Online Matching with Delays and Stochastic Arrival Times

Mathieu Mari
University of Warsaw & IDEAS NCBR
Warsaw, Poland
m.mari@mimuw.edu.pl

Michał Pawłowski
University of Warsaw & IDEAS NCBR
Warsaw, Poland
m.pawlowski@mimuw.edu.pl

Runtian Ren
University of Warsaw & IDEAS NCBR
Warsaw, Poland
r.ren@mimuw.edu.pl

Piotr Sankowski
University of Warsaw, IDEAS NCBR & MIM Solutions
Warsaw, Poland
sank@mimuw.edu.pl

ABSTRACT

Consider a platform where independent agents arrive at random times and need to be matched into pairs, eventually after waiting for some time. This, for example, models job markets, gaming platforms, kidney exchange programs, etc. The role of the platform is to decide how to match agents together while optimizing two conflicting objectives: the quality of the matching produced, and the total waiting time of the agents. This can be modeled as an online problem called Min-cost Perfect Matching with Delays (MPMD), which has recently drawn a lot of attention. It is known that in the case when agents arrive in an adversarial order, no online algorithm can achieve a constant-competitive ratio. In this paper, we study the more realistic case where agents' arrival times follow some stochastic assumptions, and we present two matching mechanisms, which give constant-competitive solutions. The first one is a simple greedy algorithm in which agents act in a distributed manner requiring only local communication. The second one builds global analysis tools in order to obtain even better performance guarantees. This result is rather surprising as the greedy approach cannot achieve a competitive ratio better than $O(m^{\log 1.5+\varepsilon})$ in the adversarial model, where m denotes the number of agents. Finally, we extend our results to the case where the delay cost corresponds to an arbitrary positive and non-decreasing function of the waiting time, as well as the case where the platform is allowed to pay a penalty cost to clear some agents' requests.

KEYWORDS

online algorithms; matchings; stochastic model; Poisson arrivals

ACM Reference Format:
Mathieu Mari, Michał Pawłowski, Runtian Ren, and Piotr Sankowski. 2023. Online Matching with Delays and Stochastic Arrival Times. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 9 pages.

1 INTRODUCTION

Imagine players logging into an online platform to compete against each other in a two-player game. The platform needs to pair them up in a way that maximizes the overall satisfaction from the gameplay. Since each player prefers to be matched with someone with similar gaming skills, the platform has to consider the experience gap when pairing players. This skill level difference is referred to as the *connection cost*. Additionally, once logged in, a player can tolerate some waiting time to be matched — this is why the platform can postpone the pairing decision in the hope of a better matching to be found (i.e., the login of another player with similar skills). Nonetheless, the waiting time for each player has its limits. A player may become impatient if its request has been ignored for too long time. This time gap between logging into the platform and joining a gaming session is referred to as the *delay cost*. The platform's goal is to pair all the online players into sessions, such that the total connection cost plus the total delay cost produced is minimized.

This problem can be modeled as a special case of an online problem called Min-cost Perfect Matching with Delays (MPMD) defined by Emek et al. [30]. MPMD has drawn researchers' attention recently [2, 3, 7, 8, 16, 17, 30, 43] due to many real-life applications ranging from Uber rides, dating platforms, kidney exchange programs etc. Formally, MPMD is defined as follows. The input is a set of m requests (each representing an independent agent) arriving at arbitrary times in a metric space $\mathcal{M} = (\mathcal{X}, d)$ equipped with a distance function d. Here, m is an even integer, and $\mathcal{X}$ denotes the set of points in $\mathcal{M}$. Each request r is characterized by its *location* $\ell(r) \in \mathcal{X}$ and *arrival time* $t(r) \in \mathbb{R}^+$. When two requests r and r' are matched into a pair at time $t \geq \max\{t(r), t(r')\}$, a *connection cost* $d(\ell(r), \ell(r'))$ plus a *delay cost* $(t - t(r)) + (t - t(r'))$ is incurred. The target is to minimize the total cost produced by the online algorithm for matching all the requests into pairs.

Previously, the MPMD problem was studied in the case where the requests are generated by an adversary. Unfortunately, no online algorithm can achieve a constant competitive ratio in this adversarial model [2]. It is often too pessimistic to assume no stochastic information on the input is available — again, consider the example of matching players on gaming platforms. The online platform has all the historical data and can estimate the arrival frequency of the players with each particular skill level. Therefore, it is reasonable to assume that the gaming requests follow some stochastic distribution. Depending on the time of day, though, there may be more or fewer players logging in. However, if we divide the timeline into small intervals, it is reasonable to assume that within each of them, the distribution is regular and the requests are mutually independent (since the players don't know each other). Based on these observations, the following question can be naturally stated: *in the case when stochastic information on the input is available, can we devise online algorithms with constant performance guarantees?*

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

Here, we provide an affirmative answer to this question. We achieve this by providing a natural greedy algorithm that matches players based on local information. We present a novel analysis of this algorithm for a stochastic online version of MPMD, by assuming that the requests arrive following a Poisson arrival process. To be more precise, the waiting time between any two consecutive requests arriving at any metrical point x, follows an exponential distribution $\mathrm{Exp}(\lambda_x)$ with parameter $\lambda_x \geq 0$. Under such a model, the goal of the platform is to minimize the expected cost produced by an algorithm ALG to deal with a random input sequence consisting of m requests. To evaluate the performance of our algorithms on stochastic inputs, we use the *ratio-of-expectations*, which corresponds to the ratio of the expected cost of the algorithm to the expected cost of the optimal offline solution (see Definition 4).

Our Contribution. We prove that in the Poisson arrival model, we can obtain a significantly better performance guarantee compared with the current best competitiveness obtained in the adversarial model. More specifically, we show that the *Greedy* algorithm, which matches any two requests immediately when their total delay cost reaches their distance, achieves a constant ratio-of-expectations.

Theorem 1. *For MPMD in the Poisson arrival model, the Greedy algorithm achieves a ratio-of-expectations of* $16/(1-e^{-2})$.

A notable advantage of this algorithm is that it can be implemented in a distributed manner. Indeed, the decision taken by two agents depends only on their local information, i.e., their waiting time and the distance between them. Moreover, it is worth emphasizing that in the adversarial model, the greedy algorithm has a competitive ratio of $\Omega(m^{\log 1.5+\varepsilon})$ (see the example in [7], Appendix A).

To prove Theorem 1, we apply the following strategy. We first notice that the connection cost of a Greedy solution is at most its delay cost. Thus, it becomes the core of the proof to upper bound the delay cost. For this purpose, in Section 3, we define the *radius* $\rho_x \geq 0$ for each metric point x. Such a radius depends on the parameters of the problem and roughly corresponds to the expected delay time for matching the requests located on x. Then, we show how to use the radius to lower bound the cost of the optimal offline solution. Intuitively, we prove that a request located on x is in expectation responsible for a total cost of $\Omega(\rho_x)$.

The notion of radius suggests another algorithm for MPMD with stochastic inputs. Indeed, when a new request r arrives on a point x, we know that this request will wait for a time $O(\rho_x)$ in average before being matched by the Greedy algorithm. In particular, r will be matched with another request that is at distance $O(\rho_x)$. Therefore, if at the time of the r's arrival, there is another pending[1] request r' that is at distance less than ρ_x, why not matching these two requests directly? In Section 3, we formalize this intuition and design an algorithm called *Radius*. Thanks to these anticipated pairings, the performance ratio is improved by a factor of 2.

Theorem 2. *For MPMD in the Poisson arrival model, the Radius algorithm achieves a ratio-of-expectations of* $8/(1-e^{-2})$.

Finally, we show how to adjust the Greedy and the Radius algorithms to deal with other variants of the MPMD problem, while preserving a constant performance ratio. In Section 7, we look at the generalization of the problem where a request can be delayed for a time t at a cost $f(t)$, where f is a positive and non-decreasing function. We show that, unless f is such that the expected cost of the optimal offline solution is infinite, our algorithms achieve constant performance ratios, where the constants only depend on function f. In Section 8, we consider the variant of MPMD that allows clearing pending requests for a fixed penalty cost.

