Online Algorithms for Maximum Cardinality Matching with Edge Arrivals*

Niv Buchbinder[1], Danny Segev[2], and Yevgeny Tkach[3]

1 **Department of Statistics and Operations Research, School of Mathematical Sciences, Tel Aviv University, Israel**
niv.buchbinder@gmail.com

2 **Department of Statistics, University of Haifa, Haifa 31905, Israel**
segevd@stat.haifa.ac.il

3 **Department of Statistics and Operations Research, School of Mathematical Sciences, Tel Aviv University, Israel**
ivgitk@gmail.com

Abstract

In the adversarial edge arrival model for maximum cardinality matching, edges of an unknown graph are revealed one-by-one in arbitrary order, and should be irrevocably accepted or rejected. Here, the goal of an online algorithm is to maximize the number of accepted edges while maintaining a feasible matching at any point in time. For this model, the standard greedy heuristic is $1/2$-competitive, and on the other hand, no algorithm that outperforms this ratio is currently known, even for very simple graphs.

We present a clean *Min-Index* framework for devising a family of randomized algorithms, and provide a number of positive and negative results in this context. Among these results, we present a $5/9$-competitive algorithm when the underlying graph is a forest, and prove that this ratio is best possible within the Min-Index framework. In addition, we prove a new general upper bound of $\frac{2}{3+1/\phi^2} \approx 0.5914$ on the competitiveness of any algorithm in the edge arrival model. Interestingly, this bound holds even for an easier model in which vertices (along with their adjacent edges) arrive online, and when the underlying graph is a tree of maximum degree at most 3.

1998 ACM Subject Classification F.1.2 Online computation; G.2.2 Graph algorithms.

Keywords and phrases Maximum matching, online algorithms, competitive analysis, primal-dual method.

Digital Object Identifier 10.4230/LIPIcs.ESA.2017.22

1 Introduction

Graph matchings are cornerstone problems in combinatorial optimization, that have extensively been studied by the discrete mathematics, computer science, and operations research communities. In the most fundamental setting, given an undirected graph $G = (V, E)$, our objective is to identify a maximum cardinality matching, namely, a subset of edges $M \subseteq E$ without any vertices in common. Motivated by emerging applications in online advertising, numerous generalizations of this classic problem have been investigated in the last two decades from the perspective of both offline and online settings.

* The research of Niv Buchbinder is supported by ISF grant 1585/15 and US-Israel BSF grant 2014414. The research of Danny Segev is supported by ISF grant 148/16.

25th Annual European Symposium on Algorithms (ESA 2017).
Editors: Kirk Pruhs and Christian Sohler; Article No. 22; pp. 22:1–22:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In online computation, the seminal work of Karp, Vazirani, and Vazirani [14] studies an online model of maximum cardinality matching in which the underlying graph is bipartite. Specifically, the vertices on one side of the partition are known in advance, whereas those on the other side arrive one-by-one in online fashion. Upon the arrival of a vertex, all its adjacent edges are revealed simultaneously; the algorithm is then required to irrevocably decide how to match the newly arrived vertex. For this setting, Karp et al. designed a randomized $(1 - 1/e)$-competitive algorithm, and showed that the latter factor is best possible. Due to the breadth and depth of subsequent research on this one-sided arrival model, it is beyond the scope of this paper to provide a comprehensive literature review. For this purpose, we refer the reader to a number of selected papers on this topic [12, 3, 18, 1, 6, 7], as well as to additional work on non-adversarial settings, in which the input sequence is randomly generated [10, 5, 9, 15, 13, 16], and finally, to the excellent survey of Mehta [17].

Additional online models, most of which are somewhat more difficult in terms of the achievable competitive ratio, have been proposed in recent years. Wang and Wong [20] introduced a *vertex arrival* model, where vertices from either side of the partition arrive online. Here, whenever a vertex arrives, all edges connecting this vertex to previously arrived vertices are revealed simultaneously. Wang and Wong demonstrated that this model is strictly harder than the one-sided vertex arrival model of Karp et al. [14] by proving an upper bound of $0.6252 < 1 - 1/e$. In addition, they presented a fractional matching algorithm with a competitive ratio of 0.526. An even harder setting is the *edge arrival* model, where edges are revealed one-by-one in arbitrary order, and should be irrevocably accepted or rejected. For this model, the simple greedy heuristic, that deterministically adds an arriving edge to the current matching whenever possible, is $1/2$-competitive. Consequently, the main open question is whether we can attain competitiveness strictly better than $1/2$.

Existing results for relaxed models. As addressing the above question in the edge arrival model seems particularly challenging, recent efforts have mainly concentrated on various relaxations. One such relaxation allows for preemption, where the algorithm is allowed to discard previously accepted edges. For this model, Epstein, Levin, Segev, and Weimann [8] established an upper bound of $\frac{1}{1+\ln 2} \approx 0.591$ on the competitiveness of any algorithm, even on bipartite graphs. To our knowledge, this is the best known upper bound for the edge arrival model without preemption as well. Chiplunkar, Tirodkar, and Vishwanathan [4] designed a $15/28 \approx 0.535$-competitive algorithm for a special case of the vertex arrival model on a tree graph. Very recently, Tirodkar and Vishwanathan [19] devised a $33/64 \approx 0.515$-competitive algorithm for trees in the edge arrival model. As mentioned earlier, these results are heavily based on preemptions. Guruganesh and Singla [11] studied a different type of relaxation, in which edges arrive according to a uniformly-picked random permutation, rather than in an arbitrary adversarial order. Under this assumption, they were able to design a $(1/2 + \delta)$-competitive algorithm, for some absolute constant $\delta > 0$. Nevertheless, for the adversarial edge arrival model, no algorithm that outperforms the basic greedy heuristic is known at present time, even for seemingly-simple network topologies, such as trees or bounded-degree graphs.

The Min-Index framework. In this paper, we consider the adversarial edge arrival model for maximum cardinality matching. Here, a randomized algorithm can be thought of as a procedure that maintains at any given time, explicitly or implicitly, a distribution over matchings. These are updated whenever a new edge arrives, subject to the respective online constraints on the allowable updates. However, since maintaining a general distribution of this

---

Algorithm 1 Min-Index($k, p_1, \dots, p_k$)

Initialization: $M_i \leftarrow \varnothing$, for every $i = 1, \dots, k$.
When edge e arrives:
If e cannot be added to any of the matchings $M_1, \dots, M_k$, reject this edge. Otherwise, update $M_i \leftarrow M_i \cup \{e\}$, where i is the minimal index for which $M_i \cup \{e\}$ is a feasible matching.
Return M_i with probability p_i.

