( h(x) - h(z - y) \geq \int_{(x,y,z)} h'(t)dt = h(x) - h(x - y) \). As a result, \( h(x - y) + h(z) - h(x) \geq h(z - y) \), which is exactly the desired inequality.

**Lemma 15.** Suppose that \( \rho_1, \ldots, \rho_n \in R \) 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 \( t \in (t(\rho_1), T(\rho_1)] \) and \( T \in (t(\rho_n), T(\rho_n)] \), 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 \( t \), \( f(t_1 - t(\rho_1)) = \int_{(t(\rho_1), t_1]} f(t - t(\rho_1))dt \leq \gamma \).

Because \( c_1 = \int_{(t_1, T_1]} f(t - t(\rho_1))dt = f(T_1 - t(\rho_1)) - f(t_1 - t(\rho_1)) \), \( T_1 - t_1 = f^{-1}(c_1 + x) - f^{-1}(x) \) where \( x = f(t_1 - t(\rho_1)) \). By convexity of \( f \), we have \( f^{-1}(c_1 + x) - f^{-1}(x) \geq f^{-1}(c_1 + \gamma) - f^{-1}(\gamma) \).

Then \( \sum_{i=1}^n f(T_i - t_i) \geq f(f^{-1}(c_1 + x) - f^{-1}(x)) \) and \( \sum_{i=1}^n f(f^{-1}(c_1 + x) - f^{-1}(x)) \geq f(f^{-1}(x) - f^{-1}(\gamma)) \), where the second inequality follows from Lemma 14.

**Corollary 16.** In a round \( \pi \), if a point \( v \) is aligned throughout a phase \( \phi \in \Phi_x \cap \Phi_v \), then \( \text{phase-cost}_{time}(\pi, A^*) \geq \min\{\sigma(\phi), 2\delta\} \).

**Proof.** Let \( \rho_1, \ldots, \rho_n \in R \) with \( T(\rho_i) < t(\rho_{i+1}) \) for any \( 1 \leq i < n \) be the requests at \( v \) that are successively pending during \( \pi \). Without loss of generality, assume that \( \sigma(\phi) \leq 2\delta \).

Since \( v \) is aligned throughout \( \phi \), \( A^* \) has requests \( \rho'_1, \ldots, \rho'_n \in R \) 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 15, \( \text{phase-cost}_{time}(\pi, A^*) \geq \sum_{i=1}^n f(T(\rho_i) - t(\rho_i)) \geq \sigma(\phi) \).

**Lemma 17.** In any good round \( \pi \), if \( A \) has an external match that is initiated by an aligned point, then \( \text{phase-cost}_{time}(\pi, A^*) \geq \delta \).

**Proof.** Arbitrarily choose an external match \( m \) in \( \pi \) that is initiated by an aligned point \( v \).

Since \( \pi \) is a good round, \( v \) is aligned throughout the phase \( \phi_m \). The lemma immediately follows from Corollary 16.

**Lemma 18.** In any good round \( \pi \), if \( \Psi \) has a misaligned point, then \( \text{phase-cost}_{time}(\pi, A^*) \geq \delta \) or \( \text{round-cost}_{space}(\pi, A^*) \geq \delta \).

**Proof.** Let \( v \) be the first misaligned point in \( \Psi \) during the round \( \pi \), namely, any points in \( \Psi \) is aligned before \( v \) gets misaligned, during the round \( \pi \). 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 \( m \) in \( \pi \) initiated by \( v \). Hence, before \( m \) occurs, \( v \) is aligned. Then \( \text{phase-cost}_{time}(\pi, A^*) \geq \delta \) by Lemma 17.

**Case 2:** \( v \) is aligned when it goes into \( \Psi \), but gets misaligned due to an external match of \( A^* \). Obviously, \( \text{round-cost}_{space}(\pi, A^*) \geq \delta \).
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 $v$ be the first misaligned point in $\Psi$ during this round. As a result, $\text{phase\_cost\_time}(\pi, A^*) \geq \delta$ by Lemma 17.

Roughly speaking, the next lemma claims that under some conditions, even if an external match between requests located in $\Psi$ and outside $\Psi$, the cost of $A^*$ must increase substantially.

Lemma 19. In any good round $\pi$, if there is an external match $m$ between requests located at $v \notin \Psi$ and $v' \notin \Psi$ such that $m$ is initiated by $v$ and $\phi_m \subseteq \pi$, then $\text{round\_cost\_time}(\pi, A^*) \geq f(f^{-1}(2\delta) - f^{-1}(\delta))$, $\text{round\_cost\_space}(\pi, A^*) \geq \delta$, or $\text{phase\_cost\_time}(\pi, A^*) \geq \delta$.