Related Work. The MPMD problem was introduced by Emek et al. [30]. In their paper, they proposed a randomized online algorithm that achieves a competitive ratio of $O(\log^2 n + \log \Delta)$, where n is the number of points of the metric space and Δ is the aspect ratio. Later, Azar et al. [3] improved the competitive ratio to $O(\log n)$, thereby removing the dependence of Δ in the competitive ratio. Both of these papers randomly embed the metric space into a tree of distortion $O(\log n)$, and then propose online algorithms on tree metrics. In the adversarial model, this bound is essentially tight, since Ashlagi et al. [2] showed that any randomized algorithm achieves a competitive ratio of $\Omega(\log n/\log\log n)$. Note that the above results assume that the n-point metric is given in advance. When the metric is not known in advance, Bienkowski et al. proposed a $O(m^{2.46})$-competitive online greedy algorithm [17] and a $O(m)$-competitive online algorithm based on the primal-dual method [16], where m denotes the number of requests released. Azar and Jacob-Fanani [7] later proposed a $O(m^{\log 1.5+\varepsilon})$-competitive greedy algorithm, which is currently the best deterministic online algorithm. Emek et al. [31] proposed a 3-competitive greedy algorithm for a two-point metric case. Deryckere and Umboh [27] studied the set delay case where the delay cost at any time is an arbitrary function of the set of pending requests. Another line of work considered a bipartite variant of MPMD, i.e., the Min-cost Bipartite Perfect Matching with (linear) Delays (MBPMD), where each request can be either red or blue, and only two requests of different colors can be matched into a pair. For MBPMD, the current best online algorithm achieves a competitive ratio of $O(\log n)$ and the lower bound on the competitiveness is $\Omega(\sqrt{\log n/\log\log n})$ [2]. Further, both MPMD and MBPMD problems have been investigated in the more general case when any request can be delayed for a duration t at a cost $f(t)$, with $f(\cdot)$ being convex [43] or concave [8].

Besides MPMD, many other online problems have been also considered with the additional delay constraints, such as online service with delay [6, 9, 18], multi-level aggregation [9, 13, 14, 22, 23, 42], facility location [9, 10, 15], bin packing [5, 32], set cover [4, 42, 48] and many others [10, 24, 35, 46, 48].

We remark that matching is a huge topic, drawing attentions from both theory and real applications perspectives since Edmonds [28, 29] and Karp et al. [38]. In recent years, motivated by job market, kidney exchanges etc, many other online matching results have also been conducted, e.g., [11, 19–21, 25, 33, 34, 37, 39, 40, 44, 47, 49]. Different from MPMD, these works assume that the matching decision must be made immediately at request arrival. One another similar stochastic online matching problem assumes that requests are released with Poisson arrival and Poisson departures [1, 12, 26, 36, 41]. However, the target is to maximize the total value of the matching pairs produced. To the best of our knowledge, we are the first to consider MPMD in the stochastic arrival model.

[1] By pending we mean that at that time, the request is still unmatched by the algorithm.

2 PRELIMINARIES

Problem Statement. A *metric space* $\mathcal{M} = (\mathcal{X}, d)$ is a set of points $\mathcal{X}$ equipped with a distance function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ that satisfies the triangle inequality. The input consists of a sequence σ of m requests (m being an even integer), where each request $r \in \sigma$ is characterised by its *location* $\ell(r) \in \mathcal{X}$ and *arrival time* $t(r) \in \mathbb{R}^+$ (w.l.o.g., suppose that no two requests arrive at the same time). Now, given any solution for an input sequence σ, let M denote the set of paired requests (i.e., the perfect matching generated for σ), and let $s(r) \geq t(r)$ denote the moment when a request r is matched. Note that if r and r' are matched into a pair, i.e., $(r, r') \in M$, we have $s(r) = s(r')$. Using this notation, the total cost of a solution (M, s) is the sum of its *delay cost* and its *connection cost* defined as follows. The delay cost produced by the solution is the sum of the delay costs $s(r) - t(r)$ incurred for each request r. Similarly, the connection cost is the sum of distances between all the paired requests, i.e., $\sum_{(r,r') \in M} d(\ell(r), \ell(r'))$.

Let $\mathrm{OPT}(\sigma)$ denote the minimum cost of a feasible solution for σ. Notice that it corresponds to a minimum weight perfect matching for σ, where the weight of an edge $(r, r') \in \sigma \times \sigma$ is given by $d(\ell(r), \ell(r')) + |t(r) - t(r')|$. For any pair (r, r') produced by the optimal solution it holds that $s(r) = s(r') = \max\{t(r), t(r')\}$, which implies that $\mathrm{OPT}(\sigma)$ can be computed in poly(m) time. In this paper, we are interested in the design of *online* algorithms for the problem: the decision of matching a pair (r, r') at time t only depends on $\{r \in \sigma : t(r) \leq t\}$, and this decision is irrevocable.

Stochastic Model. In the stochastic version of MPMD, the goal is to design an online algorithm that processes a sequence of requests arriving at "random moments", instead of being generated by an online adversary. To formalize the notion of random arrival times, we use the Poisson arrival process: given any point $x \in \mathcal{X}$, we assume that the requests arrive at x with a Poisson arrival rate equal to some $\lambda_x > 0$. Recall that an exponential variable $X \sim \mathrm{Exp}(\lambda)$ with parameter $\lambda > 0$ has a probability density function $f_\lambda(t) = \lambda e^{-\lambda t}$ for $t \geq 0$ and expectation $\mathbb{E}[X] = 1/\lambda$. The exponential distribution may be viewed as a continuous counterpart of the geometric distribution, which describes the number of Bernoulli trials necessary for a discrete process to change state (here, observing a new request on a given point).

Definition 3 (distributed Poisson arrival model). *A (random)* requests sequence *σ follows distributed Poisson arrival model if the* waiting time *between any two consecutive requests arriving at the same point $x \in \mathcal{X}$ follows an* exponential distribution *with parameter $\lambda_x > 0$ and the variables representing waiting times are mutually independent.*

From now on, when we say that σ is *a random request sequence of length m*, for some integer m, we mean that σ consists of m requests, and that their arrival times follow the distributed Poisson arrival model. In this context we measure the performance of our algorithms using the *ratio-of-expectations*:

Definition 4. *We say that an algorithm* ALG *for MPMD has a ratio-of-expectations $C \geq 1$, if*

$$\varlimsup_{m \to \infty} \frac{\mathbb{E}_\sigma^m[\mathrm{ALG}(\sigma)]}{\mathbb{E}_\sigma^m[\mathrm{OPT}(\sigma)]} \leq C,$$

where $\mathrm{ALG}(\sigma)$ *(resp.* $\mathrm{OPT}(\sigma)$*) denotes the cost produced by* ALG *(resp. an optimal offline solution) on the request sequence σ, and* $\mathbb{E}_\sigma^m[\mathrm{ALG}(\sigma)]$ *(resp.* $\mathbb{E}_\sigma^m[\mathrm{OPT}(\sigma)]$*) denotes the expected cost of* ALG *(resp.* OPT*) on a random input sequence σ consisting of m requests generated by the Poisson arrival process.*

The rest of this section is devoted to presenting the standard Poisson arrival model and showing that it is equivalent to the one we defined earlier. This will allow us to use them interchangeably since, depending on the context, it might be easier to consider one or the other. However, in order to distinguish between them, we call the second model centralized.

Definition 5 (centralized Poisson arrival model). *A (random)* requests sequence *σ follows the centralized Poisson arrival model if the* waiting time *between any two consecutive requests in the given metric space follows an* exponential distribution *with parameter $\lambda(\mathcal{X}) := \sum_{x \in \mathcal{X}} \lambda_x$ and each time a request arrives, the probability of it appearing at point x equals $\lambda_x / \lambda(\mathcal{X})$. We assume that the waiting times and requests location choices are all mutually independent.*

To show the equivalence between the two processes, we exploit the two well-known properties of the exponential distribution.

