

Online Stochastic Matching: Online Actions Based on Offline Statistics

Vahideh H. Manshadi^{*} Shayan Oveis Gharan[†] Amin Saberi[†]

August 3, 2011

Abstract

We consider the online stochastic matching problem proposed by Feldman et al. [4] as a model of display ad allocation. We are given a bipartite graph; one side of the graph corresponds to a fixed set of bins and the other side represents the set of possible ball types. At each time step, a ball is sampled independently from the given distribution and it needs to be matched upon its arrival to an empty bin. The goal is to maximize the number of allocations.

We present an online algorithm for this problem with a competitive ratio of 0.702. Before our result, algorithms with a competitive ratio better than $1 - 1/e$ were known under the assumption that the expected number of arriving balls of each type is integral. A key idea of the algorithm is to collect statistics about the decisions of the optimum offline solution using Monte Carlo sampling and use those statistics to guide the decisions of the online algorithm. We also show that our algorithm achieves a competitive ratio of 0.705 when the rates are integral.

On the hardness side, we prove that no online algorithm can have a competitive ratio better than 0.823 under the known distribution model (and henceforth under the permutation model). This improves upon the $\frac{5}{6}$ hardness result proved by Goel and Mehta [7] for the permutation model.

1 Introduction

We study a natural variation of bipartite matching problem motivated in the context of online advertising: suppose we are given a bipartite graph $G(Y, Z, E)$ where Y is the set of stochastic nodes (or ball types) and Z is the set of non-stochastic nodes (or bins). At times $t = 1, 2, \dots, b$, a ball of type $y \in Y$ is chosen independently at random from a given distribution. The algorithm can assign the ball to at most one of the empty bins that are adjacent to it. Further, each bin can be matched to at most one ball. The goal is to maximize the expected number of non-empty bins at time b . We refer to this model as the *known distribution model*.

When the balls are chosen by an adversary instead of a random process, Karp, Vazirani, and Vazirani [10] gave a simple and elegant randomized algorithm that achieves a competitive ratio of $1 - 1/e$. We present the first algorithm for this problem that improves the $1 - 1/e$ competitive ratio for the stochastic version in its general form. Previously, Feldman et al. [4] (and later [1]) used a very interesting combinatorial algorithm to show that this is possible when the arrival rate of every ball, that is the expected number of times it appears, is integral (this is also known as the *i.i.d. model*). This assumption, even though not very restrictive for the display ad allocation,

^{*}Department of Electrical Engineering, Stanford University, Stanford, CA 94305. Email: vahidehh@stanford.edu.

[†]Department of Management Science and Engineering, Stanford University, Stanford, CA 94305. Email: {shayan, saberi}@stanford.edu.

is somewhat unnatural. For example, when the distribution is uniform, it requires $b/|Y|$ to be an integer.

One of the key ideas in designing our algorithm is to approximately compute the expected matching used by the optimum offline algorithm and use it to guide the decisions of the online algorithm. In particular, using Monte Carlo sampling, one can compute $f_{(y,z)}$, the probability that the optimum offline algorithm allocates a ball of type y to a bin of type z , for every y and z . Without loss of generality, we can assume f is a fractional matching.

Our first algorithm writes f as a distribution over integral matchings and samples two matchings M_1 and M_2 from it. Then, in the online phase, it will use these two matchings for allocating the arriving balls to the bins (see Section 3). The analysis of our algorithm is much shorter and simpler than both [4, 1]. All these algorithms are non-adaptive, in the sense that they decide the allocation of all the balls regardless of the allocation of the bins before they arrive. We present a simple example to show that no non-adaptive algorithm can achieve a competitive ratio better than $1 - 1/e$ when the arrival rates are non-integral (see Proposition 5.1).

The main result of the paper is an *adaptive algorithm* that obtains a competitive ratio of 0.702 for arbitrary rates, and 0.705 for the i.i.d. model (see Section 4). Unlike the non-adaptive algorithms, our adaptive algorithm decides the allocation of each arriving ball based on the current allocation of the bins. In particular, when a ball arrives the algorithm samples two neighbor bins from a joint distribution and tries to match it to the first bin; if the bin is already matched the algorithm tries the second bin. To the best of our knowledge, this is the first algorithm that beats the $1 - 1/e$ ratio in the general form. The adaptivity of the algorithm imposes a lot of dependencies in the distribution of full bins and because of that our analysis is somewhat intricate.

On the hardness side, we present an example that gives an upper bound of 0.823 on the competitive ratio of any deterministic or randomized online algorithm in the known distribution model (see Proposition 5.3). For analyzing this example, we use the expected size of a maximum matching of a random bipartite graph recently computed by [3, 6, 5] in the context of random SAT and cuckoo hashing.

1.1 Related Work

Bipartite matching problems are central in algorithms and combinatorial optimization and arise naturally in several applications such as resource allocation, scheduling, and online advertising.

The online matching problem was first studied by Karp, Vazirani, and Vazirani [10] in the adversarial model where the graph is unknown; when a ball arrives it reveals its incident edges. They proved that a simple randomized on-line algorithm achieves $(1 - 1/e)$ and this factor is the best possible performance.

More recently, Feldman et al. [4] studied the problem under stochastic assumptions. They assumed that the graph is known but the sequence of arrivals are *i.i.d.* samples from a given distribution. Further, they assumed that sampling rates are integral and developed an online algorithm that beats $(1 - 1/e)$. They also showed that there is no $1 - o(1)$ -approximation algorithm for this setting. Recently, Bahmani and Kapralov [1] improved the upper and lower bounds of Feldman et al. to 0.902 and 0.699 respectively in the same setting. Also, they showed that for d -regular graphs, a simple randomized algorithm achieves a competitive ratio of $1 - O(1/\sqrt{d})$ [1].

Goel and Mehta [7] considered a different stochastic model: they assumed the graph is unknown but the sequence of arrivals is a random permutation. This is known as the *random permutation model*, and it is a generalization of the known distribution model. They showed that a greedy

algorithm achieves $(1 - 1/e)$ factor. Further, they showed that no online algorithm can achieve competitive ratio better than $\frac{5}{6}$. Since the known distribution model is a special case of the random permutation model, our hardness result improves their upper-bound to 0.823. Since the first appearance of this paper, Karande et al. [9], and Mahdian and Yan [11] independently improve the $(1 - 1/e)$ competitive ratio in the random permutation model to 0.653 and 0.696 respectively.

A close line of work to the online matching is the online b -matching and the AdWords problem [12, 2]. Mehta et al. [12] developed a $(1 - 1/e)$ online algorithm in the adversarial case. Recently, Devanur and Hayes [2] improved the competitive ratio to $(1 - \epsilon)$ in the stochastic case where the sequence of arrivals is a random permutation or it consists of *i.i.d.* samples.

2 Problem Definition

Let $G(Y, Z, E)$ be a bipartite graph where Y is the set of stochastic nodes (or ball types) and Z is the set of non-stochastic nodes (or bins). There is a rate r_y associated to every type of ball $y \in Y$. The online stochastic matching problem is as follows: at times $t = 1, 2, \dots, b$, a ball of type $y \in Y$ is chosen independently and with probability proportional to r_y . The algorithm can assign this ball to at most one of the empty bins that are adjacent to it; each bin can be matched to at most one ball. The goal of the algorithm is to maximize the expected number of non-empty bins at time b .

Without loss of generality, we assume that $\sum_{y \in Y} r_y = b$, thus the expected number of balls of type y in the sequence is r_y . Also, we assume that $r_y \leq 1$; if a node has a rate greater than 1, we can easily split it into a set of identical nodes with rates at most 1.

We will study two classes of algorithms: non-adaptive and adaptive. A non-adaptive algorithm is equivalent to an ordering of the neighbors $N(y)$ of every node $y \in Y$. If $z_1, z_2, \dots, z_{|N(y)|}$ is such an ordering for y , then the k -th time a ball of type y arrives, the algorithm will allocate it to bin z_k if it is empty. If $k > |N(y)|$ or z_k is full then the ball will not be allocated. On the other hand, adaptive algorithms can choose the assignment of every ball when it arrives.

