Stochastic Online Metric Matching

Anupam Gupta[*†1], Guru Guruganesh[‡§2], Binghui Peng[¶‖3], and David Wajc[**††1]

[1]Carnegie Mellon University and University of Pittsburgh
[2]Google Research
[3]Tsinghua University

Abstract

We study the minimum-cost metric perfect matching problem under online i.i.d arrivals. We are given a fixed metric with a server at each of the points, and then requests arrive online, each drawn independently from a known probability distribution over the points. Each request has to be matched to a free server, with cost equal to the distance. The goal is to minimize the expected total cost of the matching.

Such stochastic arrival models have been widely studied for the *maximization* variants of the online matching problem; however, the only known result for the *minimization* problem is a tight $O(\log n)$-competitiveness for the random-order arrival model. This is in contrast with the adversarial model, where an optimal competitive ratio of $O(\log n)$ has long been conjectured and remains a tantalizing open question.

In this paper, we show improved results in the i.i.d arrival model. We show how the i.i.d model can be used to give substantially better algorithms: our main result is an $O((\log\log\log n)^2)$-competitive algorithm in this model. Along the way we give a 9-competitive algorithm for the line and tree metrics. Both results imply a strict separation between the i.i.d model and the adversarial and random order models, both for general metrics and these much-studied metrics.

1 Introduction

We study the minimum-cost metric (perfect) matching problem under online i.i.d. arrivals. In this problem, we are given a fixed metric (S, d) with a server at each of the $n = |S|$ points. Then n requests arrive online, where each request is at a location that is drawn independently from a known probability distribution $\mathcal{D}$ over the points. Each such arriving request has to be matched immediately and irrevocably to a free server, whereupon it incurs a cost equal to distance of its location to this server. The goal is to minimize the total expected cost.

[*] Email address: anupamg@cs.cmu.edu.

[†] Supported in part by NSF awards CCF-1536002, CCF-1540541, and CCF-1617790, and the Indo-US Joint Center for Algorithms Under Uncertainty.

[‡] Email address: gurug@google.com

[§] Work done in part while the author was at Carnegie Mellon University.

[¶] Email address: pbh15@mails.tsinghua.edu.cn.

[‖] Work done in part while the author was visiting Carnegie Mellon University.

[**] Email address: dwajc@cs.cmu.edu.

[††] Supported in part by NSF grants CCF-1618280, CCF-1814603, CCF-1527110, NSF CAREER award CCF-1750808 and a Sloan Research Fellowship.

The minimization version of online matching was first considered in the standard adversarial setting by Khuller et al. [27] and Kalyanasundaram and Pruhs [24]; both papers showed $(2n-1)$-competitive deterministic algorithms, and proved that this was tight for, say, the star metric. After about a decade, a randomized algorithm with an $O(\log^3 n)$-competitiveness was given by Meyerson et al. [32]; this was improved to $O(\log^2 n)$ by Bansal et al. [4], which remains the best result known. (Recall that the maximization version of matching problems have been very widely studied, but they use mostly unrelated techniques.)

The competitive ratio model with adversarial online arrivals is often considered too pessimistic, since it assumes an all-powerful adversary. One model to level the playing field, and to make the model perhaps closer to practice, is to restrict the adversary's power. Two models have been popular here: the *random-order arrivals* (or *secretary*) model, and the *i.i.d.* model defined above. The random-order model is a *semi-random* model, in which the worst-case input is subjected to random perturbations. Specifically, the adversary chooses a *set* of requests, which are then presented to the algorithm in a uniformly random order. The min-cost online matching problem in this random-order model was studied by Raghvendra, who gave a tight $O(\log n)$-competitive algorithm [37]. The random-order model also captures the i.i.d. setting, so the natural goal is to get a better algorithm for the i.i.d. model. Indeed, our main result for the i.i.d. model gives exactly such a result:

Theorem 1.1 (Main Theorem)**.** *There is an $O((\log\log\log n)^2)$-competitive algorithm for online minimum-cost metric perfect matching in the i.i.d. setting.*

Observe that the competitiveness here is better than the lower bounds of $\Omega(\log n)$ known for the worst-case and random-order models.

Matching on the Line and Trees. There has also been much interest in solving the problem for the line metric: a deterministic lower bound of $(9+\varepsilon)$ for some $\varepsilon > 0$ is known, showing it is strictly harder than the optimal search (or "cow-path") problem, which it generalizes [14]. However, getting better results for the line than for general metrics has been elusive: an $O(\log n)$-competitive *randomized* algorithm for line metrics (and for doubling metrics) was given by [19]. In the *deterministic* setting, recently Nayyar and Raghvendra [36] gave an $O(\log^2 n)$-competitive algorithm, whose competitive ratio was subsequently proven to be $O(\log n)$ by Raghvendra [38], improving on the $o(n)$-competitive algorithm of Antoniadis et al. [2]. To the best of our knowledge, nothing better is known for tree metrics than for general metrics in both the adversarial and the random-order models. Our second result for the i.i.d. model is a constant-competitive algorithm for tree metrics.

Theorem 1.2 (Algorithm for Trees)**.** *There is a 9-competitive algorithm for online minimum-cost metric perfect matching on tree metrics in the i.i.d. setting.*

Observe that the competitiveness here is better than the lower bound of $9+\epsilon$ for line metrics in the worst-case model.

Max-Weight Perfect Matching. Recently, Chang et al. [7] presented a $1/2$-competitive algorithm for the *maximum*-weight perfect matching problem in the i.i.d. setting. We show that our algorithm is versatile, and that a small change to our algorithm gives us a maximization variant matching this factor of $1/2$. Our approach differs from that of [7], in that we match an arriving request based on the realization of free servers, while they do so based on the "expected realization". See Appendix D for details.

1.1 Our Techniques

Both Theorems 1.1 and 1.2 are achieved by the same algorithm. The first observation guiding this algorithm is that we may assume that the distribution $\mathcal{D}$ of request locations is just the uniform distribution on the server locations. (In Appendix A we show how this assumption can be removed with a constant factor loss in the competitiveness.) Our algorithm is inspired by the following two complementary consequences of the uniformity of $\mathcal{D}$.

- Firstly, each of the $n-t+1$ free servers' locations at time t are equally likely to get a request in the future, and as such they should be left unmatched with equal probability. Put otherwise, we should match to them with equal probability of $1/(n-t+1)$. However, matching *any* arriving request to any free server with probability $1/(n-t+1)$ is easily shown to be a bad choice.

- So instead, we rely on the second observation: the t^{th} request is equally likely to arrive at each of the n server locations. This means we can couple the matching of free server locations with the location of the next request, to guarantee a marginal probability of $1/(n-t+1)$ for each free server to be matched at time t.

Indeed, the constraints that each location is matched at time t with probability $1/n$ (i.e., if it arrives) and each of the free servers are matched with marginal probability $1/(n-t+1)$ can be expressed as a *bipartite flow* instance, which guides the coupling used by the algorithm. Loosely speaking, our algorithm is fairly intuitive. It finds a min-cost fractional matching between the current open server locations and the expected arrivals, and uses that to match new requests. The challenge is to bound the competitive ratio—in contrast to previously used approaches (for the maximization version of the problem) it does not just try to match vertices using a fixed template of choices, but rather dynamically recomputes a template after each arrival.

A major advantage of this approach is that we understand the distribution of the open servers. We maintain the invariant that after t steps, the set of free servers form a uniform random $(n-t)$-subset of $[n]$—the randomness being over our choices, and over the randomness of the input. This allows us to relate the cost of the algorithm in the t^{th} step to the expected cost of this optimal flow between the original n points and a uniformly random subset of $(n-t)$ of these points. The latter expected cost is just a statistic based on the metric, and does not depend on our algorithm's past choices. For paths and trees, we bound this quantity explicitly by considering the variance across edge-cuts in the tree—this gives us the proof of Theorem 1.2.

Since general metrics do not have any usable cut structure, we need a different idea for Theorem 1.1. We show that tree-embedding results can be used either explicitly in the algorithm or just implicitly in the proof, but both give an $O(\log n)$ loss. To avoid this loss, we use a different balls-and-bins argument to improve our algorithm's competitiveness to $O((\log\log n))^2)$. In particular, we provide better bounds on our algorithm's per-step cost in terms of $\mathbb{E}[OPT]$ and the expected load of the k most loaded bins in a balls and bins process, corresponding to the number of requests in the k most frequently-requested servers. Specifically, we show that $\mathbb{E}[OPT]$ is bounded in terms of the expected *imbalance* between the number of requests and servers in these top k server locations. Coupling this latter uniform k-tuple with the uniform k-tuple of free servers left by our algorithm, we obtain our improved bounds on the per-step cost of our algorithm in terms of $\mathbb{E}[OPT]$ and these bins' load, from which we obtain our improved $O((\log\log n)^2)$ competitive ratio. Interestingly, combining both balls and bins and tree embedding bounds for the per-step cost of step k (appealing to different bounds for different ranges of k) gives us a further improvement: we prove that our algorithm is $O((\log\log\log n)^2)$ competitive.

1.2 Further Related Work

I.i.d. stochastic arrivals have been studied for various online problems, e.g., for Steiner tree/forest [16], set cover [18], and k-server [9]. Closer to our work, stochastic arrivals have been widely studied in the online matching literature, though so far mostly for maximization variants. Much of this work was motivated by applications to online advertising, for which the worst-case optimal $(1 - 1/e)$-competitive ratios [1, 26, 31] seem particularly pessimistic, given the financial incentives involved and time-learned information about the distribution of requests. Consequently, many stochastic arrival models have been studied, and shown to admit better than $1 - 1/e$ competitive guarantees. The stochastic models studied for online matching and related problems, in increasing order of attainable competitive ratios, include random order (e.g., [17, 25, 29]), unknown i.i.d.—where the request distribution is unknown—(e.g., [10, 33]), and known i.i.d. (e.g., [3, 6, 13]). Additional work has focused on interpolating between adversarial and stochastic input (e.g., [11, 28]). See Mehta's survey [30] and recent work [8, 15, 21, 22, 23, 35] for more details. The long line of work on online matching, both under adversarial and stochastic arrivals, have yielded a slew of algorithmic design ideas, which unfortunately do not seem to carry over to minimization problems, nor to perfect matching problems.

As mentioned above, the only prior work for stochastic online matching with minimization objectives was the random order arrival result of Raghvendra [37]. We are hopeful that our work will spur further research in online minimum-cost perfect matching under stochastic arrivals, and close the gap between our upper bounds and the (trivial) lower bounds for the problem.

2 Our Algorithm

In this section we present our main algorithm, together with some of its basic properties. Throughout the paper we assume that the distribution over request locations is uniform over the n servers' locations. We show in Appendix A that this assumption is WLOG: it increases the competitive ratio by at most a constant. In particular, we show the following.

Lemma 2.1. *Given an α-competitive algorithm $\mathrm{ALG}_{\mathcal{U}}$ for the* uniform *distribution over server locations, $\mathcal{U}$, we can construct a $(2\alpha + 1)$-competitive algorithm $\mathrm{ALG}_{\mathcal{D}}$ for any distribution $\mathcal{D}$.*

Focusing on the uniform distribution over server locations, our algorithm is loosely the following: in each round of the algorithm, we compute an optimal fractional matching between remaining free servers and remaining requests (in expectation). Now when a new request arrives, we just match the newly-arrived request according to this matching.