Proposition 6 (memoryless property)**.** *If X is an exponential variable with parameter λ, then for all $s, t \geq 0$, we have*

$$\mathbb{P}(X > s + t \mid X > s) = \mathbb{P}(X > t) = e^{-\lambda t}.$$

Proposition 7. *Given n independent exponential variables $X_i \sim \mathrm{Exp}(\lambda_i)$ for $i \in \{1, 2, \ldots, n\}$, let $Z := \min\{X_1, X_2, \ldots, X_n\}$ and let $\lambda := \sum_{i=1}^n \lambda_i$. It holds that*

1. $Z \sim \mathrm{Exp}(\lambda)$, 2. $\mathbb{P}(Z = X_i) = \frac{\lambda_i}{\lambda}$, 3. $Z \perp \{Z = X_i\}$, where $\perp$ denotes independence.

Let us now consider the distributed model where for each point $x \in \mathcal{X}$, we define an exponential variable Y_x^1 representing the time of arrival of the first request located at x. Then, if we look at the whole metric space, the time of arrival of the first request r is determined by the minimum of all these variables, $\min_{x \in \mathcal{X}} Y_x^1$. We denote this variable by Y_1. By Proposition 7, we know that Y_1 follows an exponential distribution with parameter λ being the sum of components' parameters. Moreover, by the second property presented in this proposition, we know that the probability of r arriving at point x equals λ_x / λ for each $x \in \mathcal{X}$.

Finally, at time $t(r) = Y_1$ we associate each point $x \neq \ell(r)$ with a new independent exponential random variable $Z_x^1 \sim \mathrm{Exp}(\lambda_x)$. By the memoryless property from Proposition 6, we get that for each x the arrival time determined by $t(r) + Z_x^1$ follows the same distribution as Y_x^1 conditioned on being greater than $t(r)$. This shows that we can look at the first request arrival as it was defined by the centralized model and the consequent requests still follow the distributed model. Of course, we can also continue this process for them to transform the distributed model into a centralized one.

To get a better understanding of what is the relation between the two models, see Figure 1. Notice that, since both are equivalent, it gives us another way of looking at the stochastic process we work with — it is sufficient to define an arrival rate for the whole metric space and adjust the requests' appearance distribution over the points.

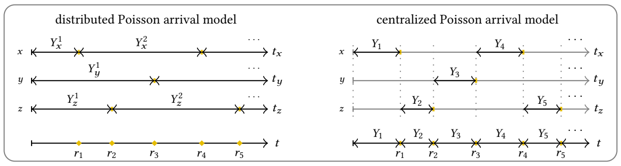
distributed Poisson arrival model
centralized Poisson arrival model
x
y
z
Y_x^1
Y_x^2
Y_y^1
Y_z^1
Y_z^2
t_x
t_y
t_z
t
r_1 r_2 r_3 r_4 r_5
Y_1 Y_2 Y_3 Y_4 Y_5
. . .

Figure 1: Example on the correspondence between the distributed and centralized Poisson arrival model.

3 CONSTANT COMPETITIVE ALGORITHMS

In this section, we introduce two deterministic online algorithms for the MPMD problem: Greedy and Radius. We formally define the radius of each metric point which is used to design the Radius algorithm. We present the upper bounds on the expected cost of our algorithms (Lemmas 11 and 12) and the lower bound on the expected cost of the optimal offline solution (Lemma 13). We give an overview of the techniques used to obtain these bounds. Finally, with these Lemmas, we prove Theorems 1 and 2.

3.1 The Greedy Algorithm

First, we present a simple greedy algorithm: once the total waiting time of any two pending requests exceeds the distance between them, it matches them into one pair. It is easy to show that this algorithm is well-defined: since the metric space $\mathcal{M}$ is bounded (as it contains a finite number of points), the waiting time of the last request is bounded by the diameter of $\mathcal{M}$; together with the assumption that the input sequence σ has an even number of requests, Greedy thus outputs a perfect matching on σ. Notice that this algorithm works more generally in the online adversarial model, and additionally that it does not require knowing the metric space or the exponential parameters in advance. For a formal description of Greedy, see the pseudo-code below.

```
Algorithm 1: Greedy
  Input: A sequence $\sigma$ of requests.
  Output: A perfect matching of the requests.
  for any time $t$ do
      if there exist pending requests $r, r'$ such that
        $(t - t(r)) + (t - t(r')) \geq d(\ell(r), \ell(r'))$ then
          match them into a pair with ties broken arbitrarily.
```

To better understand the algorithm, we can look at its geometrical interpretation. Here, when a request r appears at some point x, a ball centred at x starts growing with a uniform rate as time passes by. The radius of this ball represents the delay cost incurred due to leaving r unmatched. Hence, once two balls intersect, the pending requests located at their centres are paired (see Figure 2).

The remaining part of this subsection presents a sketch of how to prove the constant ratio-of-expectations for Greedy (Theorem 1). First, we observe that for each request served by this algorithm, its connection cost does not exceed its delay cost. Thus, if we find the upper bound for the latter, we will be able to estimate the total expected cost of the matching generated by Greedy on a request sequence σ. To do so, let us focus on finding the expected delay cost of a single request r arriving at some point $x \in \mathcal{X}$. We say that it is matched with a close request if the distance between them is bounded by some threshold ρ_x that we will refer to as a radius. For now, it suffices to know that this value depends on the arrival location x of r and will be defined later. To introduce formally the radius, we use the following notation for closed and open balls.

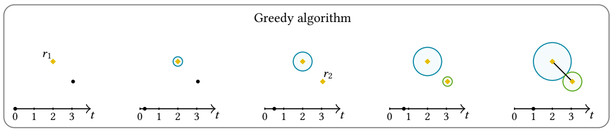

Figure 2: An example of how Greedy works on a sequence of two requests arriving at times 0 and 0.5 in a 2-point metric space with the distance between the points equal to 1.5.

Definition 8. *For each point $x \in \mathcal{X}$, let $\overline{B}(x, u)$ (resp. $B^\circ(x, u)$) denote the set of metric points $y \in \mathcal{X}$ with a distance no more than (resp. strictly less than) u from x.*

The next part of the analysis heavily depends on whether there exists a request arriving after r at any point in $\overline{B}(x, \rho_x)$ or not. When the latter happens, we call r a late request and upper bound the cost of serving it by the highest value possible — the sum of the metric space diameter and the expected waiting time for the next request to arrive. Although the estimation may seem exaggerated, it can be proved that only a few such requests exist. For the first case, when a close request arrives after r, with the right choice of ρ_x, the expected cost of serving r can be upper bounded by a constant times the radius. We define radius as follows. For any subset of points $\mathcal{S} \subseteq \mathcal{X}$, we denote $\lambda(\mathcal{S}) := \sum_{x \in \mathcal{S}} \lambda_x$.

Definition 9. *For each point $x \in \mathcal{X}$, define its radius ρ_x as the minimum value $u \geq 0$, such that $\frac{1}{\lambda(\overline{B}(x,u))} \leq u$.*

The idea behind is that it balances the relationship between the diameter of $\overline{B}(x, u)$ and the expected waiting time between consecutive request arrivals within the points of this ball. Indeed, using the information from the preliminaries, one can show that the latter is equal to the left-hand side of the inequality. Finally, we note that the radius is well-defined as the function $u \mapsto 1/\lambda(\overline{B}(x, u))$ is non-increasing and left-continuous. Thus, the minimal value satisfying the given inequality exists and belongs to $(0, 1/\lambda_x]$. See Figure 3 for a pictorial example.