We will compare our algorithms to the optimum offline solution. Given the sequence of arrived balls $\omega = (y_1, y_2, \dots, y_b)$, one can compute the optimum allocation, $\text{OPT}(\omega)$, in polynomial time by solving a maximum matching problem. Fix a particular maximum matching algorithm and let $F(\omega) : E \rightarrow \{0, 1\}$ be the vector indicating which edges are used in the optimum allocation given ω . Clearly, $\text{OPT}(\omega) = 1^T F(\omega)$ and the competitive ratio of an online algorithm ALG is defined as $\frac{\mathbb{E}[\text{ALG}]}{\mathbb{E}[\text{OPT}]}$. In our case, both ALG and OPT are concentrated around their expected values, therefore the above competitive ratio is fairly robust (see Feldman et al. [4] for a more detailed discussion).

Our algorithms will crucially use the optimum offline solution for making decisions. In particular, define

$$f = \sum_{\omega} F(\omega) \mathbb{P}(\omega), \quad (1)$$

where $\mathbb{P}(\omega)$ is the probability of the sequence $\omega = (y_1, y_2, \dots, y_b)$. By definition, f is a convex combination of matchings and therefore it is in the convex hull of the matchings of G . We will refer to f as the *fractional matching* defined by OPT. For each edge $e = (y, z) \in E$, f_e is the probability that a ball of type y is allocated to bin z by the optimum offline algorithm. Similarly we define the fractional degree of a node to be $f_v = \sum_{e \sim v} f_e$ for $v \in Y \cup Z$.

Proposition 2.1 *The vector f is a fractional matching in G . i.e.*

$$f_y \leq r_y \leq 1, \quad y \in Y, \quad \text{and} \quad f_z \leq 1, \quad z \in Z. \quad (2)$$

Moreover, for $e = (y, z)$, we have $f_e \leq 1 - e^{-r_y} + o(1/b)$.

Proof: Given ω , let $N_y(\omega)$ be the number of balls of type y in ω . Clearly $\sum_{e \sim y} F_e(\omega) \leq N_y(\omega)$. Taking expectations from both sides results in the first inequality in (2). Similarly, the second inequality in (2) can be proved by noting that in any instance of the problem, z can be matched to at most one ball. Finally, for $e = (y, z)$, we have

$$f_e \leq \mathbb{P}(N_y(\omega) \geq 1) = 1 - (1 - \frac{r_y}{b})^b \leq 1 - e^{-r_y} + o(1/b).$$

□

Throughout the paper, we will assume that b is sufficiently large so that $o(1/b)$ is negligible. We will need to compute f_e for every edge e . Obviously, f_e 's can be computed by enumeration in time $O(|Y|^b)$. It is also easy to see that $\mathbb{E}[\text{OPT}]$ and $f(e)$ for all $e \in E$, can be approximated with great accuracy using Monte Carlo method. OPT is an integral random variable which is in interval $[0, b]$, hence its variance is upper-bounded by b^2 . Therefore, $\mathbb{E}[\text{OPT}]$ can be estimated with error of $o(1/b)$, by averaging over $O(b^3)$ independent samples of the process. A similar argument shows that with $O(|E|^2 b^4)$ samples of ω in equation (1), with high probability, one can compute the vector f with accuracy within $o(1/b|E|)$. In the rest of the paper, for simplicity of notation, we will assume that we have estimated f accurately and ignore $o(\cdot)$ terms.

Since f is a fractional matching, standard algorithmic versions of Caratheodory's theorem (see e.g. [8, Theorem 6.5.11]) say that, in polynomial time, we can decompose a feasible solution in the bipartite matching polytope into a convex combination of polynomially many bipartite matchings. More specifically, we obtain the following:

Corollary 2.2 *It is possible to efficiently and explicitly construct (and sample from) a distribution μ on the set of matchings in G such that*

$$\sum_{M, e \in M} \mu(M) = f_e, \quad \forall e \in E$$

3 A Non-adaptive algorithm

In this section, we will analyze a simple non-adaptive algorithm for the special case where all rates are one, i.e., $r_y = 1, \forall y \in Y$. This is the setting studied in Feldman et al. [4]. Our algorithm and its analysis is simpler and more intuitive than [4]. It also gives a slightly better competitive ratio.

Our non-adaptive algorithm has some similarities with the online algorithm that Feldman et al. propose [4]. Both algorithms start by computing two matchings M_1 and M_2 offline; we use the first matching, only for the first arrived ball of each type and the second one only for the second arrivals. In particular, when the first ball of type y arrives it will be allocated to the bin matched to y in M_1 , and when the second ball arrives, we will allocate it via M_2 . If the corresponding bins are already full, the balls will be dropped. Note that the probability that there are more than two balls of each type y in the sequence of arrivals is very small.

On the other hand, we use a different method from [4] to construct these matchings. Feldman et al. find M_1 and M_2 by decomposing the solution of a maximum 2-flow of G into two disjoint matchings (since all the rates are one, the expected graph is simply G). However, we will sample our matchings from the distribution μ defined by the optimum solution f .

Algorithm 1 The Online Non-adaptive Algorithm

Offline Phase:

- 1: Compute the fractional matching f , and the distribution μ using Corollary 2.2.
- 2: Sample two matchings M_1 and M_2 from μ independently; set M_1 (M_2) to be the first (second) priority matching.

Online Phase:

- 3: When the first ball of type y arrives, allocate it through the first priority matching, M_1 .
 - 4: Similarly, when a ball of type y arrives for the second time, allocate it through the second priority matching, M_2 .
-

The outline of the algorithm is presented in Algorithm 1. In the rest of this section, we analyze Algorithm 1, and show that its approximation ratio is 0.684. Let X_z be the random variable indicating the event that bin z is matched with a ball during the run of the algorithm. We analyze the competitive ratio of the algorithm by comparing $\mathbb{E}[X_z]$ with f_z :

$$\frac{\mathbb{E}[\text{ALG}]}{\mathbb{E}[\text{OPT}]} = \frac{\sum_{z \in Z} \mathbb{E}[X_z]}{\sum_{z \in Z} f_z} \geq \min_{z \in Z} \frac{\mathbb{E}[X_z]}{f_z}$$

Consider any $z \in Z$, and with a slight abuse of notation let $M_1(z)$ denote the stochastic node matched to it in M_1 . More precisely, if $(y, z) \in M_1$, define $M_1(z) = \{y\}$, and if z is not saturated in M_1 , define $M_1(z) = \emptyset$; similarly define $M_2(z)$. Note that z is saturated by M_1 (or M_2) with probability f_z , but if $M_1(z) = M_2(z)$, bin z will only be used for the first arrived ball and effectively it is not saturated by M_2 . Given M_1 and M_2 , $\mathbb{E}[X_z | M_1, M_2]$ can be computed similar to [4, section 4.2.2] by considering the following cases:

$$\mathbb{E}[X_z | M_1, M_2] = \begin{cases} 0 & \text{if } M_1(z) = M_2(z) = \emptyset \\ 1 - 1/e & \text{if } M_1(z) \neq \emptyset, \{M_1(z) = M_2(z)\} \\ 1 - 1/e & \text{if } M_1(z) \neq \emptyset, M_2(z) = \emptyset \\ 1 - 2/e & \text{if } M_1(z) = \emptyset, M_2(z) \neq \emptyset \\ 1 - 2/e^2 & \text{if } M_1(z) \neq \emptyset, M_2(z) \neq \emptyset, M_1(z) \neq M_2(z) \end{cases} \quad (3)$$

By substituting (3) into $\mathbb{E}[X_z]$ we get:

$$\begin{aligned} \mathbb{E}[X_z] &= (1 - 1/e) \sum_{e \sim z} f_e (1 - f_z + f_e) + (1 - 2/e) \sum_{e \sim z} f_e (1 - f_z) + (1 - 2/e^2) \sum_{e, e' \sim z, e \neq e'} f_e f_{e'} \\ &= f_z (2 - 3/e) - f_z^2 (1 + 2/e^2 - 3/e) - (1/e - 2/e^2) \sum_{e \sim z} f_e^2 \end{aligned}$$

The last equality can be derived by algebraic manipulation and noting that $\sum_{e \sim z} f_e = f_z$. It remains to prove a lower bound on the value of the last equation:

Lemma 3.1 *In any graph $G = (Y, Z, E)$, if f is the corresponding vector of the optimum solution, we have*

$$\frac{\mathbb{E}[X_z]}{f_z} = (2 - 3/e) - (1 + 2/e^2 - 3/e) f_z - (1/e - 2/e^2) \frac{\sum_{e \sim z} f_e^2}{f_z} \geq 0.684 \quad (4)$$

Proof: The proof of this lemma is mainly algebraic. Let us first fix f_z and find the minimum of the LHS in terms of f_z . For any f_z , the LHS is minimized when $\sum_{e \sim z} f_e^2$ is maximized. Note that $\sum_{e \sim z} f_e = f_z$, and thus to maximize the $\sum_{e \sim z} f_e^2$, we need to consider the most “unbalanced” edge probabilities that are consistent with the properties of fractional matching f . By proposition 2.1, $f_e \leq 1 - e^{-1}$ for each $e \sim z$, thus for $f_z \leq 1 - e^{-1}$, the term $\sum_{e \sim z} f_e^2$ is maximized when we have only one edge with nonzero probability. Similarly we can show that the summation of the probabilities of any 2 edges incident to z is at most $1 - e^{-2}$, thus if $1 - e^{-1} \leq f_z \leq 1 - e^{-2}$, the term $\sum_{e \sim z} f_e^2$ is maximized when we have two edges with nonzero probability; one edge with probability $1 - e^{-1}$ and one with probability $f_z - (1 - e^{-1})$. Similarly we can proceed to compute the maximum of $\sum_{e \sim z} f_e^2$ in terms of f_z for all $0 \leq f_z \leq 1$.

The only remaining task is to find the value f_z that minimizes the LHS of (4). Intuitively, the LHS is minimized when $f_z = 1$. In particular, if $f_z < 1$, we may add a dummy node y to Y , and connect it to z by an edge $e = (y, z)$ with very small probability, i.e. $f_e = \epsilon$. It is easy to see that this can only decrease the LHS. Also, one can numerically confirm that the LHS of (4) attains its minimum at $f_z = 1$ with value 0.684. □

Theorem 3.2 *Assuming all the rates are 1, the solution of Algorithm 1 is within 0.684 of the optimum offline solution.*

4 The Adaptive Algorithm

In the analysis of the non-adaptive algorithm presented in the previous section, we assumed that the arrival rates of all stochastic nodes are integral and in particular, they are at least one. This is a crucial assumption. If the rates r_y 's are not bounded from below, the probability of receiving a second ball of the same type can become arbitrary low and the competitive ratio of the algorithm can get very close to $1 - 1/e$. This is the case for all non-adaptive algorithms: In Proposition 5.1 we show that no non-adaptive (even randomized) algorithm can achieve a competitive ratio better than $1 - 1/e$ when the sampling rates are not necessarily integral.

In this section, we will analyze a simple *adaptive* algorithm that will have a better performance for arbitrary rates. The algorithm is very simple: when a ball of type y arrives, it samples two neighboring bins z_1 and z_2 from a joint distribution. If z_1 is empty then y is matched to z_1 . Otherwise, the algorithm will try z_2 and match y to it if it is empty.

The joint distribution from which z_1 and z_2 are chosen, is determined in advance for every ball type y and it has the following properties: (i) The probability that z_1 is equal to z is equal to $f_{(y,z)}$. The same is true for z_2 . Recall that rates are normalized such that $\sum_{y \in Y} r(y) = b$ and thus f is a fractional matching. (ii) The joint distribution is such that the probability of $z_1 = z_2$ is minimized. Note that such a joint probability maximizes the possibility that a ball tries a second different bin in case the first bin that it tries is full. In what follows, we will present one joint distribution with these properties.

Suppose $(y, z_1), \dots, (y, z_k)$ are the edges incident to y , and without loss of generality assume that $f_{(y,z_1)} \geq \dots \geq f_{(y,z_k)}$. Also define a dummy edge (y, z_{k+1}) that is connected to a dummy non-stochastic node z_{k+1} , with $f_{(y,z_{k+1})} = r_y - f_y$. Note that $f_{(y,z_{k+1})}$ is the probability that OPT drops a ball of type y . We will construct two different partitions of the interval $I_y = [0, r_y]$. Specifically, partitions \mathcal{I}_y and \mathcal{J}_y are defined as follows:

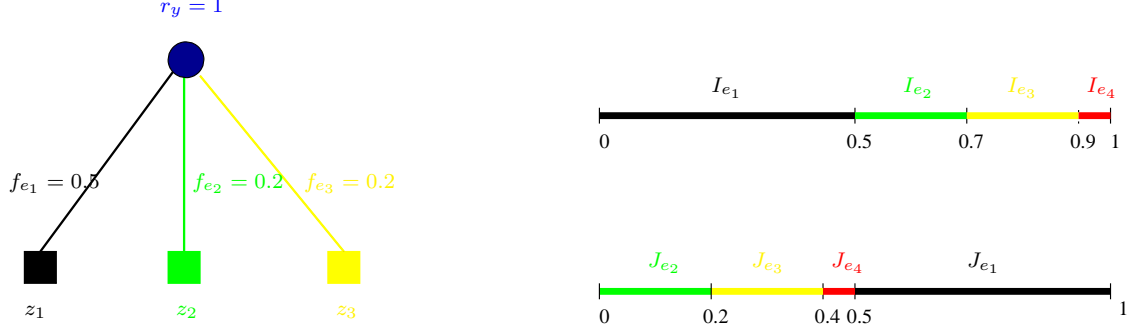


Figure 1: Illustration of partitions \mathcal{I}_y and \mathcal{J}_y for node y with edges e_1 , e_2 , and e_3

- Partition \mathcal{I}_y : let $I_{(y,z_1)} = [0, f_{(y,z_1)}]$; similarly let $I_{(y,z_l)} = [\sum_{j=1}^{l-1} f_{(y,z_j)}, \sum_{j=1}^l f_{(y,z_j)}]$, $2 \leq l \leq k+1$.
- Partition \mathcal{J}_y : let $J_{(y,z_1)} = [r_y - f_{(y,z_1)}, r_y]$, $J_{(y,z_2)} = [0, f_{(y,z_2)}]$, and similarly $J_{(y,z_l)} = [\sum_{j=2}^{l-1} f_{(y,z_j)}, \sum_{j=2}^l f_{(y,z_j)}]$, $3 \leq l \leq k+1$.

Note that the second partition is obtained by shifting the subintervals of \mathcal{I}_y to the left by $f_{(y,z_1)}$. Figure 1 illustrates the partitions through a simple example. Having \mathcal{I}_y and \mathcal{J}_y , the distribution is defined as follows: choose a number x uniformly at random from $[0, r_y]$, define $z_{1,y}(x)$ to be z if $x \in I_{(y,z)}$; similarly define $z_{2,y}(x)$ to be z' if $x \in J_{(y,z')}$. It is easy to see that this joint distribution has property (i). Also, note that the second partition \mathcal{J}_y has the minimum possible overlap with the first one which implies that the resulting joint probability has property (ii), i.e., for each stochastic node y , the probability that $z_{1,y}(\cdot) = z_{2,y}(\cdot)$ is minimized. Further, if all $f_{(y,z)}$'s are less than $\frac{1}{2}r_y$, the probability of $z_{1,y}(\cdot) = z_{2,y}(\cdot)$ is equal to zero.

Observation 4.1 For stochastic node y , suppose (y, z^*) is the edge with the maximum probability, i.e. $f_{(y,z^*)} \geq f_{(y,z)}$, $\forall z \sim y$. If $f_{(y,z^*)} < \frac{1}{2}r_y$ then $z_{1,y}(x) \neq z_{2,y}(x)$, for all $x \in [0, r_y]$. Otherwise, $z_{1,y}(x) \neq z_{2,y}(x)$ only for $x \in [r_y - f_{(y,z^*)}, f_{(y,z^*)}]$.

The outline of the algorithm is presented in Algorithm 2.