2.1 Notation

Our analysis will consider k-samples from the set $S = [n]$ both with and without replacement. We will set up the following notation to distinguish them:

- Let $\mathcal{I}_k$ be the distribution over k-sub-multisets of $S = [n]$ obtained by taking k i.i.d. samples from the uniform distribution over S. (E.g., $\mathcal{I}_n$ is the request set's distribution.)
- Let $\mathcal{U}_k$ be the distribution over k-subsets of S obtained by picking a uniformly random k-subset from $\binom{S}{k}$.

In other words, $\mathcal{I}_k$ is the distribution obtained by picking k elements from S uniformly *with* replacement, whereas $\mathcal{U}_k$ is *without* replacement.

For a sub-(multi)set $T \subseteq S$ of servers, let $M(T)$ denote the optimal fractional min-cost b-matching in the bipartite graph induced between T and the set of all locations S, with overall unit capacity on either side. That is, the capacity for each node in T is $1/|T|$ and the capacity for each node in S is $1/n$. So, if we denote by $d_{i,j}$ the distance between locations i and j, we let $M(T)$ correspond to the following linear program.

$$\begin{aligned} M(T) := \min & \sum_{i \in T, j \in S} d_{i,j} \cdot x_{i,j} & & \qquad (M(\cdot)) \\ \text{s.t.} & \sum_{j \in S} x_{i,j} = \tfrac{1}{|T|} & \forall i \in T & \\ & \sum_{i \in T} x_{i,j} = \tfrac{1}{n} & \forall j \in S & \\ & x \geq 0 & & \end{aligned}$$

We emphasize that in the above LP, several servers in S (and likewise in T) may happen to be at the same point in the metric space, and hence there is a separate constraint for each such point j (and likewise i). Slightly abusing notation, we let $M(T)$ denote both the LP and its optimal value, when there is no scope for confusion.

2.2 Algorithm Description

The algorithm works as follows: at each time k, if $S_k \subseteq S$ is the current set of free servers, we compute the fractional assignment $M(S_k)$, and assign the next request randomly according to it. As argued above, since each free server location is equally likely to receive a request later (and therefore it is worth not matching it), it seems fair to leave each free server unmatched with equal probability. Put otherwise, it is only fair to match each of these servers with equal probability. Of course, matching any arriving request to a free server chosen uniformly at random can be a terrible strategy. In particular, it is easily shown to be $\Omega(\sqrt{n})$-competitive for n servers equally partitioned among a two-point metric. Therefore, to obtain good expected matching cost, we should bias servers' matching probability according to the arrived request, and in particular we should bias it according to $M(S_k)$. This intuition guides our algorithm FAIR-BIAS, and also inspires its name.

Algorithm 1 FAIR-BIAS

```
S_n ← S.                                   ▷ S_k is the set of free servers, with |S_k| = k.
for time step k = n, n − 1, ··· , 1 do
    compute optimal fractional matching M(S_k), denoted by x^{S_k}.
    upon arrival of request r_k = r do
        randomly choose server s from S_k, where s_i is chosen w/prob. p_i = n · x^{S_k}_{s_i,r}.
        assign r to s.
    end event
    S_{k−1} ← S_k \ {s}.
end for
```

A crucial property of our algorithm is that the set S_k of free servers at each time k happens to be a uniformly random k-subset of S. Recall that FAIR-BIAS assigns each arriving request according to the assignment $M(S_k)$. This means that to analyze the algorithm, it suffices to relate the optimal assignment cost OPT to the optimal assignment costs for uniformly random subsets S_k, as follows.

Lemma 2.2. *(Structure Lemma) For each time k, the set S_k is a uniformly-drawn k-subset of S; i.e., $S_k \sim \mathcal{U}_k$. Consequently, the algorithm's cost is*

$$\mathbb{E}[ALG] = \sum_{k=1}^{n} \mathbb{E}_{S_k \sim \mathcal{U}_k}[M(S_k)].$$

Proof. The proof of the first claim is a simple induction from n down to 1. The base case of S_n is trivial. For any k-subset $T = \{s_1, \cdots, s_k\} \subseteq S$,

$$\begin{aligned}\Pr[S_k = T] &= \sum_{s \in S \setminus T} \Pr[S_{k+1} = T \cup \{s\}] \cdot \Pr[r_{k+1} \text{ assigns to } s \mid S_{k+1} = T \cup \{s\}] \\ &= (n-k) \cdot \frac{1}{\binom{n}{k+1}} \cdot \frac{1}{k+1} = \frac{1}{\binom{n}{k}},\end{aligned}$$

where the second equality follows from induction and the fact that

$$\Pr[r_{k+1} \text{ assigned to } s \mid S_{k+1} = T \cup \{s\}] = \sum_{r \in S} x_{s,r}^{S_{k+1}} = \frac{1}{k+1}.$$

To compute the algorithm's cost, we consider some set $S_k = T$ of k free servers. Since the request $r_k = r$ is chosen with probability $1/n$, following which we match it to some free server $s \in S_k$ with probability $n \cdot x_{s,r}^{S_k}$, we find that the next edge matched by the algorithm has expected cost

$$\mathbb{E}[d_{s,r_k} \mid S_k = T] = \sum_{r} \frac{1}{n} \cdot \sum_{s \in T} n \cdot x_{s,r}^{T} \cdot d_{s,r} = M(T).$$

Therefore, the expected cost of the algorithm is indeed

$$\begin{aligned}\mathbb{E}[ALG] = \sum_{k=1}^{n} \mathbb{E}[d_{s,r_k}] &= \sum_{k=1}^{n} \sum_{T \in \binom{S}{k}} \Pr_{S_k \sim \mathcal{U}_k}[S_k = T] \cdot \mathbb{E}[d_{s,r_k} \mid S_k = T] \\ &= \sum_{k=1}^{n} \sum_{T \in \binom{S}{k}} \Pr_{S_k \sim \mathcal{U}_k}[S_k = T] \cdot M(T) = \sum_{k=1}^{n} \mathbb{E}_{S_k \sim \mathcal{U}_k}[M(S_k)].\end{aligned}$$ □

The structure lemma implies that we may assume from now on that the set of free servers S_k is drawn from $\mathcal{U}_k$. In what follows, unless stated otherwise, we have $S_k \sim \mathcal{U}_k$. More importantly, Lemma 2.2 implies that to bound our algorithm's competitive ratio by α, it suffices to show that $\sum_k \mathbb{E}[M(S_k)] \leq \alpha \cdot \mathbb{E}[\text{OPT}]$. This is exactly the approach we use in the following sections.

3 Bounds for General Metrics

In Section 4 we will show that algorithm FAIR-BIAS is $O(1)$-competitive for line metrics (and more generally tree metrics), by relying on variance bounds of the number of matches across tree edges in OPT and $M(S_k)$, our algorithm's guiding LP. For general metrics, if we first embed the metric in a low-stretch tree metric [12] (blowing up the expected cost of $\mathbb{E}[\text{OPT}]$ by $O(\log n)$) and run algorithm FAIR-BIAS on the obtained metric, we immediately obtain an $O(\log n)$-competitive algorithm. In fact, explicitly embedding the input metric in a tree metric is not necessary in order to obtain this result using our algorithm. By relying on an *implicit* tree embedding, we obtain the following lemma (mirroring the variance-based bound underlying our result for tree metrics). This lemma's proof is deferred to Appendix C.1.

Lemma 3.1. $\mathbb{E}_{S_k \sim \mathcal{U}_k}[M(S_k)] \leq \frac{O(\log n)}{\sqrt{nk}} \cdot \mathbb{E}[\mathrm{OPT}]$.

Summing over all values of $k \in [n]$, we find that FAIR-BIAS is $O(\log n)$-competitive on general metrics. While this bound is no better than that of Raghvendra's t-net algorithm for random order arrival [37] (and therefore for i.i.d arrivals), the result will prove useful in our overall bound for our algorithm. In Sections 3.1 and 3.2, we use a different balls-and-bins argument to decrease our bounds on the algorithm's competitive ratio considerably, to $O((\log \log n))^2)$, by considering the imbalance between number of requests and servers in the top k most requested locations. (The former quantity corresponds to the load of the k most loaded bins in a balls and bins process – motivating our interest in this process.) Finally, in Section 3.3, we combine this improved bound with the one from Lemma 3.1, summing different bounds for different ranges of k, to prove our main result: an $O((\log \log \log n)^2)$ bound for our algorithm's competitive ratio.

3.1 Balls and Bins: The Poisson Paradigm

For our results, we need some technical facts about the classical balls-and-bins process.

The following standard lemma from [34, Theorem 5.10] allows us to use the Poisson distribution to approximate monotone functions on the bins. For $i \in [n]$, let X_i^m be a random variable denoting the number of balls that fall into the i^{th} bin, when we throw m balls into n bins. Let Y_i^m be independent draws from the Poisson distribution with mean m/n.

Lemma 3.2. *Let $f(x_1, \cdots, x_n)$ be a non-negative function such that $\mathbb{E}[f(X_1^m, \cdots, X_n^m)]$ is either monotonically increasing or decreasing with m, then*

$$\mathbb{E}[f(X_1^m, \cdots, X_n^m)] \leq 2 \cdot \mathbb{E}[f(Y_1^m, \cdots, Y_n^m)].$$

A classic result states that for $m = n$ balls, the maximum bin load is $\Theta(\log n / \log \log n)$ w.h.p. (see e.g., [34, Lemmas 5.1, 5.12]). The following lemma is a partial generalization of this result. Its proof, which relies on the Poisson approximation of Lemma 3.2, is deferred to Appendix C.

Lemma 3.3. *Let n balls be thrown into n bins, each ball thrown independently and uniformly at random. Let L_j be the load of the j^{th} heaviest bin, and $N_k := \sum_{j \leq k} L_j$ be the number of balls in the k most loaded bins. There exists a constant $C_0 > 0$ such that for any $k \leq C_0 n$,*

$$\mathbb{E}[N_k] \geq \Omega\left(k \cdot \frac{\log(n/k)}{\log\log(n/k)}\right).$$

In the next lemma, whose proof is likewise deferred to Appendix C, we rely on a simple Chernoff bound to give a weaker lower bound for $\mathbb{E}[N_k]$ that holds for all $k \leq n/2$.