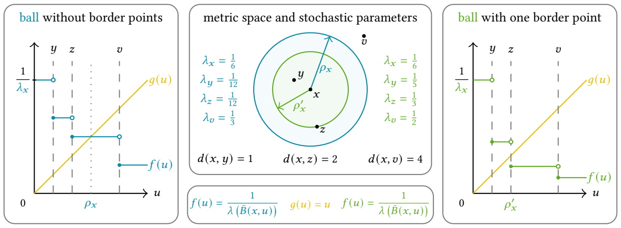

Figure 3: When determining the radius for some point x, two cases may occur. First, the plots of $f(u) = 1/\lambda(\overline{B}(x, u))$ and $g(u) = u$ may intersect explicitly (graph on the left). Second, the value of $f(u)$ may drop below $g(u)$ when approaching some point on the border of the ball (graph on the right).

By the radius definition, we have the following observation.

Observation 10. *Given any point $x \in \mathcal{X}$,*

$$\frac{1}{\lambda(B^\circ(x, \rho_x))} \geq \rho_x \geq \frac{1}{\lambda(\overline{B}(x, \rho_x))}.$$

Here, we present both the upper and the lower bound on the radius, as one of them is needed to lower bound the expected cost of the optimal offline solution, and the second one is required to upper bound the expected cost of our algorithms. To conclude, let us state the upper bound on the expected cost produced by Greedy.

Lemma 11. *For MPMD in the Poisson arrival model, the expected cost produced by the Greedy algorithm, over all random sequences consisting of m requests, satisfies*

$$\mathbb{E}^m_\sigma[\mathrm{Greedy}(\sigma)] \leq \left(4m \sum_{x \in \mathcal{X}} \frac{\lambda_x}{\lambda(\mathcal{X})} \cdot \rho_x\right) + 2|\mathcal{X}| \cdot \left(d_{\max} + \frac{1}{\lambda(\mathcal{X})}\right),$$

where $d_{\max} := \max_{x,y \in \mathcal{X}} d(x, y)$ is the diameter of the metric space.

The last term of the right-hand side describes the cost of serving the late requests. The first term represents the standard expected cost of serving requests and is proportional to the length of the sequence σ. We prove this lemma in Section 5.

3.2 The Radius Algorithm

In this subsection, our goal is to improve the performance guarantees of the Greedy algorithm on stochastic inputs. For this purpose, we design a Radius algorithm that calculates the radii upfront and uses this information to serve the requests better. The main idea here is to match any two requests whenever the closed balls of their locations (with radii defined as in Definition 9) overlap.

In the geometrical interpretation, whenever a request arrives at some point x, the algorithm directly sets its ball to be $\overline{B}(x, \rho_x)$. Hence, once a request r appears, if its location belongs to the closed ball of any pending request r', then the two are matched[2]. Otherwise, if there exists another request r'' within the distance of $\rho_{\ell(r)} + \rho_{\ell(r'')}$ from r's location, r can be matched with any such request. Finally, if no request satisfies the above conditions, r is temporarily left unmatched. See the pseudo-code shown in Algorithm 2 for a precise description of Radius. Notice that since Radius calculates the radii, it needs to know the metric space $(\mathcal{X}, d)$ and the exponential parameters $\{\lambda_x\}_{x \in \mathcal{X}}$. This is not a heavy requirement, since in the case of stochastic inputs, by the Law of large numbers, one can learn in constant time $O(1/\min_{x \in \mathcal{X}} \lambda_x)$ an arbitrarily good estimate of the arrival rates.

It turns out that using the radius information directly in the algorithm leads to a better ratio-of-expectations. Below we present an upper bound on the expected cost of the Radius solution.

Lemma 12. *For MPMD in the Poisson arrival model, the expected cost produced by the Radius algorithm, over all random sequences consisting of m requests, satisfies*

$$\mathbb{E}^m_\sigma[\mathrm{Radius}(\sigma)] \leq \left(2m \sum_{x \in \mathcal{X}} \frac{\lambda_x}{\lambda(\mathcal{X})} \cdot \rho_x\right) + \frac{1}{2} \cdot |\mathcal{X}| \cdot d_{\max},$$

where $d_{\max} := \max_{x,y \in \mathcal{X}} d(x, y)$ is the diameter of the metric space.

Algorithm 2: Radius

```
Input: A sequence $\sigma = (r_1, \dots, r_m)$ of requests, the arrival
       rate of each metric point.
Output: A perfect matching of the requests.
Compute the radius $\rho_x$ for each point $x \in \mathcal{X}$ (Definition 9);
$P \longleftarrow$ the set of pending requests, initially empty;
for $i = 1$ to $m$ do
    let $t = t(r_i)$ denote the arrival time of the $i$-th request $r_i$;
    if there exists a pending request $r' \in P$ such that
      $d(\ell(r_i), \ell(r')) \leq \rho_{\ell(r')}$ then
        match $r_i$ and $r'$ together, and remove $r'$ from $P$.
    else if there exists a pending request $r' \in P$ such that
      $d(\ell(r_i), \ell(r')) \leq \rho_{\ell(r')} + \rho_{\ell(r_i)}$ then
        match $r_i$ and $r'$ together, breaking ties arbitrarily,
          and remove $r'$ from $P$.
    else
        add $r_i$ in $P$.
if $P \neq \emptyset$ then
    match all requests in $P$ arbitrarily.
```

The formal proof of this lemma can be found in Section 6. Here, we conclude the algorithm description with an example illustrated in Figure 4.

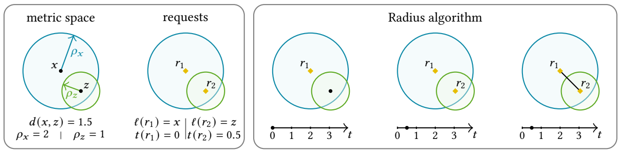

Figure 4: An example of how Radius works for two requests.

3.3 Lower Bounding OPT

It remains to present an overview of the lower bounding scheme for the optimal offline solution of the MPMD problem. Having such a result will enable us to find the performance ratio for the two algorithms introduced before and show that they both achieve constant ratio-of-expectations.

The crucial part of the lower bounding process is to analyze each request r in a sequence σ separately and observe that two situations can happen when r is not matched immediately. On one hand, r can be matched early with some distant request, thus, paying a high connection cost. On the other hand, it can wait for a closer request to arrive and pay a higher delay cost. A similar situation takes place when r is paired at the moment of its arrival with an older request. The only difference then is that we go through the timeline in the opposite direction.

Let us set the threshold for a request to be considered close to r as the radius of r's arrival location, i.e., $\rho_{\ell(r)}$. Then, the expected cost of serving r can be upper bounded by the expected value of the minimum of three things. The first one is the cost of matching r with the latest request that has arrived in $B^\circ(x, \rho_x)$ before r. The second is equal to the cost of matching r with the earliest request

[2] Notice that there exists at most one such request r'. Otherwise, if at the moment of its arrival, r belonged to the closed balls of two requests r' and r'', their balls would intersect. Thus, they should have been paired before, which leads to a contradiction.

arriving after it at any point in this ball. Finally, the third one is just the radius ρ_x as it is the lower bound for the connection cost outside the ball. When we use the stochastic assumption to compute this minimum, we obtain the following.

Lemma 13. *For MPMD in the Poisson arrival model, the expected cost of the optimal offline solution, over all random sequences consisting of m requests, satisfies*

$$\mathbb{E}^m_\sigma[\mathrm{OPT}(\sigma)] \geq m \cdot \frac{1-e^{-2}}{4} \sum_{x\in\mathcal{X}} \frac{\lambda_x}{\lambda(\mathcal{X})} \cdot \rho_x.$$

We present a detailed proof of this lemma in Section 4.

Following Lemmas 11, 13 and 12, we immediately conclude Theorems 1 and 2 stated in the introduction.

4 LOWER BOUNDING OPT

We prove Lemma 13 in this section. The main idea of the proof goes as follows. To obtain a lower bound on the expected cost of the optimal matching over a request sequence σ, we analyze each element of σ separately. First, we observe that for each request, the sum of its connection and delay cost is at least equal to the cost of connecting it to its cheapest neighbor in σ (in terms of connection + delay cost). Then, the core of the proof (Claim 17) consists of showing that, in expectation, this cost is at least a constant times the radius of the corresponding point.