Algorithm 2 Online Adaptive Algorithm

Offline Phase:

- 1: Compute the fractional matching f .
- 2: For each $y \in Y$ and $x \in [0, r_y]$, construct the functions $z_{1,y}(\cdot)$ and $z_{2,y}(\cdot)$ by defining the corresponding partitions \mathcal{I}_y and \mathcal{J}_y .

Online Phase:

- 3: If a ball of type $y \in Y$ arrives, choose a number x uniformly at random from interval $[0, r_y]$.
 - 4: Match the ball with $z_{1,y}(x)$;
 - 5: If $z_{1,y}(x)$ is full, match the ball with $z_{2,y}(x)$;
-

Theorem 4.2 For any graph G and arbitrary set of rates $\{r_y, y \in Y\}$, the competitive ratio of Algorithm 2 is at least 0.702.

Unlike Algorithm 1, the analysis of Algorithm 2 is fairly intricate, mainly because the adaptivity of the algorithm introduces new dependencies. We will present the proof in a few steps to build an intuition before getting to the actual calculations.

Proof: Consider a non-stochastic node $z \in Z$. Bin z can be matched as a first priority bin or as a second priority bin. Note that a bin will be matched once it is tried as a first or second priority. We define the event $\mathcal{A}_z(t)$ to be the event that bin z was tried as a first priority bin by time t , i.e., at any time $1, 2, \dots, t$. Also, define $\mathcal{B}_z(t)$ to be the event that bin z was tried as a second priority bin at time t . Using the notation defined in the previous section:

$$\begin{aligned} \mathbb{E}[X_z] &= \mathbb{P}\left(\mathcal{A}_z(b) \vee \bigcup_{t=1}^b \mathcal{B}_z(t)\right) = \mathbb{P}(\mathcal{A}_z(b)) + \mathbb{P}\left(\bigcup_{t=1}^b \mathcal{B}_z(t) \wedge \overline{\mathcal{A}_z(b)}\right) \\ &= \mathbb{P}(\mathcal{A}_z(b)) + \mathbb{P}\left(\bigcup_{t=1}^b \mathcal{B}_z(t) \mid \overline{\mathcal{A}_z(b)}\right) \mathbb{P}(\overline{\mathcal{A}_z(b)}) \end{aligned} \quad (5)$$

We need to compute $\mathbb{P}(\mathcal{A}_z(b))$. Instead we compute $\mathbb{P}(\mathcal{A}_z(t))$ for $1 \leq t \leq b$; at each time step, the probability that a ball tries z as a first priority bin is equal to the probability that a ball of type y arrives, where y is connected to z through edge (y, z) , and we choose a point in the interval $I_{(y,z)}$. This probability is $\frac{\sum_{y \sim z} f_{(y,z)}}{\sum_{y \in Y} r_y} = \frac{f_z}{b}$, and we have:

$$\mathbb{P}(\mathcal{A}_z(t)) = 1 - \left(1 - \frac{f_z}{b}\right)^t = 1 - e^{-\frac{t f_z}{b}} + o(1/b). \quad (6)$$

Thus $\mathbb{P}(\mathcal{A}_z(b)) = 1 - e^{-f_z}$. The more difficult part of the analysis is to lower-bound $\mathbb{P}\left(\bigcup_{t=1}^b \mathcal{B}_z(t) \mid \overline{\mathcal{A}_z(b)}\right)$. To analyze this probability, we define the parameter $q_z := \sum_{y \sim z} \int_{x \in J_{(y,z)} \setminus I_{(y,z)}} 1 dx$. Roughly speaking, we can interpret q_z as the fractional degree of z in the second priority. Note that $q_z \leq f_z$ and the equality holds iff for all $y \sim z$, $I_{(y,z)} \cap J_{(y,z)} = \emptyset$. In Lemma 4.7 we lower-bound q_z in terms of f_z . The following lemma lower-bounds $\mathbb{P}\left(\bigcup_{t=1}^b \mathcal{B}_z(t) \mid \overline{\mathcal{A}_z(b)}\right)$ in terms of f_z , q_z , and the fractional degree of nodes at distance 2 from z .

Lemma 4.3 *For any non-stochastic node z we have:*

$$\mathbb{P}\left(\bigcup_{t=1}^b \mathcal{B}_z(t) \mid \overline{\mathcal{A}_z(b)}\right) \geq \frac{1}{b} \sum_{t=1}^b \sum_{y \sim z} \int_{x \in J_{(y,z)} \setminus I_{(y,z)}} \left(1 - e^{-\frac{t f_{z_1, y(x)}}{b}}\right) dx \left[1 - \frac{q_z}{b}(b-t)\right], \quad (7)$$

Proof: Using inclusion-exclusion principle, we have:

$$\begin{aligned} \mathbb{P}\left(\bigcup_{t=1}^b \mathcal{B}_z(t) \mid \overline{\mathcal{A}_z(b)}\right) &\geq \sum_{t=1}^b \mathbb{P}(\mathcal{B}_z(t) \mid \overline{\mathcal{A}_z(b)}) - \sum_{1 \leq t < u \leq b} \mathbb{P}(\mathcal{B}_z(t) \cap \mathcal{B}_z(u) \mid \overline{\mathcal{A}_z(b)}) \\ &= \sum_{t=1}^b \mathbb{P}(\mathcal{B}_z(t) \mid \overline{\mathcal{A}_z(b)}) \left[1 - \sum_{t < u \leq b} \mathbb{P}(\mathcal{B}_z(u) \mid \overline{\mathcal{A}_z(b)} \cap \mathcal{B}_z(t))\right] \end{aligned}$$

It is sufficient to upper-bound $\mathbb{P}(\mathcal{B}_z(u) \mid \overline{\mathcal{A}_z(b)} \cap \mathcal{B}_z(t))$, and to lower-bound $\mathbb{P}(\mathcal{B}_z(t) \mid \overline{\mathcal{A}_z(b)})$. We start by showing the former, the latter is proved in Lemma 4.4.

The probability that z is tried at time u conditioned on the event $\overline{\mathcal{A}_z(b)}$ is at most the probability that a ball of type y arrives, where $y \sim z$, and a number $x \in J_{(y,z)} \setminus I_{(y,z)}$ is chosen. Note that since

we are conditioning on the event that z is not tried as a first priority, the sampled point cannot belong to $J_{(y,z)} \cap I_{(y,z)}$. By the definition of q_z , the total length of the intervals $J_{(y,z)} \setminus I_{(y,z)}$ for all $y \sim z$ is q_z .

Conditioning on the event $\overline{\mathcal{A}}_z(b)$ implies that during the run of the algorithm, no ball arrives for the subintervals $I_{(y,z)}$ for $y \sim z$. This condition is equivalent to reducing the rate of any such nodes y by $f_{(y,z)}$. In other words, we choose a point in subintervals with total length of $b - f_z$. Hence, the probability that z is tried at time u conditioned on event $\overline{\mathcal{A}}_z(b)$ is at most $\frac{q_z}{b-f_z}$. Since $t < u$, regardless of whether the event $\mathcal{B}_z(t)$ happens or not, the probability of $\mathcal{B}_z(u)$ cannot exceed $\frac{q_z}{b-f_z}$ (i.e. $\mathbb{P}(\mathcal{B}_z(u) | \overline{\mathcal{A}}_z(b) \cap \mathcal{B}_z(t)) \leq \frac{q_z}{b-f_z}$). Since $f_z \leq 1$ we can approximate this by $\frac{q_z}{b}$ with an error term of $o(1/b)$ which we ignore for simplicity.