Basic idea of the proof: Since $m$ is between $v \notin \Psi$ and $v' \notin \Psi$ and initiated by $v$, it holds that $z_v \geq 2\delta$ when $m$ occurs. All we have to prove is that in the process that $z_v$, increases from $\delta$ to $2\delta$, whenever $A$ has a pending request $\rho$ at $A^*$ also has a request $\rho'$ that stays pending for a period no shorter than $\rho$ does. Then the proof ends due to Lemma 15.

Proof. If there exists a misaligned point in $\Psi$ during $\pi$, according to Lemma 18, the assertion follows. If $v$ is aligned in the phase $\phi_m$, according to lemma 17, the assertion also follows. The lemma also holds if $A^*$ has an external match during $\pi$.

The rest of the proof focuses on the other case, namely, all points in $\Psi$ are aligned throughout $\pi$, $v$ is misaligned in $\phi_m$, and $A^*$ has no external match during $\pi$. 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 $1 \leq i < n$, and $I_i = (a_i, b_i]$ for any $1 \leq i \leq n$. Then $\sum_{i=1}^{\infty} \int_{I_i} f(t - \ell(\rho_i))dt = 2\delta - \delta = \delta$.

Now we have three observations.

1. During each time interval $I_i$, no point outside $\Psi \cup \{v\}$ has pending request. Suppose there is a pending request $\rho'$ at $u \notin \Psi$ in $I_i$. Since $\delta \leq z_v \leq 2\delta$ and $A$ has a pending request $\rho$ at $v$ during $I_i$, $A$ should match $\rho$ and $\rho'$ in $I_i$, which is a contradiction.

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 1, among points outside $\Psi$, only $v$ has a pending request, which must get matched with $\rho$ due to the priority rule. This means that $m$ is between requests outside $\Psi$, 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 $\rho'$ is pending when $m'$ occurs. By Observation 1, when $m'$ occurs, $\rho'$ must be pending and $v \in \Psi$, 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 and $v$ is misaligned, at any moment in $\bigcup_{i=1}^{\infty} I_i$ 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 $I_i$. For any $1 \leq i \leq n$, let $u_i = \ell(\rho'_i)$.
Since each $\rho'_i$ is pending throughout $I_i$ and $f'$ is increasing, $C_{time}(u_i, I_i, A^*) \geq \int_{I_i} f'(t - t(\rho_i))dt \geq \int_{I_i} f'(t - a_i)dt = f(b_i - a_i)$.

Then, $\text{round}_{\text{cost}}(\pi, A^*) \geq \sum_{i=1}^{n} C_{time}(u_i, I_i, A^*) \geq \sum_{i=1}^{n} f(b_i - a_i) \geq f(f^{-1}(2\delta) - f^{-1}(\delta))$, where the last inequality follows from Lemma 15.

It is time to prove Lemma 7, stating that in every good complete round of $A$, the cost of the optimum offline algorithm $A^*$ is not small.

**Proof of Lemma 7.** Let $\mathcal{M}$ be the set of external matches $A$ outputs during $\pi$. By definition, $|\mathcal{M}| = 2k$. Let $\mathcal{M}' = \{m \in \mathcal{M} : m \text{ causes } |\Psi| \text{ to increase by one} \}$ and $\mathcal{M}'' = \mathcal{M} \setminus \mathcal{M}'$. Since any $m \in \mathcal{M}''$ does not change $|\Psi|$ and $|\Psi| \leq k - 1$, we have $|\mathcal{M}'| \leq k - 1$, which in turn implies $|\mathcal{M}''| \geq k + 1$. There must be a point $v \in V$ which initiates at least two external matches in $\mathcal{M}''$. Let $m \in \mathcal{M}''$ be the second external match in $\mathcal{M}''$ initiated by $v$. Obviously, the phase $\phi_m$ satisfies $\phi_m \leq \pi$. Now we proceed case by case.

**Case 1:** $v \in \Psi$ during $\phi_m$. If $v$ is aligned during $\phi_m$, we have $\text{phase}_{\text{cost}}(\pi, A^*) \geq \delta$ by Lemma 17. Otherwise, by Lemma 18, it holds that $\text{phase}_{\text{cost}}(\pi, A^*) \geq \delta$ or $\text{round}_{\text{cost}}(\pi, A^*) \geq \delta$.

**Case 2:** $v \notin \Psi$ during $\phi_m$. Assume $m = (\rho, \rho')$ and $v = t(\rho), u = t(\rho')$. Since $m \in \mathcal{M}''$, it must hold that $u \in \Psi$ when $m$ occurs. Applying Lemma 19, we finish the proof.