Given any input sequence σ and any request $r \in \sigma$, we define the *minimum total cost of r in σ* as

$$c(\sigma, r) := \min_{r'\in\sigma, r'\neq r} \left\{ d(\ell(r), \ell(r')) + |t(r) - t(r')| \right\}$$

Claim 14. *For any input σ it holds that* $\mathrm{OPT}(\sigma) \geq \frac{1}{2}\sum_{r\in\sigma} c(\sigma, r)$.

Before formally stating Claim 17, we need the following two results.

Proposition 15. *Let $\sigma = (r_1, r_2, \ldots)$ be an infinite sequence of requests generated by the centralized Poisson process and ordered by their arrival times. Then, for any point $x \in \mathcal{X}$ and any index $i \geq 1$, the distribution of the waiting time for the next request to arrive after r_i at some point in a set $S \subseteq \mathcal{X}$, $x \in S$, follows an exponential distribution with parameter $\lambda(S)$.*

Claim 16. *Given any $a > 0$ and an exponential variable $Y \sim \mathrm{Exp}(\mu)$, $\mathbb{E}[\min\{Y, a\}] = \frac{1-e^{-\mu a}}{\mu a} \cdot a$.*

Now we present the core component to prove Lemma 13.

Claim 17. *Given a sequence σ, we order the requests in $\sigma = (r_1, \ldots, r_m)$ according to their arrival times. Then, for any point $x \in \mathcal{X}$ and any index i, $1 \leq i \leq m$, the expected minimum cost of the i-th request r_i in a random sequence σ, assuming that r_i is located on x, is*

$$\mathbb{E}^m_\sigma[c(\sigma, r_i) \mid \ell(r_i) = x] \geq \frac{1-e^{-2}}{2} \cdot \rho_x.$$

We provide a proof sketch here due to the space limitation. First, given any random request sequence $\sigma = (r_1, \ldots, r_m)$, we extend it by some dummy random requests r_j for $j \leq 0$ and $j \geq m$ to get an *extended* random sequence

$$\overline{\sigma} = (\ldots, r_{-2}, r_{-1}, r_0, r_1, \ldots, r_{m-1}, r_m, r_{m+1}, \ldots).$$

To generate requests before r_1 and after r_m we use the centralized Poisson arrival model (i.e., for every integer j, $(t(r_{j+1}) - t(r_j)) \sim \mathrm{Exp}(\lambda(\mathcal{X}))$ and $\mathbb{P}(\ell(r_j) = y) = \lambda_y/\lambda(\mathcal{X})$ for all $y \in \mathcal{X}$). Note that given an extended random sequence $\overline{\sigma}$, define its truncation $\overline{\sigma}_m := (r_1, \ldots, r_m)$. We have $c(\overline{\sigma}, r_j) \leq c(\sigma, r_j)$ (where $\sigma = \overline{\sigma}_m$) and

$$\mathbb{E}^m_\sigma[c(\sigma, r_i) \mid \ell(r_i) = x] \geq \mathbb{E}_{\overline{\sigma}}[c(\overline{\sigma}, r_i) \mid \ell(r_i) = x].$$

Notice that for any $j, j' \in \{0, 1, \ldots, m\}$,

$$\mathbb{E}_{\overline{\sigma}}[c(\overline{\sigma}, r_j) \mid \ell(r_j) = x] = \mathbb{E}_{\overline{\sigma}}[c(\overline{\sigma}, r_{j'}) \mid \ell(r_{j'}) = x].$$

We thus only need to show that

$$\mathbb{E}_{\overline{\sigma}}[c(\overline{\sigma}, r_0) \mid \ell(r_0) = x] \geq \frac{1-e^{-2}}{2} \cdot \rho_x.$$

To prove this bound, consider an extended sequence $\overline{\sigma}$ with $\ell(r_0) = x$. W.l.o.g., we also assume that $t(r_0) = 0$ as it can be achieved by shifting all arrival times by the same constant. Define W^- (resp. W^+) as the (random) time duration between the arrival of the last request before r_0 (resp. first request after r_0) arriving at any point $y \in B^\circ(x, \rho_x)$ and the arrival of r_0. By definition of $c(\overline{\sigma}, r_0)$,

$$c(\overline{\sigma}, r_0) \geq \min\left\{ \min\{W^-, W^+\}, \rho_x \right\}.$$

In fact, W^- and W^+ are mutually independent and follow the same exponential distribution $\mathrm{Exp}(\lambda(B^\circ(x, \rho_x)))$. Thanks to Proposition 7, we immediately have

$$\min\{W^-, W^+\} \sim \mathrm{Exp}(2\lambda(B^\circ(x, \rho_x))).$$

By Claim 16 (with $a = \rho_x$, $\mu = 2\lambda(B^\circ(x, \rho_x))$), it follows that

$$\begin{aligned}\mathbb{E}_{\overline{\sigma}}[c(\overline{\sigma}, r_0) \mid \ell(r_0) = x] \quad &\geq \quad \mathbb{E}_{\overline{\sigma}}\Big[\min\left\{\min\{W^-, W^+\}, \rho_x\right\}\Big] \\ &= \quad \frac{1-e^{-2\lambda(B^\circ(x,\rho_x))\cdot\rho_x}}{2\lambda(B^\circ(x,\rho_x))\cdot\rho_x} \cdot \rho_x.\end{aligned}$$

Since the function $t \mapsto \frac{1-e^{-t}}{t}$ is strictly decreasing, together with Observation 10, we have $\lambda(B^\circ(x, \rho_x)) \cdot \rho_x \leq 1$ and

$$\frac{1-e^{-2\lambda(B^\circ(x,\rho_x))\cdot\rho_x}}{2\lambda(B^\circ(x,\rho_x))\cdot\rho_x} \geq \frac{1-e^{-2}}{2},$$

$$\mathbb{E}_{\overline{\sigma}}[c(\overline{\sigma}, r_0) \mid \ell(r_0) = x] \geq \frac{1-e^{-2}}{2} \cdot \rho_x,$$

which concludes the proof of Claim 17.

Proof of Lemma 13. Let $\sigma = (r_1, \ldots, r_m)$ be a sequence of requests sorted by increasing order arrival times. We have

$$\begin{aligned}&\mathbb{E}^m_\sigma[\mathrm{OPT}(\sigma)] \\ &\geq \mathbb{E}^m_\sigma\left[\frac{1}{2}\sum_{i=1}^m c(\sigma, r_i)\right] && \text{(Claim 14)} \\ &= \frac{1}{2}\sum_{i=1}^m \sum_{x\in\mathcal{X}} \mathbb{P}_\sigma(\ell(r_i) = x) \cdot \mathbb{E}^m_\sigma[c(\sigma, r_i) \mid \ell(r_i) = x] \\ &\geq \frac{1}{2}\sum_{i=1}^m \sum_{x\in\mathcal{X}} \frac{\lambda_x}{\lambda(\mathcal{X})} \cdot \frac{1-e^{-2}}{2} \cdot \rho_x && \text{(Claim 17)} \\ &= m \cdot \frac{1-e^{-2}}{4} \sum_{x\in\mathcal{X}} \frac{\lambda_x}{\lambda(\mathcal{X})} \cdot \rho_x.\end{aligned}$$

This concludes the proof. □

5 UPPER BOUNDING THE GREEDY SOLUTION

In this section, we prove Lemma 11 that establishes an upper bound on the expected cost of the Greedy algorithm. We first observe that the total connection cost of the Greedy solution is at most equal to its total delay cost, and then we bound the latter.

Given any input sequence σ, let (M, s) denote the solution output by the Greedy algorithm, where M is the set of matched pairs of requests, and s is the service times of the requests. The waiting time of a request $r \in \sigma$ is denoted by $w(r) := s(r) - t(r)$. Greedy matches two requests r and r' when the sum of their delay cost $w(r) + w(r')$ is at least equal to their distance $d(\ell(r), \ell(r'))$. In particular, when summing over all requests we obtain:

Claim 18. *For any input sequence σ, the cost of the solution returned by the Greedy algorithm is at most twice its total delay cost, i.e.,* $\text{Greedy}(\sigma) \leq 2\sum_{r\in\sigma} w(r)$.