Lemma 3.4. *For sufficiently large n and any $k \leq n/2$, we have $\mathbb{E}[N_k] \geq 1.5k$.*

3.2 Relating Balls and Bins to Stochastic Metric Matching

We now bound the expected cost incurred by FAIR-BIAS at time k by appealing to the above balls-and-bins argument; this will give us our stronger bound of $O((\log \log n)^2)$. Specifically, we will derive another lower bound for $\mathbb{E}[\mathrm{OPT}]$ in terms of $\mathbb{E}_{S_k \sim \mathcal{U}_k}[M(S_k)]$. In our bounds we will partition the probability space $\mathcal{I}_n$ (corresponding to n i.i.d. requests) into disjoint parts, based on $\mathbb{T}_k$, the top k most frequently requested locations (with ties broken uniformly at random). By symmetry, $\Pr[\mathbb{T}_k = T] = 1/\binom{n}{k}$ for all $T \in \binom{S}{k}$. By coupling $\mathbb{T}_k$ with $\mathcal{U}_k$, we will lower-bound $\mathbb{E}[OPT]$ by $\mathbb{E}_{S_k \sim \mathcal{U}_k}[M(S_k)]$ times $\mathbb{E}[N_k] - k$, the expected imbalance between number of requests and servers in $\mathbb{T}_k$. Here $\mathbb{E}[N_k]$ is the expected occupancy of the k most loaded bins in the balls and bins process discussed in Section 3.1.

To relate $\mathbb{E}[OPT \mid \mathbb{T}_k = S_k]$ to $M(S_k)$, we will bound both these quantities by the cost of a min-cost perfect b-matching between S_k and $S \setminus S_k$; i.e., each vertex v has some (possibly fractional) demand b_v which is the extent to which it must be matched. To this end, we need the following simple lemma, which asserts that for any min-cost metric b-matching instance, there exists an optimal solution which matches co-located servers and requests maximally. We defer the lemma's proof, which follows from a local change argument and triangle inequality, to Appendix C.

Lemma 3.5. *Let $\mathcal{I}$ be a fractional min-cost bipartite metric b-matching instance, with demand ℓ_i and r_i for the servers and requests at location i. Then, there exists an optimal solution x for $\mathcal{I}$ with $x_{ii} = \min\{\ell_i, r_i\}$ for every point i in the metric.*

We are now ready to prove our main technical lemma, lower-bounding $\mathbb{E}[OPT \mid \mathbb{T}_k = S_k]$ in terms of $M(S_k)$ and the imbalance between number of requests of the k most requested locations, N_k, and the number of servers in those locations.

Lemma 3.6. *For all $k < n$ and $S_k \in \binom{S}{k}$, we have $\mathbb{E}[OPT \mid \mathbb{T}_k = S_k] \geq (\mathbb{E}[N_k] - k) \cdot M(S_k)$.*

Proof. Applying Lemma 3.5 to $M(S_k)$, we find that the optimal value of $M(S_k)$ is equal to that of a min-cost bipartite perfect b-matching instance with left vertices associated with S_k, each with demand $\frac{1}{k} - \frac{1}{n}$, and right vertices associated with $S \setminus S_k$, each with demand $\frac{1}{n}$.

We now turn to the meat of the proof – lower bounding $\mathbb{E}[OPT \mid \mathbb{T}_k = S_k]$. In particular, we will lower bound $\mathbb{E}[OPT \mid \mathbb{T}_k = S_k]$ by a min-cost bipartite perfect b-matching instance with left and right vertices as above (i.e., S_k and $S \setminus S_k$, respectively), but with uniform demands on both sides of at least $(\mathbb{E}[N_k]-k)/k$ and $(\mathbb{E}[N_k]-k)/(n-k)$, respectively. That is, the biregular min-cost bipartite b-matching whose cost C we showed lower bounds $M(S_k)$, but scaled by an $f \geq \frac{(\mathbb{E}[N_k]-k)}{k\cdot(1/k-1/n)}$ factor. Before proving this lower bound on $\mathbb{E}[OPT \mid \mathbb{T}_k = S_k]$, we note that it implies our desired bound, as

$$\mathbb{E}[\mathrm{OPT} \mid \mathbb{T}_k = S_k] \geq \frac{(\mathbb{E}[N_k] - k)}{k \cdot (1/k - 1/n)} \cdot C > (\mathbb{E}[N_k] - k) \cdot C = (\mathbb{E}[N_k] - k) \cdot M(S_k).$$

It remains to lower bound $\mathbb{E}[OPT \mid \mathbb{T}_k = S_k]$ in terms of such a biregular b-matching instance.

For the remainder of this proof, for notational simplicity we denote by Ω the probability space induced by conditioning on the event $\mathbb{T}_k = S_k$. To lower bound $\mathbb{E}_\Omega[OPT]$, we will provide a fractional perfect matching $\vec{x}$ of the expected instance (in Ω), and show that $\mathbb{E}_\Omega[OPT] \geq \sum_{ij} d_{ij} \cdot x_{ij}$, while $\sum_{j \in S\setminus S_k} x_{ij} \geq (\mathbb{E}[N_k]-k)/k$ for all $i \in S_k$ and $\sum_{i\in S} x_{ij} \geq (\mathbb{E}[N_k]-k)/(n-k)$ for all $j \in S \setminus S_k$. Consequently, focusing on edges $(i,j) \in S_k \times (S \setminus S_k)$, we find that the min-cost biregular bipartite perfect b-matching above lower bounds $\sum_{i\in S_k, j \in S\setminus S_k} d_{ij} \cdot x_{ij} \leq \sum_{ij} d_{ij} \cdot x_{ij} \leq \mathbb{E}_\Omega[OPT]$. We now turn to producing an $\vec{x}$ satisfying our desired properties.

For any two locations $i, j \in S$, we let $(i,j) \in OPT$ indicate that a request in location i is served by the server in location j. Let $p_{ij} := \Pr_\Omega[(i,j) \in OPT]$. We will show how small modifications to $\vec{p}$ will yield a fractional perfect matching $\vec{x}$ as discussed in the previous paragraph. Let Y_i be the number of requests at server i. By Lemma 3.5, we know that $(i,i) \in OPT \iff Y_i \geq 1$. So, $p_{ii} = \Pr_\Omega[Y_i \geq 1]$. Consequently, if we let $\Delta_{in}(j) := \sum_{j' \in S\setminus\{j\}} p_{j'j}$ and $\Delta_{out}(j) := \sum_{j'\in S\setminus\{j\}} p_{jj'}$, we have by Lemma 3.5 that $\Delta_{in}(j) = \Pr[Y_i \geq 1]$ and $\Delta_{out}(i) = \mathbb{E}[(Y_i - 1)^+]$ for all $i \in S$. (As usual, $x^+ = \max\{x, 0\}$.) Consequently, $\Delta_{in}(j) = \Delta_{in}(j')$ and $\Delta_{out}(j) = \Delta_{out}(j')$ for all $j, j' \in S \setminus S_k$, as $[Y_j \mid \Omega]$ and $[Y'_j \mid \Omega]$ are identically distributed. Moreover, as $\sum_{j\in S\setminus S_k} (\Delta_{in}(j) - \Delta_{out}(j)) = N_k - k \geq 0$, we find that $\Delta_{in}(j) - \Delta_{out}(j) \geq 0$ for all $j \in S \setminus S_k$. Now, suppose $Y_i \geq 1$ for all

$i \in S_k$ (conditioning on the complementary event is similar), we have by Lemma 3.5 that $p_{ji} = 0$ for all $i \in S_k$ and $j \in S \setminus \{i\}$. Moreover, by symmetry we have $\Delta_{out}(i) = (\mathbb{E}[N_k] - k)/k$ for all k locations $i \in S_k$. We now show how to obtain from $\vec{p}$ a fractional matching $\vec{x}$ between S_k and $S \setminus S_k$ of no greater cost than $\vec{p}$, such that $p_{jj'} = 0$ for all $j \neq j' \in S \setminus S_k$ and such that the values $\Delta_{in}(j) - \Delta_{out}(j)$ are unchanged for all $j \in S$. Consequently, all (simple) edges in the support of $\vec{x}$ go between S_k and $S \setminus S_k$, and $\Delta_{out}(i) = (\mathbb{E}[N_k] - k)/k$ for all $i \in S_k$ and $\Delta_{in}(j) = (\mathbb{E}[N_k] - k)/(n-k)$ for all $j \in S \setminus S_k$, yielding our desired lower bound on $\mathbb{E}_\Omega[OPT]$ in terms of a biregular bipartite b-matching instance.

We start by setting $\vec{x} \leftarrow \vec{p}$. While there exists a pair $j \neq j' \in S \setminus S_k$ with $x_{j'j} > 0$, we pick such a pair. As $\Delta_{in}(j) - \Delta_{out}(j) \geq 0$, there must also be some flow coming into j. We follow a sequence of edges $j_1 \leftarrow j_2 \leftarrow j_3 \leftarrow \dots$ with each $j_r \in S \setminus S_k$ and with $x_{j_r j_{r-1}} > 0$ until we either repeat some $j_r \in S\setminus$ or reach some j_r with $x_{ij_r}0$ for some $i \in S$. (Note that one such case must happen, as $\Delta_{in}(j) - \Delta_{out}(j) \geq 0$ for all $j \in S \setminus S_k$.) If we repeat a vertex, j_r, we only consider the sequence of nodes given by the obtained cycle, $j_1 \leftarrow j_2 \leftarrow j_3 \cdots \leftarrow j_r = j_1$. Let $\epsilon = \min_r x_{j_r j_{r-1}}$ be the smallest $x_{jj'}$ in our trail. If we repeated a vertex, we found a cycle, and we decrease $x_{jj'}$ by ϵ for all consecutive j, j' in the cycle. If we found some $i \in S$ and $x_{ij_r} > 0$, we decrease all $x_{jj'}$ values along the path (including x_{ij_r}) by ϵ and increase x_{ij_1} by ϵ. In both cases, we only decrease the cost of $\vec{x}$ (either trivially, or by triangle inequality) and we do not change $\Delta_{in}(j) - \Delta_{out}(j)$ for any $j \in S$, while decreasing $\sum_{j \neq j' \in S \setminus S_k} x_{jj'}$. As the initial x-values are all rational, repeating the above terminates, with the above sum equal to zero, which implies a biregular fractional solution $\vec{x}$ as required. The lemma follows. □

Coupling the distribution of $\mathbb{T}_k$ and the set of k free servers, we obtain the following.

Lemma 3.7. $\mathbb{E}_{S_k \sim \mathcal{U}_k}[M(S_k)] \leq \mathbb{E}[\mathrm{OPT}]/(\mathbb{E}[N_k] - k)$.

Proof. Taking expectations over $S_k \sim \mathcal{U}_k$, we obtain our claimed bound.

$$\begin{aligned}
\mathbb{E}_{S_k \sim \mathcal{U}_k}[M(S_k)] &= \sum_{S_k \in \binom{S}{k}} \frac{1}{\binom{n}{k}} \cdot M(S_k) && \text{defn. of } \mathcal{U}_k \\
&\leq \sum_{S_k \in \binom{S}{k}} \frac{1}{\binom{n}{k}} \frac{1}{(\mathbb{E}[N_k] - k)} \cdot \mathbb{E}[\mathrm{OPT} \mid \mathbb{T}_k = S_k] && \text{Lemma 3.6} \\
&= \frac{1}{(\mathbb{E}[N_k] - k)} \cdot \mathbb{E}[\mathrm{OPT}]. && \Pr[\mathbb{T}_k = S_k] = \frac{1}{\binom{n}{k}}. \qquad \square
\end{aligned}$$

Plugging in the lower bounds of Lemmas 3.3 and 3.4 for the top k most loaded bins' loads, $\mathbb{E}[N_k]$, we obtain the following bounds on FAIR-BIAS's per-step cost in terms of $\mathbb{E}[OPT]$.