---

nature may be a difficult task, we propose a simple family of randomized algorithms, which is referred to as the *Min-Index framework*. Our generic algorithm maintains a pre-determined distribution over k matchings, $M_1, \dots, M_k$, and associates each matching M_i with a fixed probability p_i, such that $\sum_{i=1}^{k} p_i = 1$. Whenever a new edge arrives, it is greedily accepted to the first matching (index-wise) for which this augmentation is possible; when no such matching exists, the current edge is rejected. As a result, we obtain a clean framework that directly leads to a randomized online matching algorithm, whose formal statement is given in Algorithm 1.

1.1 Our results

Our main contribution is to prove tight upper and lower bounds of $5/9 \approx 0.555$ for the Min-Index framework on forest graphs, as stated in the following theorem.

▶ **Theorem 1.** *For a forest graph, the generic Min-Index algorithm instantiated with $k = 3$ and $(p_1, p_2, p_3) = (5/9, 3/9, 1/9)$ is $5/9$-competitive for the edge arrival model. Moreover, any instantiation of Min-Index, with any number of matchings and respective probabilities, is at most $5/9$-competitive for forest graphs.*

This result improves on that of Chiplunkar et al. [4], who obtained (in a vertex arrival model) a competitive ratio of $15/28 \approx 0.535$ on forests, as well as on the results of Tirodkar and Vishwanathan [19], who obtained (for the edge arrival model) a $33/64 \approx 0.515$-competitive algorithm. In fact, as mentioned earlier, both of these bounds are in an easier model, allowing the online algorithm to preempt edges.

As a warmup, we also show that for graphs of maximum degree 2 (i.e., union of paths and cycles) the Min-Index algorithm with $k = 2$ matchings, picked with probabilities $(p_1, p_2) = (2/3, 1/3)$, is $2/3$-competitive. This result is shown to be best possible for any algorithm in the edge arrival model.

For graphs of maximum degree d, we prove that any instantiation of Min-Index is at most $\frac{1}{2}(1 + \frac{1}{2^d-1})$-competitive, even on bipartite graphs. In spite our best efforts, we could not match this bound. However, inspired by the general idea behind this framework, we design a *fractional* $\frac{1}{2}(1 + \frac{1}{2^d-1})$-competitive algorithm on graphs of maximum degree d. In other words, our online procedure computes a fractional matching whose objective value with respect to the standard LP-relaxation of maximum cardinality matching (see Figure 1) is within factor $\frac{1}{2}(1 + \frac{1}{2^d-1})$ of optimal.

Our final contribution is to establish a general upper bound for any online algorithm.

▶ **Theorem 2.** *The competitive ratio of any fractional (or randomized) online algorithm for maximum matching in the vertex arrival model even for subcubic trees is at most $\frac{2}{3+1/\phi^2} \approx 0.5914$, where $\phi = \frac{1+\sqrt{5}}{2}$ is the golden ratio.*

Interestingly, this result holds even in the vertex arrival model, when the underlying graph is a tree of maximum degree 3. On the one hand, this bound still leaves some marginal

$$
\begin{aligned}
&\text{(P) maximize} && \sum_{e\in E} y_e && \\
&\quad\text{subject to:} && \sum_{e\in\delta(v)} y_e \le 1 && \forall v\in V \\
& && y_e \ge 0 && \forall e\in E
\end{aligned}
\qquad
\begin{aligned}
&\text{(D) minimize} && \sum_{v\in V} x_v && \\
&\quad\text{subject to:} && x_u + x_v \ge 1 && \forall e\in E \\
& && x_v \ge 0 && \forall v\in V
\end{aligned}
$$

Figure 1 The primal matching problem (P) and its dual the vertex cover problem (D).

room for improvements in comparison to our $5/9 \approx 0.555$-competitive algorithm on forests, and our fractional $4/7 \approx 0.571$ algorithm for subcubic graphs (note that $\frac{1}{2}(1+\frac{1}{2^d-1}) = 4/7$ for $d=3$). On the other hand, $\frac{2}{3+1/\phi^2} \approx 0.5914$ improves on the currently best-known upper bound of 0.6252 for the vertex arrival model, due to Wang and Wong [20]. We note in passing that, for the more restrictive edge arrival model (even with preemption), a slightly better upper bound of $\frac{1}{1+\ln 2} \approx 0.5906$ was proven by Epstein et al. [8]. However, their bound holds for high-degree bipartite graphs, while our bound holds even for trees of maximum degree 3.

Organization. All algorithms are given in Section 2, with corresponding upper bounds on the Min-Index framework in Section 3. Our general upper bound is established in Section 4.

1.2 Techniques

The main technical ingredient in proving lower bounds on the competitiveness of our algorithms is based on a primal-dual approach. Specifically, we make use of the standard fractional matching LP and its dual, the fractional vertex cover LP, both stated in Figure 1. To analyze the performance of our algorithms on various classes of graphs, we construct in each setting a feasible fractional vertex cover, that will eventually allow us to bound the expected cardinality of the resulting matching with respect to the optimal vertex cover, and in turn, with respect to the optimal matching via weak duality. In some cases, this construction is performed in an offline fashion, requires complete knowledge of the final input graph, and hence can be viewed as employing a dual-fitting approach. In other cases, our construction is fully online, and therefore also yields a monotonically increasing fractional vertex cover. Such solutions can be rounded online with no loss in optimality on bipartite graphs [20].

To prove upper bounds for the Min-Index algorithm, we construct adversarial sequences that allow us to derive linear inequalities on the achievable competitive ratio in terms of the choice probabilities of the different matchings. These inequalities naturally induce a linear program whose optimal solution provides an upper bound on the best-possible competitiveness. An approach in this spirit has recently been employed by Azar, Cohen, and Roytman [2]. For our general upper bound of $\frac{2}{3+1/\phi^2}$, the adversarial sequences we consider are parameterized according to the phase number m upon which they terminate. As a result, in order to derive an upper bound on the competitiveness of any fractional online algorithm, we are required to solve a corresponding linear program, parameterized by m as well. Rather than solving this LP numerically, we obtain an explicit closed-form solution for its optimum, thereby proposing an analytical proof for the desired upper bound, as the number of phases m tends to infinity.