In lemma 4.4 we lower-bound $\mathbb{P}(\mathcal{B}_z(t) | \overline{\mathcal{A}}_z(b))$ (see inequality 8). Putting these together proves the lemma. □

Lemma 4.4 *For any non-stochastic node z , and any time $1 \leq t \leq b$ we have:*

$$\mathbb{P}(\mathcal{B}_z(t) | \overline{\mathcal{A}}_z(b)) \geq \frac{1}{b} \sum_{y \sim z} \int_{x \in J_{(y,z)} \setminus I_{(y,z)}} \left(1 - e^{-\frac{t f_{z_1, y}(x)}{b}}\right) dx \quad (8)$$

Proof: The event $\mathcal{B}_z(t)$ depends on whether the bins at distance 2 from z are full or not. In order to incorporate the effect of the allocation of these bins on the matching of z at time t , we study the evolution of the density of full bins at distance two from z as follows. For any edge $e = (y, z)$ incident to z , define $F_{(y,z)}(t)$ to be those areas from $J_{(y,z)} \setminus I_{(y,z)}$ whose corresponding first priority bin is full at time t . In other words, $x \in F_{(y,z)}(t)$ if $z_{1,y}(x)$ is full before time t . Also define $\pi_z(t)$ to be the sum of the length of those intervals (i.e. $\pi_z(t) = \sum_{y \sim z} \int_{x \in F_{(y,z)}(t)} 1 dx$). First we show that $\mathbb{P}(\mathcal{B}_z(t) | \overline{\mathcal{A}}_z(b)) = \frac{\mathbb{E}[\pi_z(t)]}{b-f_z}$, then we lower-bound $\mathbb{E}[\pi_z(t)]$.

First observe that the bin z will be tried at time t as a second priority iff a ball of type $y \sim z$ arrives, and we choose $x \in F_{(y,z)}(t)$. Thus the conditional probability that bin z is tried at time t as the second priority is $\mathbb{P}(\mathcal{B}_z(t) | \overline{\mathcal{A}}_z(b) \wedge \pi_z(t)) = \frac{\pi_z(t)}{b-f_z}$. We illustrate this through an example. In the graph of Figure 1 let e_1 be the only edge adjacent to z_1 . Suppose at time t , z_2 is full and z_3 is empty; we want to compute $\mathbb{P}(\mathcal{B}_{z_1}(t) | \overline{\mathcal{A}}_{z_1}(b) \wedge \pi_{z_1}(t))$. We have $F_{(y,z_1)}(t) = [0.5, 0.7]$, and $\pi_{z_1}(t) = 0.2$. Since z_1 will be tried as a second priority only if the arriving ball is of type y and $x \in F_{(y,z_1)}$, we get $\mathbb{P}(\mathcal{B}_{z_1}(t) | \overline{\mathcal{A}}_{z_1}(b) \wedge \pi_{z_1}(t)) = \frac{0.2}{b-0.5}$. By the law of iterative expectations we obtain:

$$\mathbb{P}(\mathcal{B}_z(t) | \overline{\mathcal{A}}_z(b)) \geq \frac{\mathbb{E}[\pi_z(t) | \overline{\mathcal{A}}_z(b)]}{b} \quad (9)$$

It remains to lower-bound $\mathbb{E}[\pi_z(t) | \overline{\mathcal{A}}_z(b)]$. Using definition of $\pi_z(t)$, we write $\mathbb{E}[\pi_z(t) | \overline{\mathcal{A}}_z(b)]$ as:

$$\begin{aligned} \mathbb{E}[\pi_z(t) | \overline{\mathcal{A}}_z(b)] &= \sum_{y \sim z} \int_{x \in J_{(y,z)} \setminus I_{(y,z)}} \mathbb{E}[\mathbb{I}(x \in F_{(y,z)}(t)) | \overline{\mathcal{A}}_z(b)] dx \\ &= \sum_{y \sim z} \int_{x \in J_{(y,z)} \setminus I_{(y,z)}} \mathbb{P}(x \in F_{(y,z)}(t) | \overline{\mathcal{A}}_z(b)) dx \end{aligned} \quad (10)$$

It suffices to lower-bound $\mathbb{P}(x \in F_{(y,z)}(t) | \overline{\mathcal{A}}_z(b))$. As explained above, $F_{(y,z)}(t)$ is a non-decreasing random process that depends on the allocation of the bins at distance 2 from z at time t . For $x \in J_{(y,z)} \setminus I_{(y,z)}$, let $z' = z_{1,y}(x)$. Note that $x \in F_{(y,z)}(t)$ iff z' is full at time t . Thus it suffices to compute the probability that z' is full at time t . Observe that if z' is full at time t , it has been tried at least once as a first or second priority bin. Therefore, the probability of z' being full at time t is at least the probability of event $\mathcal{A}_{z'}(t)$. For simplicity, we ignore the possibility of the trial of z' as a second priority and obtain the following lower bound:

$$\mathbb{P}(x \in F_{(y,z)}(t) | \overline{\mathcal{A}}_z(b)) \geq \mathbb{P}(\mathcal{A}_{z_{1,y}(x)}(t) | \overline{\mathcal{A}}_z(b)) \geq 1 - e^{-\frac{t f_{z_{1,y}(x)}}{b}}.$$

where the last inequality follows from (6). Substituting the RHS into (10) and using (9) imply the Lemma. \square

Putting equations (5), (6), (7) together and using $e^{-f_z} \geq e^{-1}$, we can lower bound the competitive ratio of Algorithm 2:

$$\frac{\mathbb{E}[\text{ALG}]}{\mathbb{E}[\text{OPT}]} \geq \frac{\sum_{z \in Z} \left\{ (1 - e^{-f_z}) + e^{-1} \left[\frac{1}{b} \sum_{t=1}^b \sum_{y \sim z} \int_{x \in J_{(y,z)} \setminus I_{(y,z)}} \left(1 - e^{-\frac{t f_{z_{1,y}(x)}}{b}} \right) dx \left[1 - \frac{q_z}{b}(b-t) \right] \right] \right\}}{\sum_{z \in Z} f_z} \quad (11)$$

In the rest of the proof we show that the ratio attains its minimum when the fractional degree of all non-stochastic nodes are exactly one, i.e., $f_z = 1, \forall z \in Z$. As a warm up, we first analyze this extreme case. We have:

$$\begin{aligned} \frac{\mathbb{E}[\text{ALG}]}{\mathbb{E}[\text{OPT}]} &\geq (1 - e^{-1}) + e^{-1} \left[\frac{1}{b} \sum_{t=1}^b \sum_{y \sim z} \int_{x \in J_{(y,z)} \setminus I_{(y,z)}} (1 - e^{-\frac{t}{b}}) dx \left[1 - \frac{q_z}{b}(b-t) \right] \right] \\ &= (1 - e^{-1}) + e^{-1} \left[\frac{q_z}{b} \sum_{t=1}^b (1 - e^{-\frac{t}{b}}) \left[1 - \frac{q_z}{b}(b-t) \right] \right] \\ &\geq 1 - e^{-1} + q_z e^{-2} - e^{-1} q_z^2 \left(\frac{1}{2} - e^{-1} \right) \geq 0.702, \end{aligned} \quad (12)$$

where the last inequality follows from the observation that for bins with $f_z = 1$ we have $q_z \geq \ln 2$ (see Lemma 4.7 for a proof).

In the remaining parts of the proof we need to show if the fractional degree of some bins are much smaller than 1, still the competitive ratio of the algorithm remains larger than 0.702. Unfortunately, the dependencies between the fractional degree of z and bins at distance 2 from z result in a significant change in the probability of z being matched as a second priority. In particular, if all of the bins at distance 2 from z have a very small rate (i.e. if $f_{z_1} \simeq \frac{1}{n}$), then $\mathbb{P}\left(\bigcup_{t=1}^b \mathcal{B}_z(t) | \overline{\mathcal{A}}_z(b)\right) = O\left(\frac{1}{n}\right)$. This implies that we can not lower bound the RHS of (11) by lower bounding the worst matching probability of a bin. Instead, in the following lemma we write the probability of z being tried as a second priority bin in terms of a **linear** function of f_z, q_z and the fractional degree of bins at distance 2 from z . This will enable us to lower-bound the RHS of (11) by a node based ratio:

Lemma 4.5 For any non-stochastic node z , we have:

$$\mathbb{P}\left(\bigcup_{t=1}^b \mathcal{B}_z(t) \mid \overline{\mathcal{A}}_z(b)\right) \geq q_z e^{-1} - q_z^2 \left(\frac{1}{2} - e^{-1}\right) - e^{-1} \sum_{y \sim z} \int_{x \in J_{(y,z)} \setminus I_{(y,z)}} [1 - f_{z_1,y}(x)] dx. \quad (13)$$