Lemma 3.8. *For C_0 a constant as in Lemma 3.3, there exists a constant C such that*

$$\mathbb{E}_{S_k \sim \mathcal{U}_k}[M(S_k)] \leq \begin{cases} C \cdot \frac{\log\log(n/k)}{k \log(n/k)} \cdot \mathbb{E}[\mathrm{OPT}] & \text{if } k < C_0 n \\ \frac{2}{k} \cdot \mathbb{E}[\mathrm{OPT}] & \text{if } C_0 n \leq k \leq n/2. \end{cases}$$

The following lemma allows us to leverage Lemma 3.8, as it allows us to focus on $\mathbb{E}_{S_k \sim \mathcal{U}_k}[M(S_k)]$ for $k \leq n/2$. Its proof relies on our characterization of $M(S_k)$ in terms of a balanced b-matching instance between S_k and $S \setminus S_k$ as in the proof of Lemma 3.6, which implies that $M(S_k) \leq M(S_{n-k})$ for all $k \leq n/2$. Its proof is deferred to Appendix C.

Lemma 3.9. $\sum_{k=1}^{n} \mathbb{E}_{S_k \sim \mathcal{U}_k}[M(S_k)] \leq 2 \cdot \sum_{k=1}^{n/2} \mathbb{E}_{S_k \sim \mathcal{U}_k}[M(S_k)]$.

Using our upper bound on $\mathbb{E}_{S_k\sim\mathcal{U}_k}[M(S_k)]$ of Lemma 3.8 and summing the two ranges of $k \leq n/2$ in Lemma 3.9 we find that FAIR-BIAS is $O((\log\log n)^2)$ competitive. We do not elaborate on this here, as we obtain an even better bound in the following section.

3.3 Our Main Result

We are now ready to prove our main result, by combining our per-step cost bounds given by our balls and bins argument (Lemma 3.8) and our implicit tree embedding argument (Lemma 3.1).

Theorem 3.10. *Algorithm* FAIR-BIAS *is $O((\log\log\log n)^2)$-competitive for the online bipartite metric matching problem under i.i.d arrivals on general metrics.*

Proof. By the structure lemma (Lemma 2.2) and Lemma 3.9, we have that

$$\mathbb{E}[ALG] = \sum_{k=1}^{n} \mathbb{E}_{S_k\sim\mathcal{U}_k}[M(S_k)] \leq 2 \cdot \sum_{k=1}^{n/2} \mathbb{E}_{S_k\sim\mathcal{U}_k}[M(S_k)]. \tag{1}$$

We use the three bounds from Lemma 3.1 and Lemma 3.8 for different ranges of k to bound the above sum. Specifically, by relying on Lemma 3.1 for $k \leq n/\log^2 n$, we have that

$$\begin{aligned}\sum_{k=1}^{n/\log^2 n} \mathbb{E}_{S_k\sim\mathcal{U}_k}[M(S_k)] &\leq \sum_{k=1}^{n/\log^2 n} \frac{O(\log n)}{\sqrt{nk}} \cdot \mathbb{E}[OPT] \\ &\leq O\left(\sqrt{\frac{n}{\log^2 n}} \cdot \frac{\log n \cdot \mathbb{E}[OPT]}{\sqrt{n}}\right) = O(1) \cdot \mathbb{E}[OPT].\end{aligned}$$

Next, by the first bound of Lemma 3.8 applied to $k \in [n/\log^2 n, C_0 n]$, we have that

$$\begin{aligned}\sum_{k=n/\log^2 n}^{C_0 n} \mathbb{E}_{S_k\sim\mathcal{U}_k}[M(S_k)] &\leq \sum_{k=n/\log^2 n}^{C_0 n} \frac{O(\log\log(n/k))}{k \cdot \log(n/k)} \cdot \mathbb{E}[OPT] \\ &\leq O\left(-(\log\log(n/k))^2\Big|_{n/\log^2 n}^{C_0 n}\right) \cdot \mathbb{E}[OPT] \\ &= O((\log\log\log n)^2) \cdot \mathbb{E}[OPT].\end{aligned}$$

Finally, by the second bound of Lemma 3.8 applied to $k \geq C_0 n$, we have that

$$\sum_{k=C_0 n}^{n/2} \mathbb{E}_{S_k\sim\mathcal{U}_k}[M(S_k)] \leq \sum_{C_0 n}^{n/2} \frac{2}{k} \cdot \mathbb{E}[OPT] \leq O\left(\log\left(\frac{n/2}{C_0 n}\right)\right) \cdot \mathbb{E}[OPT] = O(1) \cdot \mathbb{E}[OPT].$$

Combining all three bounds with Equation (1), the theorem follows. □

4 A Simple $O(1)$ Bound for Tree Metrics

In this section we show the power of the structure lemma, by analyzing FAIR-BIAS on tree metrics. Recall that a *tree metric* is defined by shortest-path distances in a tree $T = (V, E)$, with edge lengths d_e. By adding zero-length edges, we may assume that the tree has n leaves, and that servers are on the leaves of the tree. For any edge e in the tree, deleting this edge creates two components $T_1(e)$ and $T_2(e)$; denote by $T_1(e)$ the component with fewer servers/leaves. Let n_e denote the number of leaves on this smaller side, $T_1(e)$. Hence $n_e \leq n/2$ for all edges e.

We now lower bound $\mathbb{E}$[OPT], by considering the mean average deviation of the number of requests which arrive in $T_1(e)$ for each edge e.

Lemma 4.1. *The expected optimal matching cost in a tree metric on $n \geq 2$ vertices is at least $\mathbb{E}[\mathrm{OPT}] \geq \frac{1}{2} \cdot \sum_{e \in T} d_e \cdot \sqrt{n_e}$.*

Proof. Let X_e denote the number of requests that arrive in the component with fewer leaves, $T_1(e)$. Every matching will match at least $|X_e - n_e| = |X_e - \mathbb{E}[X_e]|$ requests across the edge e (with the equality due to the uniform IID arrivals). Summing over all edges and taking expectations, we find that

$$\mathbb{E}[\mathrm{OPT}] \geq \sum_e d_e \cdot \mathbb{E}\big[|X_e - n_e|\big] = \sum_e d_e \cdot \mathbb{E}\big[|X_e - \mathbb{E}[X_e]|\big]. \tag{2}$$

It remains to lower bound $\mathbb{E}[|X_e - \mathbb{E}[X_e]|]$, the mean average deviation of X_e. Observe that $X_e \sim \mathrm{Bin}(n, n_e/n)$, with $n_e \in [1, n-1]$. The following probabilistic bound appears in [5, Theorem 1]:

Claim 4.2. *Let $Y \sim Bin(n,p)$, with $n \geq 2$ and $p \in [1/n, 1-1/n]$. Then, we have both*

$$\mathbb{E}|Y - \mathbb{E}Y| \geq \mathrm{std}(Y)/\sqrt{2},$$

(Note that convexity implies that $\mathbb{E}|Y - \mathbb{E}Y| \leq \mathrm{std}(Y)$ holds for all distributions, so this is a partial converse.) Applying Claim 4.2 to our case, where $p = n_e/n \in [1/n, 1-1/n]$,

$$\mathbb{E}[|X_e - \mathbb{E}X_e|] \geq \mathrm{std}(X_e)/\sqrt{2} = \sqrt{n_e(1-n_e/n)/2} \geq \sqrt{n_e/4},$$

where the second inequality follows from $n_e \leq n/2$. Combined with (2), the lemma follows. □

To upper bound $\mathbb{E}[M(S_k)]$, we again consider the mean average deviation of the number of requests in $T_1(e)$, but this time when drawing k *i.i.d.* samples. First, we need to bound the cost of $M(S_k)$ for a set S_k resulting from k draws *without replacement* by the cost for a multiset obtained by taking k i.i.d. draws *with replacement.*

Lemma 4.3. *(Replacement Lemma) For all S and $k \in [|S|]$, we have*

$$\mathbb{E}_{S_k \sim \mathcal{U}_k}[M(S_k)] \leq \mathbb{E}_{S_k \sim \mathcal{I}_k}[M(S_k)].$$

We defer the proof of this lemma to Appendix B, where we prove a more general statement regarding stochastic convex optimization with constraints and coefficients determined by elements of a set chosen uniformly with and without replacement. Armed with this lemma, it suffices to bound $\mathbb{E}_{S_k \sim \mathcal{I}_k}[M(S_k)]$ from above, which we do in the following.

Lemma 4.4. $\mathbb{E}_{S_k \sim \mathcal{I}_k}[M(S_k)] \leq \sum_{e \in T} d_e \cdot \sqrt{n_e/(kn)}$.

Proof. Fix some edge e and let $T_1(e)$ be its smaller subtree, containing $n_e \leq n/2$ leaves. Let $X_e \sim \mathrm{Bin}(k, n_e/n)$ be the random variable denoting the number of servers in $T_1(e)$ chosen in k i.i.d samples from S. For any given realization of S_k (and therefore of X_e) the fractional solution to $M(S_k)$ utilizes edges between the different subtrees of e by exactly $|X_e/k - n_e/n|$. Since this is a tree metric, we have

$$M(S_k) = \sum_{e \in T} d_e \cdot \left| \frac{X_e}{k} - \frac{n_e}{n} \right| = \sum_{e \in T} d_e \cdot \frac{1}{k} \cdot \left| X_e - \frac{k}{n} \cdot n_e \right| = \sum_{e \in T} d_e \cdot \frac{1}{k} \cdot |X_e - \mathbb{E}[X_e]|.$$

Taking expectations over S_k, and using the fact that the mean average deviation is always upper bounded by the standard deviation (by Jensen's inequality), we find that indeed

$$\mathbb{E}_{S_k \sim \mathcal{I}_k}[M(S_k)] = \sum_{e \in T} d_e \cdot \frac{1}{k} \cdot \mathbb{E}[|X_e - \mathbb{E}[X_e]|] \leq \sum_{e \in T} d_e \cdot \frac{1}{k} \cdot \mathrm{std}(X_e)$$

$$= \sum_{e \in T} d_e \cdot \frac{1}{k} \cdot \sqrt{k \cdot \frac{n_e}{n}\left(1 - \frac{n_e}{n}\right)} \leq \sum_{e \in T} d_e \cdot \sqrt{\frac{n_e}{k \cdot n}}. \qquad \square$$

Combining the replacement lemma (Lemma 4.3) with Lemmas 4.4 and 4.1, we obtain the following upper bound on $\mathbb{E}_{S_k \sim \mathcal{U}_k}[M(S_k)]$ in terms of $\mathbb{E}[OPT]$.

Lemma 4.5. $\mathbb{E}_{S_k \sim \mathcal{U}_k}[M(S_k)] \leq 2 \cdot \frac{\mathbb{E}[OPT]}{\sqrt{nk}}$.

We can now prove our simple result for tree metrics.