2 Algorithms

In this section, we establish lower bounds on the competitive ratio of the generic Min-Index framework. Specifically, in Section 2.1, we show that for graphs of maximum degree 2, an appropriate instantiation of the Min-Index algorithm is $2/3$-competitive. In Section 2.2, we prove the first part of Theorem 1, arguing that the right instantiation of the Min-Index algorithm is $5/9$-competitive on forest graphs. Finally, in Section 2.3, we design a fractional $\frac{1}{2}(1+\frac{1}{2^d-1})$-competitive algorithm for graphs of maximum degree at most d.

2.1 A $2/3$-competitive algorithm for graphs of maximum degree 2

As a warm-up, we demonstrate some of our ideas by analyzing how the Min-Index algorithm performs on graphs of maximum degree 2. Such graphs can be viewed as a union of vertex-disjoint cycles and paths, whose edges are revealed to the algorithm one-by-one. Due to space limitations, the proof of Theorem 3 below is omitted. We remark that this proof also shows that a competitive ratio of $3/2$ can be attained for the online fractional vertex cover problem in graphs of maximum degree 2.

▶ **Theorem 3.** *On graphs on maximum degree 2, the Min-Index algorithm with $k = 2$ matchings, picked with probabilities $(p_1, p_2) = (2/3, 1/3)$, is $2/3$-competitive.*

2.2 A $5/9$-competitive algorithm for forest graphs

In this section, we design a randomized $5/9$-competitive algorithm when the underlying graph is a forest. Specifically, as stated in the next theorem, this competitive ratio is attained by our Min-Index algorithm.

▶ **Theorem 4.** *On forests, the Min-Index algorithm with $k = 3$ matchings, picked with probabilities $(p_1, p_2, p_3) = (5/9, 3/9, 1/9)$, is $5/9$-competitive.*

Proof. Clearly, the algorithm returns a feasible matching. Thus, it remains to analyze the expected cardinality of its output matching, given by $p_1 \cdot |M_1| + p_2 \cdot |M_2| + p_3 \cdot |M_3| = \frac{5}{9} \cdot |M_1| + \frac{3}{9} \cdot |M_2| + \frac{1}{9} \cdot |M_3|$. To this end, we use once again a primal-dual approach, by constructing a feasible fractional vertex cover to the dual LP (D), shown in Figure 1. We then prove that the expected cardinality of the matching produced by the algorithm is at least $5/9$ times the value of this fractional vertex cover.

In the (omitted) proof of Theorem 3, we construct a dual solution step-by-step, resulting in an algorithm with a similar competitive ratio for the online fractional vertex cover problem. On the other hand, in this case the (dual) fractional vertex cover is constructed retrospectively, once the input sequence has ended. As the final graph is guaranteed to be a forest, we separately define a feasible vertex cover for each tree of the forest. To this end, consider such a tree, T. We first root T at an arbitrarily-picked vertex r, and orient the edges from the root down toward the leaves, such that each vertex other than r has one ingoing edge. With respect to this orientation, for each edge $e = (u, v)$ that was oriented $u \to v$, we update the fractional vertex cover according to the 4 possible decisions of our algorithm:

1. When $e = u \to v$ is accepted to M_1: $x_u \leftarrow x_u + 3/5$ and $x_v \leftarrow x_v + 2/5$.
2. When $e = u \to v$ is accepted to M_2 (after being rejected from M_1): $x_u \leftarrow x_u + 2/5$ and $x_v \leftarrow x_v + 1/5$.
3. When $e = u \to v$ is accepted to M_3 (after being rejected from both M_1 and M_2): $x_u \leftarrow x_u + 1/5$ and x_v is not updated.
4. When $e = u \to v$ is rejected from M_1, M_2, and M_3: The values x_u and x_v are not updated.

First, we claim that the expected cardinality of the matching produced by our algorithm is precisely $\frac{5}{9}$ times the value of the fractional vertex cover solution we have just constructed. This claim is straightforward, as whenever an edge e is accepted to one of the matchings M_i, corresponding to cases 1-3 above, the expected cardinality of the matching increases by p_i. On the other hand, it is easy to verify that, by our construction, the increase in the fractional vertex cover solution is exactly $\frac{9}{5}p_i$. In case 4, when an edge e is not accepted to any of the matchings, the vertex cover solution remains unchanged. Thus, it remains to prove that the fractional vertex cover is indeed feasible. To this end, we show that $x_u + x_v \geq 1$ for every edge $e = u \to v$ by inspecting the 4 possible decisions of our algorithm.

Case 1: The edge $e = u \to v$ is accepted to M_1. In this case, due to the updates $x_u \leftarrow x_u + 3/5$ and $x_v \leftarrow x_v + 2/5$, we clearly have $x_u + x_v \geq 1$.

Case 2: The edge $e = u \to v$ is accepted to M_2. Since e was rejected from M_1, there must be an edge $e' \in M_1$ that is adjacent to e. By construction, the vertex cover update due to the edge e increases $x_u + x_v$ by $\frac{3}{5}$, whereas that of e' contributes at least $\frac{2}{5}$ to $x_u + x_v$, meaning that e is fractionally covered.

Case 3: The edge $e = u \to v$ is accepted to M_3. Since e was rejected from both M_1 and M_2, there must be a pair of edges, $e_1 \in M_1$ and $e_2 \in M_2$, that are both adjacent to e. Due to the update rule in this case, the edge e caused $x_u + x_v$ to increase by $\frac{1}{5}$, and it remains to show that the combined contribution of e_1 and e_2 to $x_u + x_v$ is at least $\frac{4}{5}$. Note that the orientation of T guarantees that u has at most one ingoing edge. Therefore, at most one of e_1 and e_2 is of the form $w \to u$, and we are left with considering the following cases:

- When $e_1 = w \to u$: Here, e_2 is necessarily of the form $v \to z$ or $u \to z$. It follows that e_1 contributes $\frac{2}{5}$ to x_u whereas e_2 contributes $\frac{2}{5}$ to either x_v or x_u.
- When $e_2 = w \to u$: Then, e_1 is of the form $v \to z$ or $u \to z$. In this case, e_2 contributes $\frac{1}{5}$ to x_u, and e_1 contributes $\frac{3}{5}$ to either x_v or x_u.
- When both e_1 and e_2 are of the form $v \to z$ or $u \to z$: The respective contributions of e_1 and e_2 to $x_u + x_v$ are $\frac{3}{5}$ and $\frac{2}{5}$.