Proof: The proof of this lemma is mainly algebraic. First note that we can write equation (7) as

$$\mathbb{P}\left(\bigcup_{t=1}^b \mathcal{B}_z(t) \mid \overline{\mathcal{A}}_z(b)\right) \geq \sum_{y \sim z} \int_{x \in J_{(y,z)} \setminus I_{(y,z)}} C(f_{z_1,y}(x), q_z) dx \quad (14)$$

where $C(f_{z_1,y}(x), q_z) := \frac{1}{b} \sum_{t=1}^b (1 - e^{-\frac{t f_{z_1,y}(x)}{b}}) (1 - \frac{q_z}{b}(b-t))$, is a concave function of $f_{z_1,y}(x)$; this follows from the fact that $C(\cdot, q_z)$ is a weighted sum of exponential functions with negative weights. Therefore, we can lower-bound $C(\cdot, q_z)$ by a linear function of $f_{z_1,y}(x)$. Since $0 \leq f_{z_1,y}(x) \leq 1$ we have:

$$C(f_{z_1,y}(x), q_z) \geq C(0, q_z) + [C(1, q_z) - C(0, q_z)] f_{z_1,y}(x) = C(1, q_z) f_{z_1,y}(x),$$

where the last equality follows by the observation that $C(0, q_z) = 0$. On the other hand, we have $C(1, q_z) = e^{-1} - q_z(1/2 - e^{-1})$. Therefore:

$$C(f_{z_1,y}(x), q_z) \geq (e^{-1} - q_z(\frac{1}{2} - e^{-1})) f_{z_1,y}(x) \geq e^{-1} - q_z(1/2 - e^{-1}) - e^{-1}[1 - f_{z_1,y}(x)]$$

The lemma simply follows from substituting the above equation in (14), and using the definition of q_z . \square

Substituting (13) in (11), we get:

$$\frac{\mathbb{E}[\text{ALG}]}{\mathbb{E}[\text{OPT}]} \geq \frac{\sum_{z \in Z} \left\{ (1 - e^{-f_z}) + q_z e^{-2} - q_z^2 e^{-1} (\frac{1}{2} - e^{-1}) - e^{-2} \sum_{e \sim z} \int_{x \in J_e \setminus I_e} [1 - f_{z_1,y}(x)] dx \right\}}{\sum_{z \in Z} f_z}$$

Next we rearrange the last term of the numerator to eliminate all dependencies between the fractional degree of z and the bins at distance 2 from z . This enables us to analyze the competitive ratio of the algorithm by the worst case ratio among all bins. We can write:

$$\sum_{z \in Z} \sum_{y \sim z} \int_{x \in J_{(y,z)} \setminus I_{(y,z)}} [1 - f_{z_1,y}(x)] dx = \sum_{z \in Z} \sum_{y \sim z} \int_{x \in I_{(y,z)} \setminus J_{(y,z)}} [1 - f_{z_1,y}(x)] dx,$$

Here the equality follows from the observation that for all $y \in Y$, both sides are integrating over all $x \in [0, r_y]$ where $z_{1,y}(x) \neq z_{2,y}(x)$. Since for any $x \in I_{(y,z)} \setminus J_{(y,z)}$, we have $z_{1,y}(x) = z$, and

$$\sum_{z \in Z} \sum_{y \sim z} \int_{x \in I_{(y,z)} \setminus J_{(y,z)}} [1 - f_{z_1,y}(x)] dx = \sum_{z \in Z} \sum_{y \sim z} \int_{x \in I_{(y,z)} \setminus J_{(y,z)}} [1 - f_z] dx \leq \sum_{z \in Z} f_z [1 - f_z].$$

Therefore, the competitive ratio of the algorithm is at least:

$$\frac{\mathbb{E}[\text{ALG}]}{\mathbb{E}[\text{OPT}]} \geq \min_{z \in Z} \frac{(1 - e^{-f_z}) + q_z e^{-2} - q_z^2 e^{-1} (\frac{1}{2} - e^{-1}) - e^{-2} f_z [1 - f_z]}{f_z}. \quad (15)$$

Since for $0 \leq q_z \leq 1$, the RHS is an increasing function of q_z , any lower-bound on q_z also gives a lower-bound on the competitive ratio of the algorithm. In particular, if $f_z \leq \frac{1}{2}$, we can lower-bound

q_z by zero and we get $\frac{\mathbb{E}[\text{ALG}]}{\mathbb{E}[\text{OPT}]} \geq \frac{1 - e^{-f_z} - e^{-2}f_z[1 - f_z]}{f_z} \geq 0.719$. On the other hand, if $f_z \geq \frac{1}{2}$ we use the lower-bound $q_z \geq \ln 2 + f_z - 1$ (see Lemma 4.7 for the proof), and we obtain that the worst lower-bound is attained for bins with fractional degree 1:

$$\frac{\mathbb{E}[\text{ALG}]}{\mathbb{E}[\text{OPT}]} \geq 1 - e^{-1} + e^{-2} \ln 2 - e^{-1}(\ln 2)^2 \left(\frac{1}{2} - e^{-1}\right) \geq 0.702.$$

This completes the proof of Theorem 4.2. \square

Remark 4.6 *As we discussed earlier (equation (12)) the worst competitive ratio of the algorithm is attained for bins with fractional degree 1, thus the linear bounds used in the proof of Lemma 4.5 does not change worst case analysis of the algorithm.*

Lemma 4.7 *For any non-stochastic node z , we have $q_z \geq \ln 2 + f_z - 1$*

Proof: The proof follows from Observation 4.1 and an optimization over the sampling rate of the neighboring stochastic nodes.

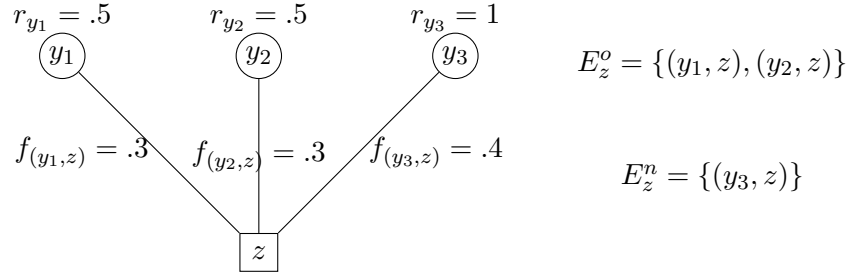


Figure 2: An example of a non-stochastic node z with $|E_z^o| > 1$.

Let E_z be the set of edges incident to z in graph G . We partition E_z into two subsets E_z^o and E_z^n , such that E_z^o consists of edges (y, z) where $f_{(y,z)} > \frac{1}{2}r_y$, and $E_z^n = E_z \setminus E_z^o$ are the rest of the edges. In words, E_z^o is the set of edges e for which $I_{(y,z)}$ and $J_{(y,z)}$ *overlap*. For example, if the rates of all stochastic nodes are 1, for any edge $(y, z) \in E_z^o$ we must have $f_{(y,z)} > \frac{1}{2}$; but since $f_z \leq 1$ we must have $|E_z^o| \leq 1$. However, this is not necessarily true if we allow the stochastic nodes to have arbitrary rates (see Figure 2 for an example). By Observation 4.1, we have:

$$q_z = \sum_{y \sim z} \int_{x \in J_{(y,z)} \setminus I_{(y,z)}} 1 dx = \sum_{y: (y,z) \in E_z^n} f_{(y,z)} + \sum_{y: (y,z) \in E_z^o} r_y - \sum_{y: (y,z) \in E_z^o} f_{(y,z)}. \quad (16)$$