Theorem 4.6. *(Tree Bound) Algorithm* FAIR-BIAS *is* 4*-competitive on tree metrics with* $n \geq 2$ *nodes, if the requests are drawn from the uniform distribution.*

Proof. We have by the structural lemma (Lemma 2.2) and Lemma 4.5 that

$$\begin{aligned} \mathbb{E}[\text{ALG}] &= \sum_{k=1}^{n} \mathbb{E}[M(S_k)] \leq \sum_{k=1}^{n} 2 \cdot \frac{\mathbb{E}[OPT]}{\sqrt{nk}} \\ &\leq 2 \cdot \frac{\mathbb{E}[OPT]}{\sqrt{n}} \cdot \left(1 + \int_{x=1}^{n} \frac{1}{\sqrt{x}} dx\right) \leq 4 \cdot \mathbb{E}[OPT]. \qquad \square \end{aligned}$$

The above bound holds for all $n \geq 2$ (for $n = 1$ any algorithm is trivially 1 competitive). For n large, however, our proof yields an improved asymptotic bound of $\sqrt{2} \cdot e + o(1) \approx (3.845 + o(1))$, by relying on the asymptotic counterpart of Claim 4.2 in [5, Corollary 2], $\mathbb{E}|Y - \mathbb{E}Y| \geq \text{std}(Y)/(e/2 + o(1))$. Combining Theorem 4.6 with our transshipment argument (Lemma 2.1), we obtain a 9-competitive algorithm under any i.i.d. distribution on tree metrics on $n \geq 2$ nodes, and even better than 9-competitive algorithms for large enough n.

5 Open Questions

In this work, we presented algorithm FAIR-BIAS and proved that it is $O((\log\log\log n)^2)$-competitive for general metrics, and 9-competitive for tree metrics. Perhaps the first question is whether our algorithm (or indeed any algorithm) is $O(1)$ competitive for (known or unknown) i.i.d arrivals for general metrics. Indeed, we do not know of any instances where Algorithm FAIR-BIAS's performance is worse than $O(1)$ competitive. However, it is not clear how to extend our proofs to establish an $O(1)$ competitive ratio.

Another question is the relationship between the known and unknown i.i.d. models and the random order model. The optimal competitive ratios for the various arrival models for online problems can be sorted as follows (see e.g. [30, Theorem 2.1])

$$C.R.(Adversarial) \geq C.R.(Random\ Order) \geq C.R.(Unknown\ IID) \geq C.R.(Known\ IID).$$

For the online metric matching problem the best bounds known for the above are, respectively, $\Theta(n)$ [24, 27], $\Theta(\log n), O(\log n)$ (both [37]), and $O((\log\log\log n)^2)$ (this work). Given the lower bound of [37], our work implies that one or both of the inequalities in $C.R.(Random\ Order) \geq C.R.(Unknown\ IID) \geq C.R.(Known\ IID)$ is strict (and asymptotically so). It would be interesting to see which of these inequalities is strict, by either presenting a $o(\log n)$-competitive algorithm for unknown i.i.d or a $\omega((\log\log\log n)^2)$ lower bound for this model. For the line metric, given the lower bound of [14], our work implies that one of the three inequalities above must be strict. Understanding the exact relationships between these arrival models for this simple metric may prove useful in understanding the relationships between the different stochastic arrival models more broadly. Moreover, it would be interesting to study these questions for other combinatorial optimization problems with online stochastic arrivals.

Appendix

A Distribution over Server Locations (Transshipment Argument)

In this section, we show that the assumption that the requests are drawn from $\mathcal{U}$, the uniform distribution over server locations, is without loss of generality.

Lemma 2.1. *Given an α-competitive algorithm* $\mathrm{ALG}_{\mathcal{U}}$ *for the* uniform *distribution over server locations, $\mathcal{U}$, we can construct a $(2\alpha+1)$-competitive algorithm* $\mathrm{ALG}_{\mathcal{D}}$ *for any distribution $\mathcal{D}$.*

Proof. As before, we identify the set of servers S with the n points on the metric and let $r_1, \ldots, r_n$ be the requests that arrive according to the distribution $\mathcal{D}$. Define $p_i := \Pr_{r\sim\mathcal{D}}[r = i]$.

Consider the linear program defined by the transshipment problem between the distribution $\mathcal{D}$ to the uniform distribution on the servers S.

$$\begin{aligned} LP := \min \sum_{i,j} d_{i,j} \cdot x_{i,j} & \\ \text{s.t. } \sum_j x_{i,j} = p_i & \qquad \forall i \in \text{metric} \\ \sum_i x_{i,j} = \frac{1}{n} & \qquad \forall j \in S \\ x \geq 0 & \end{aligned}$$

Let $M = n \cdot LP$. Given a request sequence $\{r_1, \ldots, r_n\}$ drawn from $\mathcal{D}$, we create a coupled sequence $\{\tilde{r}_1, \ldots, \tilde{r}_n\}$ by moving an arrived request r_k at server location j to location i in the metric with probability $x_{i,j}/p_i$ Each server location $j \in S$ appears with probability $\sum_i x_{i,j} = \frac{1}{n}$ and hence the sequence $\{\tilde{r}_1, \ldots, \tilde{r}_n\}$ is distributed according to the uniform distribution $\mathcal{U}$. After this move, it matches the request according to $\mathrm{ALG}_{\mathcal{U}}$.

We bound this algorithm's cost as follows. First, the probability of a given request being moved from some location i to j is precisely $p_i \cdot x_{i,j}/p_i = x_{i,j}$. Summing up over all i, j, the expected movement cost for all n time steps is precisely $M = n \cdot LP$. Secondly, the expected cost of matching from $\tilde{r}_i$ is precisely $\mathbb{E}[\mathrm{ALG}_{\mathcal{U}}]$. By the triangle inequality, we can bound the total cost by the sum of the initial costs and the matching costs according to $\mathrm{ALG}_{\mathcal{U}}$, yielding the relation

$$\mathbb{E}[\mathrm{ALG}_{\mathcal{D}}] \leq \mathbb{E}[\mathrm{ALG}_{\mathcal{U}}] + M. \tag{3}$$

We use the same coupling as above, but in the other direction to relate $\mathrm{OPT}_{\mathcal{U}}$ to M. In particular, given a request sequence $\{r_1, \ldots, r_n\}$ drawn from $\mathcal{U}$, we create a coupled sequence $\{\tilde{r}_1, \ldots, \tilde{r}_n\}$ by moving an arrived request r_k at server location j to location i in the metric with probability $n \cdot x_{i,j}$. Now $\Pr[\tilde{r}_k = i] = \frac{1}{n} \cdot \sum_j n \cdot x_{i,j} = \sum_j x_{i,j} = p_i$. That is, the resulting distribution is $\mathcal{D}$. One way to bound the optimal solution for distribution $\mathcal{U}$ is to match request r_k to the match of $\tilde{r}_k$. As before, the expected movement cost to locations $\{\tilde{r}_1, \ldots, \tilde{r}_n\}$ is M, and by triangle inequality, we find that

$$\mathbb{E}[\mathrm{OPT}_{\mathcal{U}}] \leq \mathbb{E}[\mathrm{OPT}_{\mathcal{D}}] + M. \tag{4}$$

We now bound $\mathbb{E}[\mathrm{OPT}_{\mathcal{D}}]$ in terms of M. Each location i in the metric has an expected np_i appearances, who must therefore be matched an expected np_i many times. Each server, on the other hand, is matched precisely once in expectation. Therefore, the probabilities $p_{i,j}$ of an arrival at location i being matched to a server at location j constitute a feasible solution to $n \cdot LP$, and so must have $\sum_{i,j} d_{i,j} \cdot p_{i,j} \geq n \cdot LP = M$. Therefore, $\mathbb{E}[\mathrm{OPT}_{\mathcal{D}}]$ satisfies

$$\mathbb{E}[\mathrm{OPT}_{\mathcal{D}}] \geq M. \tag{5}$$

Combining equations (3), (4) and (5) with $\mathrm{ALG}_{\mathcal{U}}$'s α-competitiveness, we obtain our desired result.

$$\begin{aligned}
\mathbb{E}[\mathrm{ALG}_{\mathcal{D}}] &\leq \mathbb{E}[\mathrm{ALG}_{\mathcal{U}}] + M && \text{Equation (3)} \\
&\leq \alpha \cdot \mathbb{E}[\mathrm{OPT}_{\mathcal{U}}] + M && \mathrm{ALG}_{\mathcal{U}} \text{ is } \alpha\text{-comp.} \\
&\leq \alpha \cdot (\mathbb{E}[\mathrm{OPT}_{\mathcal{D}}] + M) + M && \text{Equation (4)} \\
&\leq (2\alpha + 1) \cdot \mathbb{E}[\mathrm{OPT}_{\mathcal{D}}]. && \text{Equation (5)}
\end{aligned}$$

□

B Stochastic Convex Optimization, with and without Replacement

In Lemma 4.3 we claimed that the expected cost of the linear program $M(S_k)$ for S_k chosen at random from the k-subsets of S is lower than its counterpart when S_k is obtained from k i.i.d draws from S. More succinctly, we claimed that $\mathbb{E}_{S_k \sim \mathcal{U}_k}[M(S_k)] \leq \mathbb{E}_{S_k \sim \mathcal{I}_k}[M(S_k)]$. In this section we prove a more general claim for any linear program (and more generally, any convex program), implying the above. Let S be some n-element set, and for any multiset T with all its elements taken from S, let $P(T)$ be the following convex program.

$$\begin{aligned}
P(T) := \min & f(x, \chi_T) && && (P(\cdot)) \\
\text{s.t. } & g_i(x, \chi_T) \leq 0 && \forall i \in [m] \\
& h_j(x, \chi_T) = 0 && \forall j \in [\ell]
\end{aligned}$$

Here $f(x, \chi_T)$ and all $g_i(x, \chi_T)$ are convex functions and $h_j(x, \chi_T)$ are affine in their arguments x and χ_T, and χ_T is the incidence vector of the multiset T. (That is, for any $s \in S$, we let $\chi_T(s)$ denote the number of appearances of s in T.) Note that $M(T)$ defined in Section 2.1 is a linear program of the above form. As such, the following lemma generalizes – and implies – Lemma 4.3.

Lemma B.1 (Replacement Lemma). *For any convex program P as above, we have*

$$\mathbb{E}_{S_k \sim \mathcal{U}_k}[P(S_k)] \leq \mathbb{E}_{S_k \sim \mathcal{I}_k}[P(S_k)].$$

Proof. Our proof relies on a coupling argument, starting with a refined partition of the probability space of $S_k \sim \mathcal{I}_k$. This space is partitioned into equiprobable events A_M for each ordered multiset M of size k supported in S, corresponding to M being sampled. For each ordered multiset M, we denote by $\mathrm{support}(M) := \{s \in S \mid s \in M\}$ the set of elements in M. Next, we denote by $\mathrm{SUP}(M) := \{T \in \binom{S}{k} \mid T \supseteq \mathrm{support}(M)\}$ the family of k-sets which contain M's elements (i.e., supersets of M's support). We will wish to "equally partition" the event A_M among the k-tuples in $\mathrm{SUP}(M)$. To this end, when M is sampled from $\mathcal{I}_k$, we roll a $|\mathrm{SUP}(M)|$-sided die labeled by the members of $\mathrm{SUP}(M)$. For any k-set $T \in \mathrm{SUP}(M)$, we denote by $A_{M,T}$ the event that M was sampled from $\mathcal{I}_k$ and the die-roll came out T, and for any k-tuple $T \in \binom{S}{k}$, we let $A_T := \bigcup_M A_{M,T}$. It is easy to verify that by symmetry we have $Pr[A_T] = 1/\binom{|S|}{k}$ for every $T \in \binom{S}{k}$.