**Proof of Lemma 8.** An external match of $A^*$ changes the alignment status of at most two points, hence causing at most two bad phases, which in turn incur at most two bad rounds.

Recall $\Gamma = \{t : \text{ at time } t, A \text{ has a pending request } \rho \text{ with } z(t(\rho)) > 2\delta\}$. For any $v \in V$ and $\phi \in \Phi_v$, let $\Gamma_{\phi} = \{t \in \phi : \text{ at time } t, A \text{ has a pending request at } v \text{ with } z_v > 2\delta\}$. Obviously, $\Gamma = \bigcup_{\phi \in \Phi} \Gamma_{\phi}$ and all the $\Gamma_{\phi}$'s are pairwise disjoint. We now give a lower bound of the time cost of $A^*$ on every $\Gamma_{\phi}$.

**Lemma 20.** For any phase $\phi$ with $\sigma(\phi) > 2\delta$, $\sum_{u \in V} C_{time}(u, \Gamma_{\phi}, A^*) \geq f(f^{-1}(\sigma(\phi)) - f^{-1}(2\delta))$.

**Proof.** Basically, the proof is similar to that of Lemma 19.

Suppose that $\phi \in \Phi_v$ and $\Gamma_{\phi}$ 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 $v$ such that $t(\rho_i) = b_i$ for $1 \leq i \leq n$, $t(\rho_i) = a_i$ for $1 < i \leq n$, $t(\rho_1) \leq a_1$, and $\sum_{i=1}^{n} c_i = \sigma(\phi) - 2\delta$, where $c_i = \int_{I_i} f'(t - t(\rho_i))dt$ 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 at any time $a_i < t < b_i$. 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, $C_{time}(u_i, I_i, A^*) \geq \int_{I_i} f'(t - t(\rho'_i))dt \geq \int_{I_i} f'(t - a_i)dt = f(b_i - a_i)$.

By Lemma 15, $\text{round}_{\text{cost}}(\pi, A^*) \geq \sum_{i=1}^{n} C_{time}(u_i, I_i, A^*) \geq \sum_{i=1}^{n} f(b_i - a_i) \geq f(f^{-1}(\sigma(\phi)) - f^{-1}(2\delta))$.

**Proof of Lemma 9.** It is easy to see that

$$\text{cost}_{A'}(R) = \sum_{v \in V} \sum_{\phi \in \Phi_v} C_{time}(v, \phi, A^*) = \sum_{v \in V} \sum_{\phi \in \Phi_v} C_{time}(v, \phi \setminus \Gamma, A^*) + \sum_{v \in V} \sum_{\phi \in \Phi_v} C_{time}(v, \phi \cap \Gamma, A^*)$$
On the one hand, since $\Phi_v = \bigcup_{\pi \in \Pi} \Phi_v \bigcap \Phi_\pi$,

$$\sum_{v \in V} \sum_{\phi \in \Phi_v} C_{time}(v, \phi \mid \Gamma, A^*) = \sum_{v \in V} \sum_{\phi \in \Phi_v} \sum_{\pi \in \Pi} C_{time}(v, \phi \mid \Gamma, \pi, A^*)$$

$$= \sum_{\pi \in \Pi} \sum_{v \in V} \sum_{\phi \in \Phi_v} C_{time}(v, \phi \mid \Gamma, \pi, A^*)$$

$$= \sum_{\pi \in \Pi} \text{phase}_{\text{cost}}(\pi, A^*).$$

On the other hand,

$$\sum_{v \in V} \sum_{\phi \in \Phi_v} C_{time}(v, \phi \cap \Gamma, A^*) = \sum_{v \in V} C_{time}(v, \Gamma, A^*)$$

$$= \sum_{\phi \in \Phi} \sum_{v \in V} C_{time}(v, \Gamma, \phi, A^*)$$

$$= \sum_{\phi \in \Phi} \sum_{v \in V} \sum_{\phi \in \Phi_v} C_{time}(v, \Gamma, \phi, A^*) \geq \sum_{\phi \in \Phi} \sigma'(\phi),$$

where the third equality is because the $\Gamma_\phi$‘s are pairwise disjoint, and the inequality follows from Lemma 20.

 Altogether, we finish the proof. □

Proof of Lemma 10. It suffices to prove that $\frac{a-b}{(\sqrt{a} - \sqrt{b})^2}$ decreases with $a$ when $a > b$. This is equivalent to showing $g(x) = \frac{x^n - y^n}{(x-y)^n}$ decrease with $x$ when $x > y$. The claim holds since $g'(x) = \alpha \frac{y^{n-1}x^{n-1}}{(x-y)^{n+1}} \leq 0$. □

Proof of Lemma 11. Denote the round of $A$ by $\pi$. We proceed case by case.