Case 4: The edge $e = u \to v$ is rejected from M_1, M_2, and M_3. Since e is rejected from M_1, M_2, and M_3, this edge must be adjacent to some $e_1 \in M_1$, $e_2 \in M_2$, and $e_3 \in M_3$. We prove that the total contribution of these three edges to $x_u + x_v$ is at least 1. Similar to the argument used in case 3, at most one edge out of e_1, e_2, and e_3 is of the form $w \to u$, and we therefore consider the following cases:

- When $e_1 = w \to u$: The contribution of e_1 to x_u is $\frac{2}{5}$. The contribution of e_2 to x_v or x_u is $\frac{2}{5}$, and the contribution of e_3 to x_v or x_u is $\frac{1}{5}$. Hence, $x_u + x_v \geq 1$.
- When $e_2 = w \to u$: The contribution of e_2 to x_u is $\frac{1}{5}$. The contribution of e_1 to x_v or x_u is $\frac{3}{5}$, and the contribution of e_3 to x_v or x_u is $\frac{1}{5}$. Once again, $x_u + x_v \geq 1$.
- When $e_3 = w \to u$: Even though e_3 does not contribute to x_u, the respective contributions of e_1 and e_2 to $x_v + x_u$ are $\frac{3}{5}$ and $\frac{2}{5}$, implying that $x_u + x_v \geq 1$.
- When e_1, e_2, and e_3 are all of the form $v \to z$ or $u \to z$: The contributions of e_1, e_2, and e_3 to $x_v + x_u$ are $\frac{3}{5}$, $\frac{2}{5}$, and $\frac{1}{5}$, respectively, and we have $x_u + x_v = \frac{6}{5} > 1$. ◀

2.3 Fractional $\frac{1}{2}(1 + \frac{1}{2^d - 1})$-competitiveness for maximum degree d

In this section, we design a $\frac{1}{2}(1 + \frac{1}{2^d - 1})$-competitive algorithm for fractional matching and vertex cover in graphs with maximum degree d, assuming that the value d is known to the

Algorithm 2 Fractional matching and vertex cover for graphs of maximum degree d:

Initialize $y \leftarrow 0$ and $x \leftarrow 0$.
When edge $e = (u, v)$ arrives:
Let $0 \leq i \leq d-1$ be the maximal integer for which $\psi_i \leq 1 - \max\{\sum_{e' \in \delta(u)} y_{e'}, \sum_{e' \in \delta(v)} y_{e'}\}$.
Primal update: Set $y_e = \psi_i = \frac{2^i}{2^d - 1}$.
Dual update: Set $x_u \leftarrow x_u + \frac{2^i}{2^d}$ and $x_v \leftarrow x_v + \frac{2^i}{2^d}$.

algorithm in advance. A fractional algorithm should irrevocably assign each arriving edge e a fraction y_e, subject to the constraint that the total sum of fractions assigned to edges emanating from each vertex v can be at most 1, i.e., $\sum_{e \in \delta(v)} y_e \leq 1$. Although the algorithm proposed here deviates from our general Min-Index framework, it has the same flavor. As shown is Algorithm 2, each new edge is assigned a certain fraction, out of d possible values, which is (greedily) chosen as the largest possible such value. In order to simplify subsequent notation, for $i = 0, 1, \ldots, d-1$, let $\psi_i = \frac{2^i}{2^d - 1}$, noting that $\sum_{i=0}^{d-1} \psi_i = 1$.

At first, it is not clear that our algorithm is well-defined, i.e., that upon the arrival of (u, v) an integer $0 \leq i \leq d-1$ satisfying $\psi_i \leq 1 - \max\{\sum_{e' \in \delta(u)} y_{e'}, \sum_{e' \in \delta(v)} y_{e'}\}$ necessarily exists. The next claim proves this property, which is useful for our analysis later on.

▶ **Lemma 5.** *For every vertex u, as long as fewer than d edges adjacent to u have arrived, we have $1 - \sum_{e \in \delta(u)} y_e \geq \frac{1}{2^d - 1}$.*

Proof. Let us focus on a particular point in time, such that at most $d-1$ edges adjacent to u have arrived thus far. For $0 \leq i \leq d-1$, let a_i be the number of edges $e \in \delta(u)$ for which we currently have $y_e = \psi_i$. With this notation, $\sum_{e \in \delta(u)} y_e = \sum_{i=0}^{d-1} a_i \psi_i \leq 1$ and in addition $\sum_{i=0}^{d-1} a_i \leq d-1$.

Given $a_0, \ldots, a_{d-1}$, we define a corresponding sequence $b_0, \ldots, b_{d-1}$ through the following iterative procedure. Initially, $b_i = a_i$ for every i. Then, while there exists an index i with $b_i \geq 2$, we decrease b_i by 2 and increase b_{i+1} by 1. Since $\psi_{i+1} = 2\psi_i$, this operation keeps the sum $\sum_{i=0}^{d-1} b_i \psi_i$ unchanged (and always equal to $\sum_{i=0}^{d-1} a_i \psi_i$) and strictly decreases $\sum_{i=0}^{d-1} b_i$. It is worth noting that we could never have $b_{d-1} \geq 2$, or otherwise $\sum_{i=0}^{d-1} a_i \psi_i = \sum_{i=0}^{d-1} b_i \psi_i \geq 2\psi_{d-1} = 2 \cdot \frac{2^{d-1}}{2^d - 1} > 1$. At the end of this procedure, each of $b_0, \ldots, b_{d-1}$ takes a binary value, and moreover, $\sum_{i=0}^{d-1} b_i \leq \sum_{i=0}^{d-1} a_i \leq d-1$. The desired claim now follows by observing that

$$\sum_{e \in \delta(u)} y_e = \sum_{i=0}^{d-1} a_i \psi_i = \sum_{i=0}^{d-1} b_i \psi_i \leq \sum_{i=1}^{d-1} \psi_i = \sum_{i=1}^{d-1} \frac{2^i}{2^d - 1} = 1 - \frac{1}{2^d - 1} ,$$

where the above inequality holds since the binary vector with $\sum_{i=0}^{d-1} b_i \leq d-1$ that maximizes $\sum_{i=0}^{d-1} b_i \psi_i = \frac{1}{2^d - 1} \cdot \sum_{i=0}^{d-1} b_i \cdot 2^i$ is clearly $b_0 = 0$ and $b_1 = \cdots = b_{d-1} = 1$. ◀