We now wish to couple the above refinement of the probability space of $\mathcal{I}_k$ and the optimal solution to $P(S_k)$ with their counterpart under $\mathcal{U}_k$. We will need the following claim.

Claim B.2. *For all k-set $T \in \binom{S}{k}$ and element $s \in T$, we have $\mathbb{E}_{S_k \sim \mathcal{I}_k}[\chi_{S_k}(s) \mid A_T] = 1$.*

Proof. By definition, each non-empty $A_{M,T} \subseteq A_T$ satisfies $\mathbb{E}_{S_k \sim \mathcal{I}_k}[\sum_{s \in T} \chi_{S_k}(s) \mid A_{M,T}] = k$, since any ordered multiset M of size k with $\mathrm{SUP}(M) \ni T$ has all its elements in T. Therefore, taking total expectation over M with $\mathrm{SUP}(M) \ni T$, we get $\mathbb{E}_{S_k \sim \mathcal{I}_k}[\sum_{s \in T} \chi_{S_k}(s) \mid A_T] = k$. Therefore, by symmetry, we find that indeed each of the k elements $s \in T$ has $\mathbb{E}_{S_k \sim \mathcal{I}_k}[\chi_{S_k}(s) \mid A_T] = 1$. □

Now, consider some k-set $T \in \binom{S}{k}$. For any ordered multiset of k elements M such that $SUP(M) \ni T$, denote by $x^M \in \arg\min P(M)$ a solution of $P(M)$ of minimum cost. By definition, for each $i \in [m]$ we have that $g_i(x^M, \chi_M) \leq 0$ and for each $j \in [\ell]$ we have that $h_j(x^M, \chi_M) = 0$. Therefore, if we let $y^T := \mathbb{E}_{M \sim \mathcal{I}_k}[x^M \mid A_T]$ be the "average" optimal solution for $P(M)$ over all M with $SUP(M) \ni T$, then by Jensen's inequality and convexity of g_i, we have that

$$\begin{aligned} 0 &\geq \mathbb{E}_{M \sim \mathcal{I}_k}[g_i(x^M, \chi_M) \mid A_T] && \text{linearity} \\ &\geq g_i(\mathbb{E}_{M \sim \mathcal{I}_k}[x^M \mid A_T], \mathbb{E}_{M \sim \mathcal{I}_k}[\chi_M \mid A_T]) && \text{Jensen's Ineq.} \\ &= g_i(y^T, \chi_T). && \text{Claim B.2} \end{aligned}$$

Similarly, we have that $h_j(y^T, \chi_T) = \mathbb{E}_{M \sim \mathcal{I}_k}[h_j(x^M, \chi_M) \mid A_T] = 0$ for all $j \in [\ell]$, as h_j is affine. We conclude that y^T is a feasible solution to $P(T)$, and therefore $f(y^T, \chi_T) \geq P(T)$. Again appealing to Jensen's inequality, recalling that $y^T = \mathbb{E}_{M \sim \mathcal{I}_k}[x^M \mid A_T]$ and that $\mathbb{E}_{M \sim \mathcal{I}_k}[\chi_M \mid A_T] = \chi_T$ by Claim B.2, we find that

$$\mathbb{E}_{M \sim \mathcal{I}_k}[f(x^M, \chi_M) \mid A_T] \geq f(y^T, \chi_T) \geq P(T).$$

The lemma follows by total expectation over M, relying on $\Pr[A_T] = 1/\binom{|S|}{k}$ for each $T \in \binom{S}{k}$.

$$\begin{aligned} \mathbb{E}_{M \sim \mathcal{I}_k}[P(M)] &= \sum_{T \in \binom{S}{k}} \mathbb{E}_{M \sim \mathcal{I}_k}[P(M) \mid A_T] \cdot \Pr[A_T] \\ &\geq \sum_{T \in \binom{S}{k}} P(T) \cdot \Pr[A_T] = \mathbb{E}_{T \sim \mathcal{U}_k}[P(T)]. \end{aligned}$$

□

C Deferred Proofs of Section 3

In this section we provide the proofs deferred from Section 3.

C.1 Implicit Tree Embedding

In Section 4, we proved that algorithm FAIR-BIAS is $O(1)$-competitive on tree metrics. Therefore, as noted in Section 3, using tree embeddings and applying algorithm FAIR-BIAS to the points according to distances in the obtained tree embedding yields an $O(\log n)$-competitive algorithm for general metrics. Here we present an upper bound on FAIR-BIAS's expected per-arrival cost which implies the same competitive bound, by relying on an *implicit* tree embedding.

Lemma 3.1. $\mathbb{E}_{S_k \sim \mathcal{U}_k}[M(S_k)] \leq \frac{O(\log n)}{\sqrt{nk}} \cdot \mathbb{E}[\mathrm{OPT}]$.

Proof. For our proof we rely on low-stretch tree embeddings [12]. Given an n-point metric with distances $d_{i,j}$, this embedding is a distribution $\mathcal{D}$ over tree metrics T over the same point set, with tree distances $d^T_{i,j}$ satisfying the following for any two points i, j in the metric.

$$d_{i,j} \leq d^T_{i,j}. \tag{6}$$

$$\mathbb{E}_{T\sim\mathcal{D}}[d^T_{i,j}] \leq O(\log n)\cdot d_{i,j}. \tag{7}$$

For such a tree metric T, let $M^T(S)$ denote $M(S)$ with the distances $d_{i,j}$ replaced by $d^T_{i,j}$. (As before, we also let this denote the optimum value of this program.) By (6) we immediately have that $M(S)\leq M^T(S)$ for any set S, as any solution $\vec{x}$ to $M^T(S)$ is feasible for $M(S)$ and has lower cost for this latter metric, $\sum_{i,j} x_{i,j}\cdot d_{i,j} \leq \sum_{i,j} x_{i,j}\cdot d^T_{i,j}$. Consequently, we have

$$M(S)\leq \mathbb{E}_{T\sim\mathcal{D}}[M^T(S)]. \tag{8}$$

Next, we denote by OPT^T the optimum cost of the min-cost perfect matching of the requests to servers for distances $d^T_{i,j}$. By Lemma 4.5 we have that for a tree metric T

$$\mathbb{E}_{S_k\sim\mathcal{U}_k}[M^T(S_k)] \leq \frac{4\cdot\mathbb{E}[OPT^T]}{\sqrt{nk}}. \tag{9}$$

Finally, for any realization of requests, the minimum-cost matching of requests to servers under $d_{i,j}$ has expected cost (over the choice of T) at most $O(\log n)$ times higher under $d^T_{i,j}$, by (7). Therefore, by a coupling argument we get the following bound on $\mathbb{E}_{T\sim\mathcal{D}}\mathbb{E}[OPT^T]$ in terms of $\mathbb{E}[OPT]$.

$$\mathbb{E}_{T\sim\mathcal{D}}[OPT^T]\leq O(\log n)\cdot\mathbb{E}[OPT]. \tag{10}$$

Combining Equations (8), (9) and (10), we obtain our desired bound.

$$\mathbb{E}_{S_k\sim\mathcal{U}_k}[M(S_k)] \leq \mathbb{E}_{T\sim\mathcal{D}}\mathbb{E}_{S_k\sim\mathcal{U}_k}[M^T(S_k)] \leq \frac{4\cdot\mathbb{E}_{T\sim\mathcal{D}}\mathbb{E}[OPT^T]}{\sqrt{nk}} \leq \frac{O(\log n)\cdot\mathbb{E}[OPT]}{\sqrt{nk}}. \qquad \square$$

C.2 Load of k Most Loaded Bins

Here we prove our lower bounds on the sum of loads of the k most loaded bins in a balls and bins process with n balls and bins.

Lemma 3.3. *Let n balls be thrown into n bins, each ball thrown independently and uniformly at random. Let L_j be the load of the j^{th} heaviest bin, and $N_k := \sum_{j\leq k} L_j$ be the number of balls in the k most loaded bins. There exists a constant $C_0>0$ such that for any $k\leq C_0 n$,*

$$\mathbb{E}[N_k]\geq \Omega\left(k\cdot\frac{\log(n/k)}{\log\log(n/k)}\right).$$

Proof. Let $t=\frac{\log(n/k)}{\log\log(n/k)}$, and define

$$f(x_1,\cdots,x_n)=\begin{cases}1 & \text{if the } k^{th} \text{ largest number in } x_1,\cdots,x_n \text{ is less than } t/2\\ 0 & \text{otherwise}\end{cases}.$$

Clearly, the function $f(x_1,\cdots,x_n)$ satisfies the condition in Lemma 3.2, i.e., $f(x_1,\cdots,x_n)$ is non-negative and $\mathbb{E}[f(X_1^m,\cdots,X_n^m)]$ is monotonically decreasing with m. Since we have an equal number of balls and bins, we consider the case $m=n$. We abbreviate X_i^n to X_i and Y_i^n to Y_i. Let M_k be the k^{th} largest number among $Y_1,\cdots,Y_n$. Applying Lemma 3.2,

$$\Pr[L_k<t/2]=\mathbb{E}[f(X_1,\cdots,X_n)]\leq 2\cdot\mathbb{E}[f(Y_1,\cdots,Y_n)]=2\cdot\Pr[M_k<t/2].$$

Define the indicator variable $Z_i := \mathbf{1}_{(Y_i \geq t/2)}$, and observe that $\Pr[M_k < t/2] = \Pr[\sum_i Z_i < k]$. We bound the latter via a Chernoff bound, so we need a lower bound on $\mathbb{E}[\sum_i Z_i]$.

$$\mathbb{E}[\sum_i Z_i] = n \cdot \Pr[Y_i \geq t/2] \geq n \cdot \Pr[Y_i = t/2] \overset{(a)}{=} \frac{n}{e(t/2)!} \overset{(b)}{\geq} \frac{4n}{t!} \overset{(c)}{\geq} 4k. \tag{11}$$

The equality (a) uses the definition of the Poisson distribution, the inequality (b) uses that $t! \geq 4e(t/2)!$ for sufficiently large t. For inequality (c), we know $t! \leq \sqrt{t}/e\,(t/e)^t$ from Stirling's approximation, and so when n/k is sufficiently large, plugging in $t = \frac{\log(n/k)}{\log\log{(n/k)}}$ gives

$$\log(t!) \leq (t + 1/2)\log t - t - 1 \leq t \log t \leq \log(n/k).$$

Putting things together, and using a Chernoff bound, we get

$$\Pr\left[L_k < t/2\right] \leq 2 \cdot \Pr\left[M_k < t/2\right] = 2 \cdot \Pr[\sum_i Z_i < k] \leq 2e^{-\frac{(3/4)^2 .4k}{2}} \leq 2e^{-k}.$$

The lemma then follows directly, as

$$\mathbb{E}[N_k] \geq \mathbb{E}\left[N_k \mid L_k \geq t/2\right] \cdot \Pr\left[L_k \geq t/2\right] \geq k \cdot (t/2) \cdot (1 - 2e^{-k}) = \Omega\Big(\frac{k \cdot \log(n/k)}{\log\log(n/k)}\Big). \qquad \square$$

The following simple lemma states that in the min cost perfect matching, we can always match requests and servers in the same location as much as possible. That is, $x_{ii} = \frac{1}{n}$ for every requested location i.