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

Case 2: $A$ has no external matches while $A^*$ has. For any $v \in V$, let $c_v = \sigma(\phi_v)$ where $\phi_v$ is the unique phase of $v$. We have $\text{cost}_A(R) = \sum_{v \in V} c_v$. On the other hand,

$$\text{cost}_{A^*}(R) = \text{cost}_{A^*}(R) + \text{cost}_{A^*}^\prime(R) \geq \delta + \sum_{v \in V} c'_v$$

with $c'_v = \sigma'(\phi_v)$, where the inequality is due to Lemma 9 and the assumption that $A$ has external matches. Let $V'_v = \{v \in V : c_v \geq 4\delta\}$. Then $\frac{\text{cost}_A(R)}{\text{cost}_{A^*}(R)} \leq \frac{4k\delta + \sum_{v \in V} (c_v - 2\delta)}{\delta + \sum_{v \in V} c'_v}$. By Lemma 10, $\frac{\text{cost}_A(R)}{\text{cost}_{A^*}(R)} = O(k)$.

Case 3: $A$ has external matches. If $A^*$ has no external matches, the first external match $m$ of $A$ must be initiated by a point that is aligned throughout the phase $\phi_m$. Since $\sigma(\phi_m) \geq \delta$, we have $\text{round}_{\text{cost}}(\pi, A^*) \geq \delta$ by Corollary 16. As a result, either $\text{cost}_{A^*}^\prime(R) \geq \delta$ or $\text{round}_{\text{cost}}(\pi, A^*) \geq \delta$.

On the one hand, $A$ has at most $2k$ external matches in a round, so $\text{cost}_A(R) \leq 2k\delta + \sum_{\phi \in \Phi} \sigma(\phi)$. Let $\Phi' = \{\phi \in \Phi : \sigma(\phi) \geq 4\delta\}$. Because there are at most $4k$ complete phases and $k$ incomplete ones, $|\Phi| \leq 5k$, which implies that $\text{cost}_A(R) \leq 22k\delta + \sum_{\phi \in \Phi'} (\sigma(\phi) - 2\delta)$.

On the other hand, as to the cost of $A^*$, we have $\text{cost}_{A^*}(R) = \text{cost}_{A^*}^\prime(R) + \text{cost}_{A^*}^\prime(R) \geq \text{cost}_{A^*}^\prime(R) + \text{round}_{\text{cost}}(\pi, A^*) + \sum_{\phi \in \Phi} \sigma(\phi) \geq \delta + \sum_{\phi \in \Phi} \sigma(\phi)$, where the first inequality follows from Lemma 9.

Hence $\frac{\text{cost}_A(R)}{\text{cost}_{A^*}(R)} \leq \frac{22k\delta + \sum_{\phi \in \Phi'} (\sigma(\phi) - 2\delta)}{\delta + \sum_{\phi \in \Phi} \sigma(\phi)}$. By Lemma 10, $\frac{\text{cost}_A(R)}{\text{cost}_{A^*}(R)} = O(k)$. □
B Omitted Proofs in Section 5

Proof of Lemma 12. It suffices to show that the cost that $A^*$ pays for any round is at most $\delta + \frac{k\tau_n}{2} f(\tau) + f(\tau))$. 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: $\langle \rho_0, \rho_k \rangle$, $\langle \rho_{ij}, \rho_{i,j+1} \rangle$ for odd numbers $j \geq 1$ and $1 \leq i \leq k - 1$, and $\langle \rho_{kj}, \rho_{k,j+1} \rangle$ for even numbers $j \geq 2$. Since $A^*$ is an optimum offline algorithm, its cost is at most the cost of this matching.

Proof of Lemma 13. By Fact 4, it is equivalent to show that the cost that $A$ pays for requests in $R_r \setminus R_{r-1}$ is at least $k\delta$.

On the one hand, assume Case 2 in this round does happen. We have three observations:

- After time $(h_r - 1)T$, no request arrives at any $v \in C_{h_r} = C_{h_r-1}$.
- The total number of requests that have arrived at $C_{h_r}$ 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 $\ell(\rho') \notin C_{h_r}$.
- The request $\rho$ is pending throughout the interval $(T_r + ((h_r - 1)T, T_r + hT)]$, incurring time cost at least $f(T) = k\delta$.

On the other hand, assume that Case 1 happens, namely, $C_{h_r} = V$. Then $A$ has at least $k$ external matches in this round.

Altogether, the cost $A$ pays for this round is at least $k\delta$. ▶