Let f_z^n , r_z^o , and f_z^o be the first, second, and the third summations in the RHS, i.e., $q_z = f_z^n + r_z^o - f_z^o$. By Proposition 2.1 we can show $f_z^o \leq (1 - e^{-r_z^o})$; it is sufficient to replace all stochastic neighbors of z with a super node y^* of rate r_z^o , and use Proposition 2.1 to conclude that

$$f_z^o = f_{(y^*,z)} \leq (1 - e^{-r_{y^*}}) = (1 - e^{-r_z^o}). \quad (17)$$

We can obtain a lower bound on q_z simply by using the above equations and noting that $f_z = f_z^n + f_z^o$:

$$\begin{aligned} q_z &= f_z^n + r_z^o - f_z^o = f_z + r_z^o - 2f_z^o \geq f_z + r_z^o - 2(1 - e^{-r_z^o}) \\ &\geq f_z + \ln 2 - 2 + 2e^{-\ln 2} = f_z + \ln 2 - 1, \end{aligned}$$

where the first equality follows from equation (16), the first inequality follows from equation (17), and the second inequality follows from the fact that $r_z^o = \ln 2$ is the minimizer of $r_z^o + 2e^{-r_z^o}$. \square

Corollary 4.8 *If we restrict the sampling rates of all stochastic nodes to be integral (i.e. r_y 's are integral), then the competitive ratio of Algorithm 2 is at least 0.705.*

Proof: The corollary simply follows from a better lower-bound on q_z in terms of f_z . Since the rates are integral we can show $q_z \geq f_z + 2e^{-1} - 1$; in particular, in the proof of Lemma 4.7 assuming integral rates, we get $r_z^o \in \{0, 1\}$ which implies that $q_z \geq f_z + 1 - 2(1 - e^{-1}) = f_z + 2e^{-1} - 1$. Then the corollary follows from plugging this lower-bound into equation (15). \square

5 Upper Bounds for Online Algorithms

We will present three examples. The first example gives a straightforward $1 - 1/e$ upper bound for the performance of *non-adaptive randomized* algorithms. It shows that when the rates are arbitrarily small, no non-adaptive algorithm can achieve a competitive ratio better than $1 - 1/e$. Note that a randomized non-adaptive algorithm predetermines *distribution* $\mathcal{D}_{y,i}$ for the i -th arrival of type y . In other words, when the i -th ball of type y arrives it will be matched to the neighbor bin z with probability $\mathbb{P}_{y,i}(z)$.

Proposition 5.1 *There is an instance of the online stochastic matching problem with small rates, $r_y = o(1)$, for which no non-adaptive randomized algorithm can achieve a competitive ratio better than $1 - e^{-1}$.*

Proof: Suppose $G(Y, Z, E)$ is a complete bipartite graph, where $|Y| = n^2$ and $|Z| = n = b$; also suppose the rate of all types is $1/n$. Since G is a complete bipartite graph, OPT can easily allocate all the arriving balls and $\mathbb{E}[OPT] = n$. On the other hand, since $r_y = o(1)$, with high probability, there will be at most one ball of each type. Therefore, any non-adaptive randomized algorithm only needs to predetermine one distribution $\mathcal{D}_{y,1}$ for each type y . For each bin $z \in Z$, let p_z be the probability that an incoming ball is matched to z . In other words,

$$p_z = \sum_{y \in Y} r(y) \mathbb{P}_{y,1}(z) = \frac{1}{n} \sum_{y \in Y} \mathbb{P}_{y,1}(z)$$

With probability e^{-p_z} no ball will be matched to the bin z in the run of the process. Thus, $\mathbb{E}[ALG] = \sum_{z \in Z} (1 - e^{-p_z})$. Since function $1 - e^{-x}$ is concave we have:

$$\frac{\mathbb{E}[ALG]}{\mathbb{E}[OPT]} = \frac{\sum_{z \in Z} (1 - e^{-p_z})}{n} \leq (1 - e^{-\frac{1}{n} \sum_{z \in Z} p_z}).$$

On the other hand, we have:

$$\frac{1}{n} \sum_{z \in Z} p_z = \frac{1}{n^2} \sum_{z \in Z} \sum_{y \in Y} \mathbb{P}_{y,1}(z) = \frac{1}{n^2} \sum_{y \in Y} \sum_{z \in Z} \mathbb{P}_{y,1}(z) = 1.$$

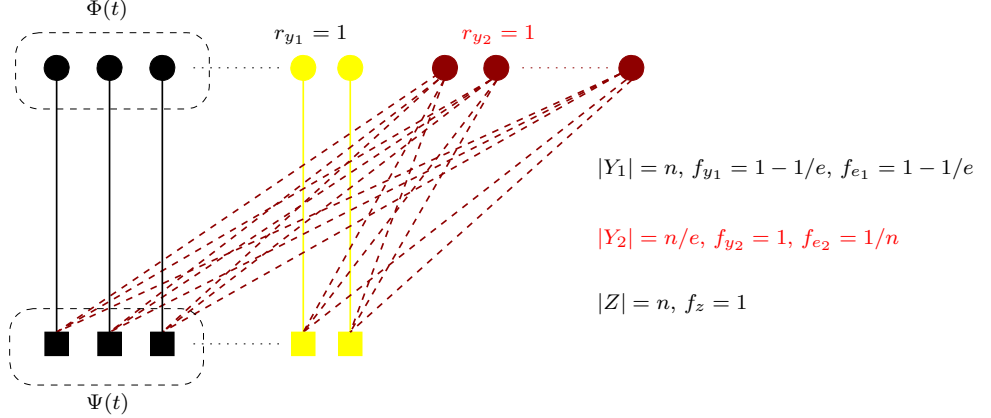


Figure 3: Graph structure for the proof of Proposition 5.2

Therefore, $\frac{\mathbb{E}[\text{ALG}]}{\mathbb{E}[\text{OPT}]} \leq 1 - e^{-1}$ which completes the proof. \square

Our next two examples give an upper bound on the performance of *any* deterministic or randomized online algorithm. In the first example, the rates are integral. Our upper bound of $1 - e^{-2}$ is slightly better than the result of [1].

Proposition 5.2 *There exists an instance of the online stochastic matching problem with integral rates for which no online algorithm can achieve an expected competitive ratio better than $1 - e^{-2} \simeq 0.86$.*

Proof: Construct a bipartite graph $G(Y, Z, E)$, where $Y = Y_1 \cup Y_2$, $|Y_1| = |Z| = n$, and $|Y_2| = n/e$. The set E of edges consists of a perfect matching between the vertices of Y_1 and Z denoted by E_1 , and a complete bipartite graph between Y_2 and Z , denoted by E_2 . See Figure 3.

First, we prove that $\mathbb{E}[\text{OPT}] = n$. Given the sequence of arrivals, first we match through the perfect matching (E_1). In other words, we match one ball of each type $y_1 \in Y_1$. Note that with probability e^{-1} , there will be no ball of type y_1 , thus, in expectation, $(1 - 1/e)$ fraction of the bins will remain empty after matching through E_1 . On the other hand, the expected number of balls of types Y_2 is n/e , which can be matched with the n/e empty bins through the edges of the complete bipartite graph, E_2 . Hence, this simple scheme finds the maximum matching and $\mathbb{E}[\text{OPT}] = n$.

On the other hand, consider an arbitrary online algorithm ALG; at time t , let $\Psi(t) \subseteq Z$ be the set of full (matched) bins, and $\Phi(t) \subseteq Y_1$ be the set of types that have a neighbor in $\Psi(t)$. If the $(t + 1)$ -st ball is of type $\Phi(t)$, it is impossible for ALG to match this ball. Thus:

$$|\Psi(t + 1)| \leq |\Psi(t)| + \mathbb{I}(t + 1^{\text{st}} \text{ ball is not of type } \Phi(t)) \quad (18)$$

Observe that,

$$\mathbb{P}(t + 1^{\text{st}} \text{ ball is not of type } \Psi(t)) = 1 - \frac{|\Psi(t)|}{n(1 + 1/e)}. \quad (19)$$

Note that $b = n(1 + 1/e)$ balls will arrive, thus $\mathbb{E}[\text{ALG}] = \mathbb{E}[|\Psi(n(1 + 1/e))|]$. Taking expectations from both sides of (18) and using (19) result in:

$$\mathbb{E}[\text{ALG}] \leq n(1 + 1/e) \times (1 - 1/e) = (1 - \frac{1}{e^2})n,$$

which proves the claim of the proposition. \square

Our last and probably most interesting example is for general online algorithms, under arbitrary rates. In this example, we use calculations on the size of perfect matchings in random bipartite graphs studied earlier in the context of Random SAT and cuckoo hashing [3, 6, 5].