Lemma 3.4. *For sufficiently large n and any $k \leq n/2$, we have $\mathbb{E}[N_k] \geq 1.5k$.*

Proof. In expectation, there are $n\,(1 - 1/n)^n \sim n/e$ empty bins, thus on average one would expect $1/(1-1/e) > 1.5$ balls in each non-empty bin. To make this intuition formal, let $t = (1-1/e+0.01)n$ and define

$$f(x_1, \cdots, x_n) = \begin{cases} 1 & \text{if more than } t \text{ of } x_1, \cdots, x_n \text{ are greater than } 0 \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to verify that the function $f(x_1, \cdots, x_n)$ is non-negative and $\mathbb{E}[f(X_1^m, \cdots, X_n^m)]$ is monotonically increasing in m. Define the variable $Z_i := \mathbf{1}_{(Y_i > 0)}$; then $Z_i \sim \text{Bernoulli}(1 - 1/e)$. Lemma 3.2 and a Chernoff bound now give that for sufficiently large n,

$$\mathbb{E}[f(X_1, \cdots, X_n)] \leq 2 \cdot \mathbb{E}[f(Y_1, \cdots, Y_n)] = 2 \cdot \Pr\left[\sum_i Z_i > tn\right] \leq 2e^{-\frac{0.01^2 \cdot (1-1/e)n}{2}} < 0.01.$$

Hence

$$\mathbb{E}[N_t] \geq \mathbb{E}\left[N_t \mid f(X_1, \cdots, X_n) = 0\right] \cdot \Pr\left[f(X_1, \cdots, X_n) = 0\right] \geq n \cdot (1 - 0.01) = 0.99n.$$

Finally, for $k \leq n/2 (\leq t)$, we have that indeed $\frac{\mathbb{E}[N_k]}{k} \geq \frac{\mathbb{E}[N_t]}{t} \geq \frac{0.99n}{(1-1/e+0.01)n} \geq \frac{3}{2}$. $\square$

C.3 Further Deferred Proofs

Lemma 3.5. *Let $\mathcal{I}$ be a fractional min-cost bipartite metric b-matching instance, with demand ℓ_i and r_i for the servers and requests at location i. Then, there exists an optimal solution x for $\mathcal{I}$ with $x_{ii} = \min\{\ell_i, r_i\}$ for every point i in the metric.*

Proof. Fix an optimal solution x^* of $\mathcal{I}$ of maximum $\sum_i x^*_{ii}$ among optimal solutions of $\mathcal{I}$. Suppose for contradiction that there exists some $i \in S_k$ such that $x^*_{ii} < \min\{\ell_i, r_i\}$. WLOG $\ell_i \leq r_i$ and so there exists some locations j, j' such that $x^*_{ij} > 0$ and $x^*_{j'i} > 0$. Let $\epsilon = \min\{x^*_{ij}, x^*_{j'i}\}$. Consider the solution $\tilde{x}$ obtained from x^* by increasing x^*_{ii} and $x^*_{j'j}$ by ϵ and decreasing x^*_{ij} and $x^*_{j'i}$ by ϵ. This $\tilde{x}$ is a feasible solution to $\mathcal{I}$ (as sums of the form $\sum_i x_{ij}$ and $\sum_j x_{ij}$ are unchanged and $\tilde{x} \geq 0$). Moreover, we find that

$$\begin{aligned}\sum_{ij} d_{ij} \cdot \tilde{x}_{ij} &= \left(\sum_{ij} d_{ij} \cdot x^*_{ij}\right) + \epsilon \cdot (d_{ii} + d_{jj'} - d_{ij} - d_{ij'}) \\ &= OPT(\mathcal{I}) + \epsilon \cdot (d_{jj'} - d_{ij} - d_{ij'}) \leq OPT(\mathcal{I}),\end{aligned}$$

by triangle inequality. That is, $\tilde{x}$ is an optimal solution to $\mathcal{I}$ with a higher $\sum_i x_{ii}$ than x^*, contradicting our assumption. The lemma follows. □

Lemma 3.9. $\sum_{k=1}^{n} \mathbb{E}_{S_k \sim \mathcal{U}_k}[M(S_k)] \leq 2 \cdot \sum_{k=1}^{n/2} \mathbb{E}_{S_k \sim \mathcal{U}_k}[M(S_k)]$.

Proof. As noted in the proof of Lemma 3.7, by Lemma 3.5, the optimal value of $M(S_k)$ is equal to that of a min-cost bipartite perfect b-matching instance with left vertices associated with S_k with demand $\frac{1}{k} - \frac{1}{n}$ and right vertices associated with $S \setminus S_k$ with demand $\frac{1}{n}$. Similarly, $M(S \setminus S_k)$ is equal to the same, but with each $i \in S_k$ having demand $\frac{1}{n}$ and each $i \in S \setminus S_k$ having demand $\frac{1}{n-k} - \frac{1}{n}$. That is, these programs are just scaled versions of each other, and we we have that for any $k \leq n/2$,

$$M(S_k) = \frac{1/k - 1/n}{1/n} \cdot M(S \backslash S_k) = \left(\frac{n}{k} - 1\right) \cdot M(S \backslash S_k) \geq M(S \backslash S_k).$$

Consequently, taking expectation over S_k (equivalently, over $S \setminus S_k$), we find that for any $k \leq n/2$, we have $E_{S_k \sim \mathcal{U}_k}[M(S_k)] \geq E_{S_{n-k} \sim \mathcal{U}_{n-k}}[M(S_{n-k})]$. The lemma follows. □

D Max Weight Perfect Matching Problem in i.i.d Model

Here we prove that, with a small modification, FAIR-BIAS achieves the optimal competitive ratio, i.e 1/2, in the max weight perfect matching problem introduced in [7]. Here, rather than compute a minimum cost perfect matching, we are tasked with computing a maximum weight perfect matching, which need not correspond to a metric. Since we are now in a maximization problem and we are no longer in a metric space, we will not make the assumption that the distribution of all requests is uniform among all servers. Moreover, we make the following modification to our algorithm: in each round of FAIR-BIAS, instead of finding a min cost perfect matching, we would find the max weight perfect matching. Correspondingly, we change the notation for $M(T)$: instead of being a min cost perfect b-matching induced by the set of free servers T and requests R, now $M(T)$ refers to the *max* weight perfect b-matching between the set of free servers T and requests R. More formally, we have

$$M(T) := \max \sum_{i \in T, j \in R} w_{i,j} \cdot x_{i,j} \tag{12}$$

$$\begin{aligned} \text{s.t.} \sum_{j\in T} x_{i,j} &= \frac{1}{|T|} && \forall i \in T \\ \sum_{i\in R} x_{i,j} &= p_i && \forall j \in R \\ x &\geq 0. \end{aligned}$$

Generalizing FAIR-BIAS, if S_k is the realized set of free servers and x^{S_k} an optimal solution to $M(S_k)$, then upon arrival of a request at location i (which happens with probability p_i), we randomly pick a server s to match this request to, chosen with probability $x^{S_k}_{i,s}/p_i$.

Difference compared to [7]. We note that Chang et al. [7] used a similar LP to $M(T)$. Essentially, they used $M(S)$, the program obtained by considering *all* servers (and not just free ones). Following [13, 20], they refer to this as the optimum of the "expected graph". Their algorithm picks a preferred server among all servers with probability $x^{S_k}_{r,s}/p_i$. If this server is already matched, in order to output a perfect matching they randomly (i.e., uniformly) pick an alternative server to match to. Our algorithm does not need to fall back on a second random choice, as it only picks a server among free servers. As we shall see, our algorithm's analysis follows rather directly from our analysis of FAIR-BIAS for the minimization variant.

A key observation is that the structure lemma (Lemma 2.2) still holds for our maximization variant of FAIR-BIAS. We restate it here.

Claim D.1. *(Structure Lemma, Restated) For each time k, the set S_k is a uniformly-drawn k-subset of S; i.e., $S_k \sim \mathcal{U}_k$. Consequently, the weight of the algorithm's output matching is*

$$\mathbb{E}[ALG] = \sum_{k=1}^{n} \mathbb{E}_{S_k\sim \mathcal{U}_k}[M(S_k)].$$

Claim D.1 holds due to the same argument in Lemma 2.2. Notice that all we needed in the proof of Lemma 2.2 is that upon arrival of a request $r_k = i$ when there are k free servers S_k we match $r_k = i$ to a any free server s with probability $x^{S_k}_{i,s}/p_i$, and so we use edge (i, s) with probability precisely $x^{S_k}_{i,s}$. This implies that each free server $s \in S_k$ is matched with probability precisely $\frac{1}{k}$ and that the expected weight of the edge matched is precisely $\sum_{i\in S, j\in S_k} w_{i,j} \cdot x^{S_k}_{i,j}$.

Next, we note that $\mathbb{E}[OPT]$ can be upper bounded in terms of $M(S)$.

Claim D.2. $\mathbb{E}[\text{OPT}] \leq n \cdot M(S)$.

The proof is exactly the same as Equation (5). See also [7, Lemma 1].

Now we can prove that the maximization variant of FAIR-BIAS is 1/2 competitive for the max weight perfect matching problem in the i.i.d model.

Theorem D.3. *The max-weight variant of* FAIR-BIAS *is 1/2 competitive.*

Proof. Letting $x^{S_k} \in \arg\max M(S_k)$ for every S_k, we have the following bound

$$\begin{aligned} \mathbb{E}_{S_k\sim\mathcal{U}_k}[M(S_k)] &= \sum_{S_k} \frac{1}{\binom{n}{k}} \sum_{i\in S_k, j\in R} w_{i,j} \cdot x^{S_k}_{i,j} && \text{def. of } x^{S_k} \\ &\geq \sum_{S_k} \frac{1}{\binom{n}{k}} \sum_{i\in S_k, j\in R} w_{i,j} \cdot x^{S}_{i,j} && \text{def. of } x^{S_k} \text{ and } M(S_k) \\ &= \sum_{i\in S} \Pr_{S_k\sim\mathcal{U}_k}[i \in S_k] \cdot \sum_{j\in R} w_{i,j} \cdot x^{S}_{i,j} \end{aligned}$$

$$
\begin{aligned}
&= \frac{k}{n} \cdot M(S) && \text{def. of } M(S) \\
&\geq \frac{k}{n^2} \cdot \mathbb{E}[\text{OPT}]. && \textit{Claim D.2}
\end{aligned}
$$

Summing these up, by the structure lemma (Claim D.1) we have

$$\mathbb{E}[\text{ALG}] = \sum_{i=1}^{n} \mathbb{E}_{S_k \sim \mathcal{U}_k}[M(S_k)] \geq \sum_{k=1}^{n} \frac{k}{n^2} \cdot \mathbb{E}[\text{OPT}] \geq \frac{1}{2} \cdot \mathbb{E}[\text{OPT}]. \qquad \square$$

E Need for Metricity (or other assumptions)

Here we outline simple examples showing that even under i.i.d arrivals, online minimum cost perfect matching does not admit even a polynomially-bounded competitive ratio. For unknown i.i.d, and therefore for random order and adversarial arrivals, even *one edge* violating the triangle inequality is enough to rule out sub-exponential competitive ratio. For random order and adversarial arrivals one such edge is enough to cause the competitive ratio to be unbounded.