▶ **Theorem 6.** *On graphs of maximum degree d, Algorithm 2 is $\frac{1}{2}(1 + \frac{1}{2^d - 1})$-competitive for online fractional matching and fractional vertex cover.*

Proof. First, the fractional matching y returned by the algorithm is feasible, as our choice of ψ_i in each step guarantees that the matching constraints are satisfied. In addition, our dual update rule ensures that the total contribution of the edge (u, v) to the fractional vertex cover value is $\Delta x_u + \Delta x_v = \frac{2^i}{2^{d-1}} = \frac{2^d - 1}{2^{d-1}} \cdot y_e = [\frac{1}{2}(1 + \frac{1}{2^d - 1})]^{-1} \cdot y_e$. Thus, the final fractional matching value is exactly $\frac{1}{2}(1 + \frac{1}{2^d - 1})$ times the fractional vertex cover produced by the algorithm. It remains to prove that the latter is indeed feasible.

For this purpose, let $e = (u, v)$ be an edge that has just arrived. We prove that, after its dual update step, this edge is fractionally covered. For any vertex u, let $y_u = \sum_{e' \in \delta(u)} y_{e'}$ be the total fractions assigned to edges adjacent to u just before the arrival of the edge e. Since the input graph is guaranteed to be of degree at most d, by Lemma 5, we necessarily assigned y_e with one of the values $\psi_0, \dots, \psi_{d-1}$. If $y_e = \psi_{d-1}$, then $\Delta x_u = \Delta x_v = \frac{1}{2}$, and we have $x_u + x_v + \Delta x_u + \Delta x_v \geq 1$. In the opposite case, where $y_e = \psi_i$ for some $0 \leq i \leq d-2$, since the edge e could not be assigned with the value ψ_{i+1}, we must have $\max\{y_u, y_v\} > 1 - \frac{2^{i+1}}{2^d-1}$. As $\psi_0, \dots, \psi_{d-1}$ are all integer multiples of $\frac{1}{2^d-1}$, it follows that y_u and y_v are such multiples as well, meaning that the latter inequality implies $\max\{y_u, y_v\} \geq 1 - \frac{2^{i+1}}{2^d-1} + \frac{1}{2^d-1}$. In addition, our primal and dual update rules ensure that $x_u = \frac{2^d-1}{2^d} \cdot y_u$ and hence, $\max\{x_u, x_v\} = \frac{2^d-1}{2^d} \cdot \max\{y_u, y_v\} \geq \frac{2^d-1}{2^d} \cdot (1 - \frac{2^{i+1}-1}{2^d-1})$. By these observations, after the current update we have

$$x_u + x_v + \Delta x_u + \Delta x_v \geq \max\{x_u, x_v\} + \Delta x_u + \Delta x_v \geq \frac{2^d-1}{2^d} \cdot \left(1 - \frac{2^{i+1}-1}{2^d-1}\right) + \frac{2^{i+1}}{2^d} = 1\,. \blacktriangleleft$$

3 Upper Bounds for our Framework

In this section, we prove upper bounds on the competitive ratio of the Min-Index algorithm, as stated in the following theorem.

▶ **Theorem 7.** *For any number of matchings $k \geq 1$ and probabilities $p_1, \dots, p_k$, the Min-Index algorithm is:*

1. *At most $2/3$-competitive on graphs of maximum degree at most 2.*
2. *At most $5/9$-competitive on forest graphs.*
3. *At most $\frac{1}{2}(1 + \frac{1}{2^d-1})$-competitive for bipartite graphs of maximum degree at most d.*

3.1 A $2/3$ upper bound for graphs of maximum degree at most 2

Proof of Theorem 7, Part (1). Consider any instantiation of the Min-Index algorithm that makes use of k matchings with probabilities $p_1, \dots, p_k$. We define two simple adversarial sequences of edge arrivals.

Sequence 1: A single edge $e = (u, v)$ arrives.

Sequence 2: First, an edge $e = (u, v)$ arrives. Then, two additional edges $e_1 = (u, z)$ and $e_2 = (v, w)$ arrive (in any order).

Clearly, both sequences form graphs of maximum degree at most 2. Let c be the competitive ratio of the algorithm. In Sequence 1, the optimal matching consists of the single edge e, whereas Min-Index adds e to M_1 and obtains a matching with expected cardinality p_1, meaning that $c \leq p_1$. In Sequence 2, the optimal matching consists of e_1 and e_2. However, Min-Index adds e to M_1, and subsequently adds e_1 and e_2 to M_2. As a result, a matching with expected cardinality $p_1 + 2p_2$ is obtained, and therefore $c \leq \frac{p_1}{2} + p_2$. To derive an upper bound on the competitive ratio c, it remains to solve the following linear program, where p_1 and p_2 are treated as probabilities (i.e., required to satisfy $p_1 + p_2 \leq 1$ and $p_1, p_2 \geq 0$):

$$\begin{aligned}
\text{maximize} \quad & c \\
\text{subject to} \quad & p_1 \geq c \\
& \frac{p_1}{2} + p_2 \geq c \\
& p_1 + p_2 \leq 1 \\
& p_1, p_2 \geq 0
\end{aligned}$$

The optimal solution to this LP is $p_1 = \frac{2}{3}$, $p_2 = \frac{1}{3}$, and $c = \frac{2}{3}$, concluding our proof. ◀

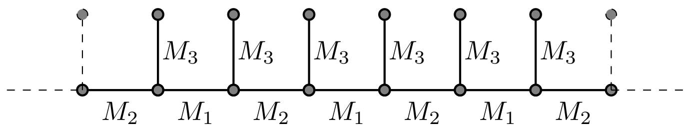

Figure 2 An example for Sequence 2.

▶ Remark. For graphs of maximum degree at most 2, a similar proof actually shows that $2/3$ is the best competitive ratio achievable by any algorithm (not necessarily in our framework), even when the algorithm is allowed to produce a fractional matching.

3.2 A $5/9$ upper bound for forests

Proof of Theorem 7, Part (2). The proof follows the same lines as that of Part (1), with more involved adversarial sequences. Consider an algorithm in our Min-Index framework that makes use of k matchings with probabilities $p_1, \ldots, p_k$. Letting $N = \lceil 1/\epsilon \rceil$, we define the following 3 adversarial sequences of edge arrivals, each forming a tree graph.