To upper bound the waiting time of each request, we distinguish two types of requests. For each request $r \in \sigma$, define $t'(r) := t(r) + \rho_{\ell(r)}$. We say that r is a *late* request if (1) r is still pending at time $t'(r)$ and (2) there is no request r' arriving within the closed ball of r's location (i.e., $d(\ell(r), \ell(r')) \leq \rho_{\ell(r)}$) after time $t'(r)$. Otherwise, we say that r is a *nice* request, and we define $Y_r^{\text{nice}} := 0$ if r is matched at time $t'(r)$; otherwise we define Y_r^{nice} as

$$\min_{r'\in\sigma} \left\{ t(r') - t'(r) \mid t(r') > t'(r) \text{ and } d(\ell(r'), \ell(r)) \leq \rho_{\ell(r)} \right\}.$$

We bound the waiting time of nice requests as follows:

Claim 19. *For each nice request $r \in \sigma$, we have $w(r) \leq \rho_{\ell(r)} + Y_r^{nice}$.*

We now bound the total delay time induced by late requests. Unfortunately, the waiting time of a late request can possibly be as large as the diameter $d_{\max} = \max_{x,y\in\mathcal{X}} d(x, y)$ of the metric space. However, we show that there are only constantly many such requests. Let $t(r_m)$ denote the arrival time of the last request in σ. For any late request r, we define Y_r^{late} as follows:

- if $t(r) + d_{\max} \geq t(r_m)$, then $Y_r^{\text{late}} := 0$,
- otherwise, $t(r) + d_{\max} < t(r_m)$, then

$$Y_r^{\text{late}} := \min_{r'\in\sigma} \{ t(r') - (t(r) + d_{\max}) \mid t(r') > t(r) + d_{\max} \}.$$

Claim 20. *For any point $x \in \mathcal{X}$, there is at most one late request located on x. In particular, there are at most $|\mathcal{X}|$ late requests. Moreover, for each late request r, we have $w(r) \leq d_{\max} + Y_r^{late}$.*

Thanks to Claims 19 and 20, we can show that

- for each late request, the expected cost is $\leq d_{\max} + \frac{1}{\lambda(\mathcal{X})}$;
- for each nice request, the expected cost is $\leq \sum_{x\in\mathcal{X}} \frac{\lambda_x}{\lambda(\mathcal{X})} \cdot 2\rho_x$,

which concludes Lemma 11 (see the full proof in [45]).

6 UPPER BOUNDING THE RADIUS SOLUTION

In this section, we prove Lemma 12 that establishes an upper bound on the expected cost of the Radius algorithm. To bound the total cost of the solution produced by the Radius, we separately analyze the delay cost and the connection cost.

To bound the connection cost we differentiate two types of edges (pairs matched by the algorithm). Let M denote the matching produced by the Radius algorithm on the input sequence σ. Let us call $e \in M$ a *nice* edge[3] if the corresponding matched pair was created during the main loop of the algorithm, i.e., before the arrival time of the last request in σ. Otherwise, we call this edge *late*. Similarly, a request is called *nice* if it is an endpoint of a nice edge, and *late* otherwise. Intuitively, since the late requests are matched arbitrarily by the Radius algorithm, the connection cost induced by each of these late edges can possibly be as large as the diameter of the metric space. Fortunately, there are only a constant number of them (i.e., independent from m).

Claim 21. *For any point $x \in \mathcal{X}$, there is at most one late request located on x. In particular, there are at most $|\mathcal{X}|/2$ late edges.*

We now bound the connection cost of the solution induced by nice edges. Two nice requests are matched together by the Radius algorithm if and only if their distance is at most the sum of their radii. By summing over all the nice edges, we obtain:

Claim 22. *For any input sequence σ, the connection cost induced by all nice edges is at most $\sum_{r\in\sigma} \rho_{\ell(r)}$.*

We now upper bound the total delay cost. Let $t(r_m)$ denote the arrival time of the last request of σ, which correspond to the time at which all remaining pending (late) requests are matched together by the Radius algorithm. Let r be any request in σ. We define Y_r as the duration between the arrivals of r and the first request r' that appears on a point of $\overline{B}(\ell(r), \rho_{\ell(r)})$ after r. If there is no such request $r' \in \sigma$, then we set $Y_r := t(r_m) - t(r)$.

Claim 23. *Each request r in σ is delayed by the Radius algorithm for a time at most equal to Y_r.*

Let σ be a sequence of m requests, and let M denote the perfect matching output by the Radius algorithm. We split it into two sets M_{nice} and M_{late} of nice and late edges, respectively. The total cost $\text{Radius}(\sigma)$ of the solution is equal to $\text{CC}(M_{\text{nice}}) + \text{CC}(M_{\text{late}}) + \text{DC}$, the sum of the connection cost $\text{CC}(M_{\text{nice}})$ induced by the nice edges, the connection cost $\text{CC}(M_{\text{late}})$ induced by the late edges and the total delay cost DC. We show that

- $\text{CC}(M_{\text{late}}) \leq \frac{1}{2} \cdot |\mathcal{X}| \cdot d_{\max}$;
- $\mathbb{E}_\sigma^m [\text{CC}(M_{\text{nice}})] \leq m \sum_{x\in\mathcal{X}} \frac{\lambda_x}{\lambda(\mathcal{X})} \cdot \rho_x$;
- $\mathbb{E}_\sigma^m [\text{DC}] \leq m \sum_{x\in\mathcal{X}} \frac{\lambda_x}{\lambda(\mathcal{X})} \cdot \rho_x$,

which concludes Lemma 12 (see the full proof in [45]).

7 EXTENSION TO GENERAL DELAY COSTS

In this section, we study a generalization of the MPMD problem where the delay cost function is not required to be linear. In this version of the problem, referred to as f-MPMD, the decision of matching a request can be postponed for time t at a delay cost of $f(t)$, where f is the *delay cost function.* We require this function to be *positive* (otherwise some solutions may have negative value), and *non-decreasing*. In the full version of this paper [45], we show that w.l.o.g., we can assume $f(0) = 0$, i.e., if a request is directly matched at its arrival time, no delay cost is incurred.

This more general version of MPMD has been investigated for the classic online adversarial model among others by Azar et al. [8] and Liu et al. [43]. Their works suggest that in general, the f-MPDM

[3] Notice that the current definition differs from the previous section.

problem is even more challenging than the original MPMD problem. For instance, in [8], Azar et al. considered a special type of concave delay cost function and showed that even for the single-point metric, obtaining a constant competitive algorithm is a non-trivial task (whereas for the linear case, the optimal algorithm simply matches two consecutive requests). Liu et al. [43] showed that under some natural requirements for function f, any deterministic online algorithm for f-MPMD on a k-points metric must have a competitive ratio $\Omega(k)$.

In the online stochastic Poisson arrival model, we show in Theorem 25 that our Greedy and Radius algorithms for MPMD can be adapted to f-MPMD, and that their corresponding ratios-of-expectations remain a constant, which depends on f.

The adaptation of the Greedy algorithm for f-MPMD is quite straightforward: when the sum of the delay cost of two pending requests exceeds their distance, match them together, i.e., match pending requests r and r' at time t whenever $f(t - t(r)) + f(t - t(r')) \geq d(\ell(r), \ell(r'))$. The Radius algorithm works in the general case exactly as in the linear case but using the following generalized definition of radius.