Lemma E.1. *The competitive ratio of any online min cost perfect matching algorithm under known i.i.d arrivals is at least $\Omega(2^{n/2}/n^3)$. This is true even under a uniform distribution and if the costs of all but 2 request types obey triangle inequality.*

Proof. Let n be even and let $[n]$ be the set of servers. Consider the following set of request types (each with probability $1/n$ of being drawn at each arrival): the first $n-2$ request types have cost 1 to be served by all servers. So far the instance corresponds to the uniform metric on $2n-2$ points. Now, second to last request type has cost 1 to be served by serves in $[n/2]$, and cost $2^{n/2}$ to be served by serves in $[n/2+1, n]$, and the last request type has the exact opposite costs. When fewer than $n/2$ of the last two types arrive, OPT is exactly n, whereas in the opposite case, which happens with probability at most $2^{-n/2}$ by standard Chernoff Bounds, OPT is at most $n \cdot \exp^{n/4}$, and so $E[OPT] \leq 2n$. On the other hand, with probability $\Omega(1/n^2)$, exactly one request from the last two request types arrives, and this is the last of all arrivals. In this case, as the algorithm must match $n-1$ servers before this arrival, with constant probability the sole remaining unmatched server has cost $2^{n/2}$ to match to this last request. Therefore, we have $E[ALG] = \Omega(2^{n/2}/n^2)$. □

A similar argument implies that for the unknown i.i.d arrival model, even a single edge which violates triangle inequality is enough to rule out sub-exponential competitive ratio .

Lemma E.2. *The competitive ratio of any online min-cost perfect matching algorithm under unknown i.i.d arrivals is at least $n^{n-2}/2$. This is true even under a uniform distribution and if the costs of all edges but one satisfy the triangle inequality.*

Proof (Sketch). The distribution is similar to that of Lemma E.1. We have $[n]$ denote the servers and have $n-1$ request types with service cost 1 for each server. The final type has service cost 1 for all servers except for one (unknown) server for which the service cost is n^n. Each request is drawn uniformly from this distribution. Unless n copies of the last request type arrive (an even which happens with probability $1/n^n$), the cost of the optimal matching is $OPT = n$, and otherwise it is $n-1+n^n$, and so $E[OPT] \leq 2n$. On the other hand, with probability $\Omega(1/n)$, the special request type has exactly one arrival, and this is at the last time step, and so with probability $1/n$ this request's "costly" serve is the sole unmatched server, implying $E[ALG] = \Omega(n^{n-2})$. □

Finally, the same argument can show that the same input as in Lemma E.2, with the sole costly edge being arbitrarily high, rules out any bounded competitive ratio, as having exactly one request of each type yields and input with $OPT = n$ but with ALG's matching cost being unboundedly bad with probability $\Omega(1/n)$.

Corollary E.3. *The competitive ratio of any online min-cost perfect matching algorithm under random arrival order is* unbounded. *This is true even if the costs of all edges but one edge satisfy the triangle inequality.*

References

[1] Aggarwal, G., Goel, G., Karande, C., and Mehta, A. 2011. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 1253–1264.

[2] Antoniadis, A., Barcelo, N., Nugent, M., Pruhs, K., and Scquizzato, M. 2014. A $o(n)$-competitive deterministic algorithm for online matching on a line. In *Proceedings of the 12th Workshop on Approximation and Online Algorithms (WAOA)*. 11–22.

[3] Bahmani, B. and Kapralov, M. 2010. Improved bounds for online stochastic matching. In *Proceedings of the 18th Annual European Symposium on Algorithms (ESA)*. 170–181.

[4] Bansal, N., Buchbinder, N., Gupta, A., and Naor, J. S. 2007. An $O(\log^2 k)$-competitive algorithm for metric bipartite matching. In *Proceedings of the 15th Annual European Symposium on Algorithms (ESA)*. 522–533.

[5] Berend, D. and Kontorovich, A. 2013. A sharp estimate of the binomial mean absolute deviation with applications. *Statistics & Probability Letters 83,* 4, 1254–1259.

[6] Brubach, B., Sankararaman, K. A., Srinivasan, A., and Xu, P. 2016. New algorithms, better bounds, and a novel model for online stochastic matching. In *Proceedings of the 24th Annual European Symposium on Algorithms (ESA)*. 24:1–24:16.

[7] Chang, M., Hochbaum, D. S., Spaen, Q., and Velednitsky, M. 2018. DISPATCH: an optimally-competitive algorithm for maximum online perfect bipartite matching with iid arrivals. In *Proceedings of the 16th Workshop on Approximation and Online Algorithms (WAOA)*. 149–164.

[8] Cohen, I. R. and Wajc, D. 2018. Randomized online matching in regular graphs. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 960–979.

[9] Dehghani, S., Ehsani, S., Hajiaghayi, M., Liaghat, V., and Seddighin, S. 2017. Stochastic k-server: How should uber work? In *Proceedings of the 44th International Colloquium on Automata, Languages and Programming (ICALP)*. 126:1–126:14.

[10] Devanur, N. R., Sivan, B., and Azar, Y. 2012. Asymptotically optimal algorithm for stochastic adwords. In *Proceedings of the 13th ACM Conference on Electronic Commerce (EC)*. 388–404.

[11] Esfandiari, H., Korula, N., and Mirrokni, V. S. 2015. Online allocation with traffic spikes: Mixing adversarial and stochastic models. In *Proceedings of the 16th ACM Conference on Economics and Computation (EC)*. 169–186.

[12] Fakcharoenphol, J., Rao, S., and Talwar, K. 2004. A tight bound on approximating arbitrary metrics by tree metrics. *Journal of Computer and System Sciences 69,* 3, 485–497.

[13] Feldman, J., Mehta, A., Mirrokni, V., and Muthukrishnan, S. 2009. Online stochastic matching: Beating $1 - 1/e$. In *Proceedings of the 50th Symposium on Foundations of Computer Science (FOCS)*. 117–126.

[14] Fuchs, B., Hochstättler, W., and Kern, W. 2005. Online matching on a line. *Theoretical Computer Science (TCS) 332,* 1-3, 251–264.

[15] Gamlath, B., Kapralov, M., Maggiori, A., Svensson, O., and Wajc, D. 2019. Online matching with general arrivals. *arXiv preprint arXiv:1904.08255*.

[16] Garg, N., Gupta, A., Leonardi, S., and Sankowski, P. 2008. Stochastic analyses for online combinatorial optimization problems. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 942–951.

[17] Goel, G. and Mehta, A. 2008. Online budgeted matching in random input models with applications to adwords. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 982–991.

[18] Grandoni, F., Gupta, A., Leonardi, S., Miettinen, P., Sankowski, P., and Singh, M. 2013. Set covering with our eyes closed. *SIAM Journal on Computing (SICOMP) 42,* 3, 808–830.

[19] Gupta, A. and Lewi, K. 2012. The online metric matching problem for doubling metrics. In *Proceedings of the 39th International Colloquium on Automata, Languages and Programming (ICALP)*. 424–435.

[20] Haeupler, B., Mirrokni, V. S., and Zadimoghaddam, M. 2011. Online stochastic weighted matching: Improved approximation algorithms. In *Proceedings of the 7th Conference on Web and Internet Economics (WINE)*. 170–181.

[21] Huang, Z., Kang, N., Tang, Z. G., Wu, X., Zhang, Y., and Zhu, X. 2018a. How to match when all vertices arrive online. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC)*. 17–29.

[22] Huang, Z., Peng, B., Tang, Z. G., Tao, R., Wu, X., and Zhang, Y. 2019. Tight competitive ratios of classic matching algorithms in the fully online model. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2875–2886.

[23] Huang, Z., Tang, Z. G., Wu, X., and Zhang, Y. 2018b. Online vertex-weighted bipartite matching: Beating 1-1/e with random arrivals. In *Proceedings of the 45th International Colloquium on Automata, Languages and Programming (ICALP)*. 1070–1081.

[24] Kalyanasundaram, B. and Pruhs, K. 1993. Online weighted matching. *Journal of Algorithms 14,* 3, 478–488.

[25] Karande, C., Mehta, A., and Tripathi, P. 2011. Online bipartite matching with unknown distributions. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC)*. 587–596.

[26] Karp, R. M., Vazirani, U. V., and Vazirani, V. V. 1990. An optimal algorithm for on-line bipartite matching. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing (STOC)*. 352–358.

[27] Khuller, S., Mitchell, S. G., and Vazirani, V. V. 1994. On-line algorithms for weighted bipartite matching and stable marriages. *Theoretical Computer Science (TCS) 127,* 2, 255–267.

[28] Mahdian, M., Nazerzadeh, H., and Saberi, A. 2007. Allocating online advertisement space with unreliable estimates. In *Proceedings of the 8th ACM Conference on Electronic Commerce (EC)*. 288–294.

[29] Mahdian, M. and Yan, Q. 2011. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC)*. 597–606.

[30] Mehta, A. 2013. Online matching and ad allocation. *Foundations and Trends® in Theoretical Computer Science 8,* 4, 265–368.

[31] Mehta, A., Saberi, A., Vazirani, U., and Vazirani, V. 2007. Adwords and generalized online matching. *Journal of the ACM (JACM) 54,* 5, 22.

[32] Meyerson, A., Nanavati, A., and Poplawski, L. 2006. Randomized online algorithms for minimum metric bipartite matching. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 954–959.

[33] Mirrokni, V. S., Gharan, S. O., and Zadimoghaddam, M. 2012. Simultaneous approximations for adversarial and stochastic online budgeted allocation. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 1690–1701.

[34] Mitzenmacher, M. and Upfal, E. 2005. *Probability and computing: Randomized algorithms and probabilistic analysis.* Cambridge university press.

[35] Naor, J. S. and Wajc, D. 2018. Near-optimum online ad allocation for targeted advertising. *ACM Transactions on Economics and Computation (TEAC) 6,* 3-4, 16.

[36] Nayyar, K. and Raghvendra, S. 2017. An input sensitive online algorithm for the metric bipartite matching problem. In *Proceedings of the 58th Symposium on Foundations of Computer Science (FOCS)*. 505–515.

[37] Raghvendra, S. 2016. A robust and optimal online algorithm for minimum metric bipartite matching. In *Proceedings of the 19th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*. Vol. 60.

[38] Raghvendra, S. 2018. Optimal analysis of an online algorithm for the bipartite matching problem on a line. In *Proceedings of the 34th Symposium on Computational geometry (SoCG)*. 67:1–67:14.