Sequence 1: A single edge $e = (u, v)$ arrives.

Sequence 2: In this sequence, a path of length $2N + 1$ is constructed, one edge after the other. By first presenting edges located at even positions of the path, and then those in odd positions, we ensure that the algorithm picks N edges in M_1 and $N + 1$ edges in M_2. Once the entire path is constructed, we add next to each internal vertex u an additional edge (u, v) connecting it to a distinct vertex v. In total, there are $2N$ such edges. The final tree is depicted in Figure 2.

Sequence 3: Figure 3 describes our last sequence. Here, the overall tree is comprised of N copies of a basic gadget, that consists of a 9-edge path with 3 additional edges emanating from middle vertices. Two edges are "going up" from the 5-th and 6-th vertex on the path (marked with bold lines) and another edge is "going down" from the 6-th vertex (marked with dashed lines). The edge arrival sequence proceeds as follows: First, all edges marked with M_1 over all copies arrive, then those marked with M_2, then M_3, and finally M_4. Clearly, Min-Index accepts each edge according to its marked matching, since every edge in M_i is adjacent upon arrival to edges that have already been accepted to $M_1, \ldots, M_{i-1}$. Note that all edges marked with bold lines are accepted to M_4, except for a single edge in the first copy.

Let c be the competitive of the algorithm. To obtain bounds on c in terms of the probabilities $p_1, \ldots, p_4$, for each arrival sequence we compare between the cardinality of the optimal matching and the expected cardinality of the matching produced by the algorithm:

- In Sequence 1, the optimal matching consists of the single edge e, whereas Min-Index adds e to M_1 and obtains a matching with expected cardinality p_1, meaning that $c \le p_1$.
- In Sequence 2, the optimal matching is composed of all $2N$ edges in M_3, whereas the expected cardinality of the matching returned by Min-Index is $N \cdot p_1 + (N+1) \cdot p_2 + 2N \cdot p_3$. Thus, $c \le \frac{1}{2} p_1 + \frac{1}{2} p_2 + p_3 + \frac{1}{2N} p_2 \le \frac{1}{2} p_1 + \frac{1}{2} p_2 + p_3 + \epsilon$, where the last inequality holds since $N = \lceil 1/\epsilon \rceil$.
- In Sequence 3, the optimal matching consists of $6N$ edges, by picking from each gadget the two edges marked in bold and the 1-st, 3-rd, 7-th, and 9-th edges on the path. It is easy to verify that this matching is indeed optimal, as its cardinality is equal to the vertex cover created by picking the 2-nd, 4-th, 5-th, 6-th, 7-th, and 9-th vertices on each path. On the other hand, copies $2, \ldots, N$ of the gadget have 3 edges in M_1, 4 edges in M_2, 3

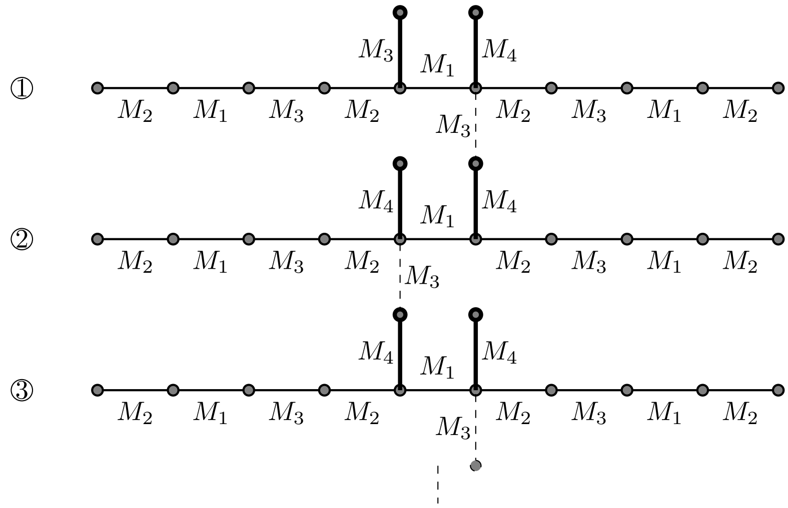

Figure 3 An example for Sequence 3.

edges in M_3, and 2 edges of M_4 each. The first copy has one more edge in M_3 and one less edge in M_4. Thus, the expected cardinality of the matching returned by Min-Index is $N \cdot (3p_1 + 4p_2 + 3p_3 + 2p_4) + p_3 - p_4$. Therefore, $c \le \frac{1}{2}p_1 + \frac{2}{3}p_2 + \frac{1}{2}p_3 + \frac{1}{3}p_4 + \frac{1}{6N}(p_3 - p_4) \le \frac{1}{2}p_1 + \frac{2}{3}p_2 + \frac{1}{2}p_3 + \frac{1}{3}p_4 + \epsilon$, where the last inequality holds since $N = \lceil 1/\epsilon \rceil$.

To obtain an upper bound on the competitive ratio c, we now solve the following linear program, where $p_1, \dots, p_4$ are treated as probabilities:

$$\begin{aligned} \mathrm{LP}(\epsilon) = \text{maximize} \quad & c \\ \text{subject to} \quad & p_1 \ge c \\ & \frac{1}{2}p_1 + \frac{1}{2}p_2 + p_3 + \epsilon \ge c \\ & \frac{1}{2}p_1 + \frac{2}{3}p_2 + \frac{1}{2}p_3 + \frac{1}{3}p_4 + \epsilon \ge c \\ & p_1 + p_2 + p_3 + p_4 \le 1 \\ & p_1, p_2, p_3, p_4 \ge 0 \end{aligned}$$

It is easy to verify that the optimal solution to $\mathrm{LP}(\epsilon)$ has $c \le \frac{5}{9} + \epsilon$. To see this, note that $\mathrm{LP}(0) \le \mathrm{LP}(\epsilon) \le \mathrm{LP}(0) + \epsilon$ and that the optimal solution to $\mathrm{LP}(0)$ is given by $p_1 = \frac{5}{9}$, $p_2 = \frac{3}{9}$, $p_3 = \frac{1}{9}$, $p_4 = 0$, and $c = \frac{5}{9}$. We conclude that $\frac{5}{9}$ is the best competitive ratio achievable through the Min-Index framework, even for trees with maximum degree 4. ◀

3.3 A $\frac{1}{2}(1 + \frac{1}{2^d - 1})$ upper bound for bipartite graphs

Proof of Theorem 7, Part (3). Consider an algorithm in our Min-Index framework that makes use of k matchings with probabilities $p_1, \dots, p_k$. Given an integer parameter d, we define the following d adversarial sequences of edge arrivals, each forming a bipartite graph of maximum degree at most d:

Sequence 1: A single edge $e = (u, v)$ arrives.