For a set Z of bins, define Y_k to be a set of $\binom{|Z|}{k}$ vertices, each connected to a distinct subset of cardinality k of Z . These sets will play an important role in constructing examples with large competitive ratio. Let us start with a simple example. Consider an instance of online stochastic matching where $Y = Y_3$, $|Z| = n$. Also suppose that all the rates are equal and $b = 0.9n$, i.e. the rate of each ball $r_y = n / \binom{0.9n}{3}$.

From the perspective of the algorithm, we will have a sequence of $0.9n$ arriving balls each connected to three bins chosen independently and uniformly at random. Because of that, all the empty bins are equivalent; thus the online algorithm can assign the arriving ball to any of its unoccupied neighbors, if there is any. Similar to the proof of Proposition 5.2, let $\Psi(t) \subseteq Z$ be the set of full bins at time t , and $\Phi(t) \subseteq Y$ be the set of types of balls that have no neighbor in $Z \setminus \Psi(t)$ at time t . Note that if the $(t + 1)$ -st ball is of type $\Phi(t)$, it is impossible for any online algorithm to match it. Note that:

$$\mathbb{P}(t + 1^{st} \text{ ball is not of type } \Phi(t)) = 1 - \frac{|\Phi(t)|}{\binom{n}{3}} = 1 - \frac{\binom{|\Psi(t)|}{3}}{\binom{n}{3}}.$$

Thus we can simply write a recurrence relation to compute the expected performance of the online algorithm.

The more difficult part is to compute the optimum solution. The optimum offline algorithm will essentially find the maximum matching between all arrived ball types and the bins. The size of this maximum matching is studied by Path and Rodler [13]. There, the problem is defined as follows: there are b keys to be hashed into n buckets, each capable of holding a single key. Each key has $k \geq 2$ (distinct) associated buckets chosen uniformly at random and independently of the choices of other keys. A hash table can be constructed successfully if each key can be placed into one of its buckets.

Define c_k^* to be the threshold such that if $b/n < c_k^*$ and n is large enough, the resulting bipartite graph has a matching of size b . There has been extensive effort to compute c_k^* [6, 5, 3]. In particular, it has been shown that $c_3^* > 0.91$. Therefore, we can argue that if $b/|Z| = 0.9 < c_3^*$ then the optimum can match all of the balls with high probability. Dietzfelbinger et al. [3] considered an irregular version of the cuckoo hashing, where the number of choices corresponding to a key is a random variable depending on the key. In particular, they considered the case where a key has 2 choices with probability $1/2$ and 3 choices with probability $1/2$ (say 2.5 choices in average), and they defined the number $c_{2.5}^*$ similarly. Interestingly, they show that $c_{2.5}^* \simeq 0.81034$ which is much larger than c_2^* .

In the next proposition we use a combination of the irregular cuckoo hashing idea and the idea of the proof of Proposition 5.2 (adding the type Y_n) to obtain a better upper bound on the performance of optimal online algorithms.

Proposition 5.3 *There is an instance of the online stochastic matching problem for which no algorithm can achieve a competitive ratio better than 0.823.*

Proof: Let $Y = Y_2 \cup Y_3 \cup Y_n$, $|Z| = n$; note that Y_n and Z form a complete bipartite graph. Suppose in expectation we throw $m := 1/2c_{2.5}^*n$ balls of types in Y_2 , m of types in Y_3 and $n - 2m$ of type in Y_n . Therefore, we have $b = n$, and $r_y = m/\binom{n}{2}$ for $y \in Y_2$, $r_y = m/\binom{n}{3}$ for $y \in Y_3$, and $r_y = n - 2m$ for $y \in Y_n$. The optimum offline solution would first match the balls of types in Y_2 and Y_3 , and because the expected number of these balls is at most $c_{2.5}^*n$, it can match all of them with high probability. Then, it matches all the balls of type Y_n to the unoccupied bins. Therefore $\mathbb{E}[\text{OPT}] = n$. Let ALG be an online algorithm and let $\Psi(t)$ and $\Phi(t)$ be defined as above. Similar to the equation (20) we can compute the probability that an incoming ball can be matched by ALG. Note that if a ball of types in Y_n arrives the online algorithm can always match it through the complete graph; on the other hand, if a ball of type Y_2 or Y_3 arrives it can only be matched if it has at least one neighbor in $Z \setminus \Psi(t)$. Note that:

$$\mathbb{P}(\text{the type of } t + 1^{\text{st}} \text{ ball is not in } \Phi(t)) = 1 - \frac{m}{n} \left[\frac{\binom{|\Psi(t)|}{2}}{\binom{n}{2}} + \frac{\binom{|\Psi(t)|}{3}}{\binom{n}{3}} \right]$$

Therefore, we have

$$\begin{aligned} \mathbb{E}[|\Psi(t+1)|] &\leq \mathbb{E}[|\Psi(t)|] + 1 - \frac{m}{n} \mathbb{E} \left[\frac{\binom{|\Psi(t)|}{2}}{\binom{n}{2}} + \frac{\binom{|\Psi(t)|}{3}}{\binom{n}{3}} \right] \\ &\leq \mathbb{E}[|\Psi(t)|] + 1 - \frac{m}{n} \left[\frac{\mathbb{E}[\binom{|\Psi(t)|}{2}]}{\binom{n}{2}} + \frac{\mathbb{E}[\binom{|\Psi(t)|}{3}]}{\binom{n}{3}} \right], \end{aligned}$$

where the last inequality follows from Jensen's inequality. One can numerically compute $\mathbb{E}[|\Psi(n)|]$ and show that $\mathbb{E}[|\Psi(n)|] \leq 0.823n$ for $n > 1000$. Thus for $n > 1000$, we have:

$$\mathbb{E}[\text{ALG}] \leq \mathbb{E}[|\Psi(n)|] \leq 0.823n,$$

which implies that the approximation ratio of the online algorithm is at most 0.823. \square

6 Discussion

We should also point out that competitive analysis is not the only possible or necessarily the most suitable approach for this problem. Because the distribution from which the input is generated is known, one can use dynamic programming (or enumeration of future events) to derive the optimal allocation policy. Unfortunately, the dynamic programming approach takes exponential time. In fact, one can show that the problem of computing the optimal allocation policy is NP-hard. We leave it as an open problem whether it is possible to come up with a polynomial-time algorithm with an approximation guarantee that is better than the best possible competitive ratio for this problem or the competitive ratio that we obtain here.

References

- [1] B. Bahmani and M. Kapralov. Improved bounds for online stochastic matching. In *ESA*, pages 170–181, 2010.

- [2] N. R. Devanur and T. P. Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *EC*, pages 71–78, 2009.
- [3] M. Dietzfelbinger, A. Goerdt, M. Mitzenmacher, A. Montanari, R. Pagh, and M. Rink. Tight thresholds for cuckoo hashing via xorsat. *SIAM Journal on Computing*, 2009.
- [4] J. Feldman, A. Mehta, V. S. Mirrokni, and S. Muthukrishnan. Online stochastic matching: Beating $1-1/e$. In *FOCS*, pages 117–126, 2009.
- [5] N. Fountoulakis and K. Panagiotou. Sharp load thresholds for cuckoo hashing. *arXiv*, cs.DS, Jan. 2009.
- [6] A. Frieze and P. Melsted. Maximum matchings in random bipartite graphs and the space utilization of cuckoo hashtables. arxiv report 0910.5535v3, 2009.
- [7] G. Goel and A. Mehta. Online budgeted matching in random input models with applications to adwords. In *SODA*, pages 982–991, 2008.
- [8] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.
- [9] C. Karande, A. Mehta, and P. Tripathi. Online bipartite matching with unknown distributions. In *STOC*, 2011.
- [10] R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for on-line bipartite matching. In *STOC*, pages 352–358. ACM, 1990.
- [11] M. Mahdian and Q. Yan. Online bipartite matching with random arrivals: A strongly factor revealing lp approach. In *STOC*, 2011.
- [12] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. Adwords and generalized online matching. *J. ACM*, 54(5):22, 2007.
- [13] R. Pagh and F. F. Rodler. Cuckoo hashing. *J. Algorithms*, 51(2):122–144, 2004.