Definition 24. *Given the positive and non-decreasing delay cost function f, for any point $x \in \mathcal{X}$, define its radius ρ_x as the smallest value $u \in \mathbb{R}^{+} \cup \{\infty\}$ such that $u \geq \mathbb{E}[f(\overline{W}(x,u))]$. Here $\overline{W}(x,u)$ is an exponential variable of parameter $\lambda(\overline{B}(x,u)) := \sum_{y \in \mathcal{X}: d(x,y) \leq u} \lambda_y$.*

Since functions $u \mapsto \lambda(\overline{B}(x,u))$ and f are both non-decreasing, the function $u \mapsto \mathbb{E}[f(\overline{W}(x,u))]$ is non-increasing. This implies that the radius of each point is well-defined and unique. Moreover, since $\mathbb{E}[\overline{W}(x,u))] = 1/\lambda(\overline{B}(x,u))$, in the case when $f(t) = t$, this definition coincides with our initial Definition 9. Similarly as presented in Observation 10, it is easy to see that $\mathbb{E}[f(W^{\circ}(x,\rho_x))] \geq \rho_x$, where $W^{\circ}(x,\rho_x)$ is a random variable of parameter $\lambda(B^{\circ}(x,\rho_x)) := \sum_{y \in \mathcal{X}: d(x,y) < \rho_x} \lambda_y$.

Theorem 25. *Consider an instance of the f-MPMD problem such that $\mathbb{E}[f(X)] < \infty$, where $X \sim \mathrm{Exp}(\lambda(\mathcal{X}))$ is an exponential variable of parameter $\lambda(\mathcal{X}) := \sum_{x \in \mathcal{X}} \lambda_x$. Then, both the Greedy and Radius algorithms achieve ratio-of-expectations of $O(K_f)$, where k_f denotes*

$$\max_{\mu > 0} \left\{ \frac{\mathbb{E}[f(X)]}{\mathbb{E}[\min\{f(X'), \mathbb{E}[f(X)]\}]} \mid X \sim \mathrm{Exp}(\mu), X' \sim \mathrm{Exp}(2\mu) \right\}.$$

8 PAYING PENALTIES TO CLEAR REQUESTS

Let us now consider a variant of MPMD called MPMDfp [30], where it is allowed to clear any request by paying a fixed penalty $p > 0$. For this problem, we propose the following algorithm ALG, which works similarly to Radius, obtaining a constant ratio-of-expectations.

Given the metric space $(\mathcal{X}, d)$, define $\mathcal{X}^{(1)} = \{x \in \mathcal{X} : \rho_x < p\}$ and $\mathcal{X}^{(2)} = \{x \in \mathcal{X} : \rho_x \geq p\}$ (where ρ_x is the radius of point $x \in \mathcal{X}$ as defined in Definition 9). Suppose that at time t, a new request r arrives. Then, our algorithm performs the following actions depending on whether $\ell(r) \in \mathcal{X}^{(2)}$ or $\ell(r) \in \mathcal{X}^{(1)}$:

- Suppose $\ell(r) \in \mathcal{X}^{(2)}$. If there exists a pending request r' located at point $y \in \mathcal{X}^{(1)}$ and $x \in \overline{B}(y, \rho_y)$, then match r with r'. Otherwise, clear r.
- Suppose $\ell(r) \in \mathcal{X}^{(1)}$. Apply the Radius algorithm to match this request.

There possibly exist an odd number of late requests (due to clearing an odd number of requests arriving at points $\mathcal{X}^{(2)}$). In that case, ALG has to clear the last request r_m even when $\ell(r_m) \in \mathcal{X}^{(1)}$.

Theorem 26. *For MPMDfp in the Poisson arrival model,* ALG *achieves a ratio-of-expectations of* $8/(1 - e^{-2})$.

9 CONCLUSION

In this paper, we studied MPMD with additional stochastic assumptions on the sequence of the input requests. In the case where the requests follow a Poisson arrival process, we presented two simple deterministic online algorithms with constant ratios-of-expectations. In particular, we observed that the cost of the optimal offline solution is proportional to the number of requests in the sequence, and gave a tight (up to a constant factor independent from the instance) estimation of the constant of proportionality. In the following text, we briefly discuss some potential future directions.

Bipartite Case in the Poisson Arrival Model. Previously, the bipartite version of MPMD (i.e., MBPMD) has been considered in the adversarial model [2] where each request has a color, either red or blue, and only requests of different colors can be matched into a pair[4]. In an equivalent definition, given the metric space $\mathcal{M} = (\mathcal{X}, d)$, the points of $\mathcal{X}$ are partitioned into two subsets A and B, such that the requests arriving at points A can only be matched with requests from points B. Ashlagi et al. [2] proposed two O(log n)-competitive randomized online algorithms for this problem. Besides, they established a lower bound of $\Omega(\sqrt{\log n / \log\log n})$ on the competitive ratio of any online algorithm. Note that the MBPMD problem can be seen as a special case of the *non-metric* perfect matching problem with delays, where the connection cost function $d : \mathcal{X} \times \mathcal{X} \to \mathcal{R}_{+} \cup \{\infty\}$ can have infinite values and is no longer assumed to satisfy the triangle inequality.

A natural direction would be to explore MBPMD in the Poisson arrival model. Unfortunately, an initial difficulty is established: the expected cost of the offline optimal algorithm on a random sequence of length m cannot be upper bounded by $O(m)$ (see the detail in the full version of this paper [45]).

k-way Min-cost Perfect Matching with Delays. Another direction, that was introduced by [46] for the online adversarial model, would be to consider a generalized k-way min-cost perfect matching with delays (k-MPMD) in the stochastic input model, where each pair (a.k.a., k-tuple) consists of k different requests ($k \geq 2$ is an arbitrary integer). Note that such k-MPMD problem indeed has real applications from ride-sharing taxi platforms (when a taxi picks up k passengers from different locations for one ride) and online gaming platforms (when a gaming session consists of k different players). To attack this version of the MPMD problem, one should first come out with a suitable notion of "connection cost" of a k-set. This might be for instance measured by the maximum distance between any two requests of that set, the average distance, the weight of a minimum spanning tree, etc.

[4] For an application, imagine that the red requests come from customers and the blue ones represent the suppliers.

ACKNOWLEDGMENTS

This work was partially supported by the ERC CoG grant TUgbOAT no 772346 and NCN grant no 2020/37/B/ST6/04179.

REFERENCES

[1] Ali Aouad and Ömer Saritaç. 2020. Dynamic stochastic matching under limited time. In *Proc. EC.* 789–790.

[2] Itai Ashlagi, Yossi Azar, Moses Charikar, Ashish Chiplunkar, Ofir Geri, Haim Kaplan, Rahul Makhijani, Yuyi Wang, and Roger Wattenhofer. 2017. Min-cost bipartite perfect matching with delays. In *Proc. APPROX / RANDOM.* 1:1–1:20.

[3] Yossi Azar, Ashish Chiplunkar, and Haim Kaplan. 2017. Polylogarithmic bounds on the competitiveness of min-cost perfect matching with delays. In *Proc. SODA.* 1051–1061.

[4] Yossi Azar, Ashish Chiplunkar, Shay Kutten, and Noam Touitou. 2020. Set Cover with Delay–Clairvoyance Is Not Required. In *Proc. ESA.*

[5] Yossi Azar, Yuval Emek, Rob van Stee, and Danny Vainstein. 2019. The price of clustering in bin-packing with applications to bin-packing with delays. In *Proc. SPAA.* 1–10.

[6] Yossi Azar, Arun Ganesh, Rong Ge, and Debmalya Panigrahi. 2017. Online service with delay. In *Proc. STOC.* 551–563.

[7] Yossi Azar and Amit Jacob-Fanani. 2020. Deterministic min-cost matching with delays. *Theory of Computing Systems* 64, 4 (2020), 572–592.

[8] Yossi Azar, Runtian Ren, and Danny Vainstein. 2021. The min-cost matching with concave delays problem. In *Proc. SODA.* 301–320.

[9] Yossi Azar and Noam Touitou. 2019. General framework for metric optimization problems with delay or with deadlines. In *Proc. FOCS.* 60–71.