Sequences $\ell = 2, \dots, d$: Let G be an $(\ell - 1)$-regular bipartite graph, with n vertices on each side. It is well-known, as an immediate corollary of Hall's Marriage Theorem, that the edge set of such graphs can be partitioned into $\ell - 1$ perfect matchings. In the

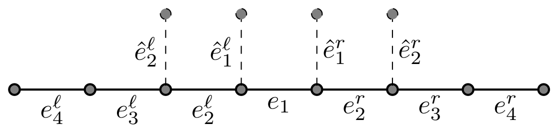

Figure 4 A sequence terminated at round $n = 4$.

sequence of edge arrivals, these matchings are presented one after the other, with an arbitrary order for the edges within each matching. Clearly, Min-Index accepts the first matching into M_1, the second into M_2, and so on. Next, we create a new edge emanating from each of the $2n$ vertices into a new distinct vertex. As these edges are disjoint and cannot be accepted to any of the matchings $M_1, \ldots, M_{\ell-1}$, they are all accepted into M_ℓ.

Let c be the competitive of the algorithm. To obtain bounds on c in terms of the probabilities $p_1, \ldots, p_k$, for each arrival sequence we compare between the cardinality of the optimal matching and the expected cardinality of the matching produced by the algorithm:

- In Sequence 1, the optimal matching consists of the single edge e, whereas Min-Index adds e to M_1 and obtains a matching with expected cardinality p_1, meaning that $c \leq p_1$.
- In Sequence $2 \leq \ell \leq d$, the optimal matching consists of all $2n$ edges in M_ℓ, whereas the expected cardinality of the matching returned by Min-Index is $n \cdot \sum_{t=1}^{\ell-1} p_t + 2n \cdot p_\ell$. Therefore, $c \leq \frac{1}{2} \cdot \sum_{t=1}^{\ell-1} p_t + p_\ell$.

Multiplying both sides of the upper bound due to Sequence ℓ by $\frac{1}{2^{d-\ell}}$, and summing the resulting inequalities over all $1 \leq \ell \leq d$, we get $\sum_{\ell=1}^{d} p_\ell \geq c \cdot \sum_{\ell=1}^{d} \frac{1}{2^{d-\ell}} = c \cdot (2 - \frac{1}{2^{d-1}})$. Since $\sum_{\ell=1}^{d} p_\ell \leq 1$, it follows that the competitive ratio satisfies $c \leq (2 - \frac{1}{2^{d-1}})^{-1} = \frac{1}{2}(1 + \frac{1}{2^d - 1})$. ◀

4 Upper Bound for any Algorithm

In this section, we present our general upper bound, formally stated in Theorem 2. In particular, we prove that the competitive ratio of any fractional (or randomized) online algorithm for maximum matching is at most $\frac{2}{3+1/\phi^2} \approx 0.5914$, where $\phi = \frac{1+\sqrt{5}}{2}$ is the golden ratio. In fact, this result holds even in the vertex arrival model, when the underlying graph is a tree of maximum degree 3.

We first note that any randomized algorithm induces a marginal expected value of y_e for accepting each edge e. As these marginal values must satisfy the packing constraints of the standard matching linear program (P), shown in Figure 1, they induce a valid fractional algorithm. Therefore, proving an upper bound for fractional online algorithms suffices.

Arrival sequence. To understand the upcoming construction, we advise the reader to consult Figure 4. Consider an adversarial sequence consisting of $2n - 1$ edges (and $2n$ vertices) that eventually forms a path as follows. In the first round, an edge $e_1 = (v_1^\ell, v_1^r)$ arrives. Then, for $i \geq 2$, the i-th round introduces two edges of the form $e_i^\ell = (v_i^\ell, v_{i-1}^\ell)$ and $e_i^r = (v_{i-1}^r, v_i^r)$, that augment the path on both sides. The adversary may terminate the sequence once round n ends. Terminating the sequence for any $n \geq 3$ is done by introducing $2(n-2)$ additional "leaf edges", adjacent to the inner vertices $v_1^\ell, \ldots, v_{n-2}^\ell$ and $v_1^r, \ldots, v_{n-2}^r$. The leaf edges adjacent to v_i^ℓ and v_i^r are denoted by $\hat{e}_i^\ell$ and $\hat{e}_i^r$, respectively.

Upper bound as a linear program. Consider any fractional algorithm. For an edge e, let y_e be the fraction given to this edge. In addition, for $i \geq 2$, the sum of fractions given to the edges e_i^ℓ and e_i^r is denoted by $y_i = y_{e_i^\ell} + y_{e_i^r}$; it is convenient to denote $y_1 = y_{e_1}$ as well.

First observe that, since the algorithm is required to meet the matching constraints $\sum_{e\in\delta(v)} y_e \leq 1$ at any point in time, as soon as e_2^ℓ and e_2^r arrive we must have $y_{e_1} + y_{e_2^\ell} \leq 1$ and $y_{e_1} + y_{e_2^r} \leq 1$. By adding up these inequalities, it follows that

$$2y_1 + y_2 \leq 2 \ . \tag{1}$$

Based on precisely the same logic, for $i \geq 2$, once e_{i+1}^ℓ and e_{i+1}^r arrive we would get $y_{e_i^\ell} + y_{e_{i+1}^\ell} \leq 1$ and $y_{e_i^r} + y_{e_{i+1}^r} \leq 1$, implying in turn that

$$y_i + y_{i+1} \leq 2 \ . \tag{2}$$

Now let c be the competitive ratio of the algorithm. After rounds 1 and 2, the optimal matchings are of cardinality 1 and 2, respectively, and therefore

$$c \leq y_1 \quad \text{and} \quad c \leq \frac{1}{2}(y_1 + y_2) \ . \tag{3}$$

In addition, if the adversarial sequence ends at round $n \geq 3$, the matching constraints due to the inner vertices $v_1^\ell, \ldots, v_{n-2}^\ell$ and $v_1^r, \ldots, v_{n-2}^r$ lead to the aggregate inequality

$$2(n-2) \geq \sum_{i=1}^{n-2} \left(\sum_{e\in\delta(v_i^\ell)} y_e + \sum_{e\in\delta(v_i^r)} y_e \right) = y_{n-1} + 2 \cdot \sum_{i=1}^{n-2} y_i + \sum_{i=1}^{n-2} \left(y_{\hat{e}_i^\ell} + y_{\hat{e}_i^r} \right) \ .$$

Consequently, it follows that the total fractions assigned by the algorithm to all edges is

$$\sum_{i=1}^{n} y_i + \sum_{i=1}^{n-2} \left(y_{\hat{e}_i^\ell} + y_{\hat{e}_i^r} \right) \leq \sum_{i=1}^{n} y_i + 2(n-2) - y_{n-1} - 2 \cdot \sum_{i=1}^{n-2} y_i = 2(n-2) + y_n - \sum_{i=1}^{n-2} y_i \ .$$

However, the optimal matching consists of $2(n-1)$ edges: $e_n^\ell, e_n^r, \hat{e}_1^\ell, \ldots, \hat{e}_{n-2}^\ell, \hat{e}_1^r, \ldots, \hat{e}_{n-2}^r$. Thus, we get the following upper bound on the competitive ratio c:

$$c \leq \frac{1}{2(n-1)} \cdot \left(2(n-2) + y_n - \sum_{i=1}^{n-2} y_i \right) \qquad \forall n \geq 3 \tag{4}$$

To summarize, the competitive ratio is upper bounded by the supremum value of c that satisfies Inequalities (1), (2), (3), and (4), noting that the latter actually provides a separate inequality for each $n \geq 3$. Therefore, any finite subset of these inequalities provides a concrete upper bound on the value c. For every $m \geq 4$, let c_m be the bound attained by the following (finite) LP, consisting of a subset of the constraints that are equivalent to truncating the input sequence after m rounds:

$$\begin{aligned} c_m = \text{maximize} \quad & c \\ \text{subject to} \quad & c \leq y_1 && (5) \\ & c \leq \frac{1}{2}(y_1 + y_2) && (6) \\ & c \leq \frac{1}{2(n-1)} \cdot \left(2(n-2) + y_n - \sum_{i=1}^{n-2} y_i \right) \qquad \forall n = 3, \ldots, m && (7) \\ & y_{m-1} + y_m \leq 2 && (8) \end{aligned}$$

▶ **Lemma 8.** *$c_m = \frac{2F_{m+1}-2}{3F_{m+1}+F_{m-1}-4}$, where F_m is the m-th Fibonacci number.*

Due to space limitations, we omit the proof. As the competitive ratio of any algorithm is at most c_m for any $m \geq 4$, and $\lim_{m\to\infty} \frac{F_{m-1}}{F_{m+1}} = \frac{1}{\phi^2}$, we conclude the proof by observing that, $\lim_{m\to\infty} c_m = \lim_{m\to\infty} \frac{2F_{m+1}-2}{3F_{m+1}+F_{m-1}-4} = \frac{2}{3+1/\phi^2} \approx 0.591372$.

References

1. Gagan Aggarwal, Gagan Goel, Chinmay Karande, and Aranyak Mehta. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1253–1264, 2011.
2. Yossi Azar, Ilan Reuven Cohen, and Alan Roytman. Online lower bounds via duality. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1038–1050, 2017.
3. Niv Buchbinder, Kamal Jain, and Joseph Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *Proceedings of the 15th Annual European Symposium on Algorithms*, pages 253–264, 2007.
4. Ashish Chiplunkar, Sumedh Tirodkar, and Sundar Vishwanathan. On randomized algorithms for matching in the online preemptive model. In *Proceedings of the 23rd Annual European Symposium on Algorithms*, pages 325–336, 2015.
5. Nikhil R. Devanur and Thomas P. Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings 10th ACM Conference on Electronic Commerce*, pages 71–78, 2009.
6. Nikhil R. Devanur and Kamal Jain. Online matching with concave returns. In *Proceedings of the 44th ACM Symposium on Theory of Computing*, pages 137–144, 2012.
7. Nikhil R. Devanur, Kamal Jain, and Robert D. Kleinberg. Randomized primal-dual analysis of RANKING for online bipartite matching. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 101–107, 2013.
8. Leah Epstein, Asaf Levin, Danny Segev, and Oren Weimann. Improved bounds for online preemptive matching. In *Proceedings of the 30th International Symposium on Theoretical Aspects of Computer Science*, pages 389–399, 2013.
9. Jon Feldman, Aranyak Mehta, Vahab S. Mirrokni, and S. Muthukrishnan. Online stochastic matching: Beating $1 - 1/e$. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 117–126, 2009.
10. Gagan Goel and Aranyak Mehta. Online budgeted matching in random input models with applications to adwords. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 982–991, 2008.
11. Guru Prashanth Guruganesh and Sahil Singla. Online matroid intersection: Beating half for random arrival. Preprint, arXiv:1512.06271, 2015.
12. Bala Kalyanasundaram and Kirk Pruhs. An optimal deterministic algorithm for online b-matching. *Theoretical Computer Science*, 233(1-2):319–325, 2000.
13. Chinmay Karande, Aranyak Mehta, and Pushkar Tripathi. Online bipartite matching with unknown distributions. In *Proceedings of the 43rd ACM Symposium on Theory of Computing*, pages 587–596, 2011.
14. Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, pages 352–358, 1990.
15. Mohammad Mahdian and Qiqi Yan. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps. In *Proceedings of the 43rd ACM Symposium on Theory of Computing*, pages 597–606, 2011.
16. Vahideh H. Manshadi, Shayan Oveis Gharan, and Amin Saberi. Online stochastic matching: Online actions based on offline statistics. *Mathematics of Operations Research*, 37(4):559–573, 2012.
17. Aranyak Mehta. Online matching and ad allocation. *Foundations and Trends in Theoretical Computer Science*, 8(4):265–368, 2013.
18. Aranyak Mehta, Amin Saberi, Umesh V. Vazirani, and Vijay V. Vazirani. Adwords and generalized online matching. *Journal of the ACM*, 54(5):22, 2007.

19 Sumedh Tirodkar and Sundar Vishwanathan. Maximum matching on trees in the online preemptive and the incremental dynamic graph models. Preprint, arXiv:1612.05419, 2016.

20 Yajun Wang and Sam Chiu-wai Wong. Two-sided online bipartite matching and vertex cover: Beating the greedy algorithm. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming*, pages 1070–1081, 2015.