[10] Yossi Azar and Noam Touitou. 2020. Beyond tree embeddings–a deterministic framework for network design with deadlines or delay. In *Proc. FOCS.* 1368–1379.

[11] Haris Aziz, Péter Biró, Tamás Fleiner, Serge Gaspers, Ronald de Haan, Nicholas Mattei, and Baharak Rastegari. 2017. Stable Matching with Uncertain Pairwise Preferences. In *Proc. AAMAS.* 344–352.

[12] Johannes Bäumler, Martin Bullinger, Stefan Kober, and Donghao Zhu. 2022. High Satisfaction in Thin Dynamic Matching Markets. *arXiv preprint arXiv:2206.10287* (2022).

[13] Marcin Bienkowski, Martin Böhm, Jaroslaw Byrka, Marek Chrobak, Christoph Dürr, Lukáš Folwarczný, Łukasz Jeż, Jiri Sgall, Kim Thang Nguyen, and Pavel Veselỳ. 2016. Online Algorithms for Multi-Level Aggregation. In *Proc. ESA.*

[14] Marcin Bienkowski, Martin Böhm, Jaroslaw Byrka, Marek Chrobak, Christoph Dürr, Lukáš Folwarczný, Łukasz Jeż, Jiří Sgall, Nguyen Kim Thang, and Pavel Veselỳ. 2021. New results on multi-level aggregation. *Theoretical Computer Science* 861 (2021), 133–143.

[15] Marcin Bienkowski, Martin Böhm, Jarosław Byrka, and Jan Marcinkowski. 2022. Online Facility Location with Linear Delay. In *Proc. APPROX/RANDOM.* 45:1–45:17.

[16] Marcin Bienkowski, Artur Kraska, Hsiang-Hsuan Liu, and Paweł Schmidt. 2018. A primal-dual online deterministic algorithm for matching with delays. In *Proc. WAOA.* 51–68.

[17] Marcin Bienkowski, Artur Kraska, and Paweł Schmidt. 2017. A match in time saves nine: Deterministic online matching with delays. In *Proc. WAOA.* 132–146.

[18] Marcin Bienkowski, Artur Kraska, and Paweł Schmidt. 2018. Online service with delay on a line. In *Proc. SIROCCO.* 237–248.

[19] Niclas Boehmer, Markus Brill, and Ulrike Schmidt-Kraepelin. 2022. Proportional Representation in Matching Markets: Selecting Multiple Matchings under Dichotomous Preferences. In *Proc. AAMAS.* 136–144.

[20] Angelina Brilliantova and Hadi Hosseini. 2022. Fair Stable Matching Meets Correlated Preferences. In *Proc. AAMAS.* 190–198.

[21] B. Brubach, KA Sankararaman, A. Srinivasan, and Pan Xu. 2017. Attenuate Locally, Win Globally: An Attenuation-based Framework for Online Stochastic Matching with Timeouts. In *Proc. AAMAS.* 1223–1231.

[22] Niv Buchbinder, Moran Feldman, Joseph Naor, and Ohad Talmon. 2017. O(depth)-competitive algorithm for online multi-level aggregation. In *Proc. SODA.* 1235–1244.

[23] Rodrigo A. Carrasco, Kirk Pruhs, Cliff Stein, and José Verschae. 2018. The online set aggregation problem. In *Proc. LATIN.* 245–259.

[24] Ryder Chen, Jahanvi Khatkar, and Seeun William Umboh. 2022. Online Weighted Cardinality Joint Replenishment Problem with Delay. In *Proc. ICALP.* 40:1–40:18.

[25] Sung-Ho Cho, Taiki Todo, and Makoto Yokoo. 2022. Two-Sided Matching over Social Networks. In *Proc. IJCAI.* 186–193.

[26] Natalie Collina, Nicole Immorlica, Kevin Leyton-Brown, Brendan Lucier, and Neil Newman. 2020. Dynamic weighted matching with heterogeneous arrival and departure rates. In *Proc. WINE.* 17–30.

[27] Lindsey Deryckere and Seeun William Umboh. 2022. Online Matching with Set Delay. *arXiv preprint arXiv:2211.02394* (2022).

[28] Jack Edmonds. 1965. Maximum matching and a polyhedron with 0, 1-vertices. *Journal of research of the National Bureau of Standards B* 69, 125-130 (1965), 55–56.

[29] Jack Edmonds. 1965. Paths, trees, and flowers. *Canadian Journal of mathematics* 17 (1965), 449–467.

[30] Yuval Emek, Shay Kutten, and Roger Wattenhofer. 2016. Online matching: haste makes waste!. In *Proc. STOC.* 333–344.

[31] Yuval Emek, Yaacov Shapiro, and Yuyi Wang. 2019. Minimum cost perfect matching with delays for two sources. *Theoretical Computer Science* 754 (2019), 122–129.

[32] Leah Epstein. 2021. On bin packing with clustering and bin packing with delays. *Discrete Optimization* 41 (2021), 100647.

[33] Alireza Farhadi, Jacob Gilbert, and MohammadTaghi Hajiaghayi. 2022. Generalized Stochastic Matching. In *Proc. AAAI.* 10008–10015.

[34] Mohak Goyal. 2022. Secretary Matching With Vertex Arrivals and No Rejections. In *Proc. AAAI*, Vol. 36. 5051–5058.

[35] Anupam Gupta, Amit Kumar, and Debmalya Panigrahi. 2020. Caching with time windows. In *Proc. STOC.* 1125–1138.

[36] Naonori Kakimura and Donghao Zhu. 2021. Dynamic Bipartite Matching Market with Arrivals and Departures. In *Proc. WINE.* 544.

[37] Naoyuki Kamiyama. 2020. On stable matchings with pairwise preferences and matroid constraints. In *Proc. AAMAS.* 584–592.

[38] Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. 1990. An optimal algorithm for on-line bipartite matching. In *Proc. STOC.* 352–358.

[39] Yasushi Kawase. 2020. Approximately Stable Matchings with General Constraints. In *Proc. AAMAS.* 602–610.

[40] Walter Kern, Péter Biró, Dömötör Pálvölgyi, and Daniël Paulusma. 2019. Generalized matching games for international kidney exchange. In *Proc. AAMAS.* 413–421.

[41] Kristen Kessel, Ali Shameli, Amin Saberi, and David Wajc. 2022. The stationary prophet inequality problem. In *Proc. EC.* 243–244.

[42] Ngoc Mai Le, Seeun William Umboh, and Ningyuan Xie. 2023. The Power of Clairvoyance for Multi-Level Aggregation and Set Cover with Delay. In *Proc. SODA.* 1594–1610.

[43] Xingwu Liu, Zhida Pan, Yuyi Wang, and Roger Wattenhofer. 2018. Impatient online matching. In *Proc. ISAAC*, Vol. 123. 62:1–62:12.

[44] Will Ma, Pan Xu, and Yifan Xu. 2022. Group-level Fairness Maximization in Online Bipartite Matching. In *Proc. AAMAS.* 1687–1689.

[45] Mathieu Mari, Michał Pawłowski, Runtian Ren, and Piotr Sankowski. 2022. Online matching with delays and stochastic arrival times. *arXiv preprint arXiv:2210.07018* (2022).

[46] Darya Melnyk, Yuyi Wang, and Roger Wattenhofer. 2021. Online k-Way Matching with Delays and the H-Metric. *arXiv preprint arXiv:2109.06640* (2021).

[47] Maria Silvia Pini, Francesca Rossi, and Kristen Brent Venable. 2014. Stable matching problems with soft constraints. In *Proc. AAMAS.* 1511–1512.

[48] Noam Touitou. 2021. Nearly-Tight Lower Bounds for Set Cover and Network Design with Deadlines/Delay. In *Proc. ISAAC.* 53:1–53:16.

[49] Yu-Hang Zhou, Chen Liang, Nan Li, Cheng Yang, Shenghuo Zhu, and Rong Jin. 2019. Robust online matching with user arrival distribution drift. In *Proc. AAAI*, Vol. 33. 459–466.