2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)

Online Matching with General Arrivals

Buddhima Gamlath*, Michael Kapralov*, Andreas Maggiori*, Ola Svensson* and David Wajc†
* *EPFL. Email: {buddhima.gamlath, michael.kapralov, andreas.maggiori, ola.svensson}@epfl.ch*
† *CMU. Email: dwajc@cs.cmu.edu*

***Abstract*—The online matching problem was introduced by Karp, Vazirani and Vazirani nearly three decades ago. In that seminal work, they studied this problem in bipartite graphs with vertices arriving only on one side, and presented optimal deterministic and randomized algorithms for this setting. In comparison, more general arrival models, such as edge arrivals and general vertex arrivals, have proven more challenging and positive results are known only for various relaxations of the problem. In particular, even the basic question of whether randomization allows one to beat the trivially-optimal deterministic competitive ratio of $1/2$ for either of these models was open. In this paper, we resolve this question for both these natural arrival models, and show the following.**

1) **For edge arrivals, randomization does not help — no randomized algorithm is better than $1/2$ competitive.**
2) **For general vertex arrivals, randomization helps — there exists a randomized $(1/2+\Omega(1))$-competitive online matching algorithm.**

***Keywords*-online algorithms; online matching; edge arrivals; general vertex arrivals**

I. Introduction

Matching theory has played a prominent role in the area of combinatorial optimization, with many applications [26, 28]. Moreover, many fundamental techniques and concepts in combinatorial optimization can trace their origins to its study, including the primal-dual framework [24], proofs of polytopes' integrality beyond total unimodularity [10], and even the equation of efficiency with polytime computability [11].

Given the prominence of matching theory in combinatorial optimization, it comes as little surprise that the maximum matching problem was one of the first problems studied from the point of view of online algorithms and competitive analysis. In 1990, Karp, Vazirani, and Vazirani [23] introduced the online matching problem, and studied it under one-sided bipartite arrivals. For such arrivals, Karp et al. noted that the trivial $1/2$-competitive greedy algorithm (which matches any arriving vertex to an arbitrary unmatched neighbor, if one exists) is optimal among deterministic algorithms for this problem. More interestingly, they provided an elegant randomized online algorithm for this problem, called RANKING, which achieves an optimal $(1-1/e)$ competitive ratio. (This bound has been re-proven many times over the years [3, 8, 9, 13, 18].) Online matching and many extensions of this problem under one-sided bipartite vertex arrivals were widely studied over the years, both under adversarial and stochastic arrival models. See recent work [7, 20, 21, 22] and the excellent survey of Mehta [27] for further references on this rich literature.

Despite our increasingly better understanding of one-sided online bipartite matching and its extensions, the problem of online matching under more general arrival models, including edge arrivals and general vertex arrivals, has remained staunchly defiant, resisting attacks. In particular, the basic questions of whether the trivial $1/2$ competitive ratio is optimal for the adversarial edge-arrival and general vertex-arrival models have remained tantalizing open questions in the online algorithms literature. In this paper, we answer both of these questions.

A. Prior Work and Our Results

Here we outline the most relevant prior work, as well as our contributions. Throughout, we say an algorithm (either randomized or fractional) has *competitive ratio* α, or equivalently is α*-competitive*, if the ratio of the algorithm's value (e.g., expected matching size, or overall value, $\sum_e x_e$) to OPT is at least $\alpha \leqslant 1$ for all inputs and arrival orders. As is standard in the online algorithms literature on maximization problems, we use upper bounds (on α) to refer to hardness results, and lower bounds to positive results.

Edge Arrivals: Arguably the most natural, and the least restricted, arrival model for online matching is the edge arrival model. In this model, edges are revealed one by one, and an online matching algorithm must decide immediately and irrevocably whether to match the edge on arrival, or whether to leave both endpoints free to be possibly matched later.

On the hardness front, the problem is known to be strictly harder than the one-sided vertex arrival model of Karp et al. [23], which admits a competitive ratio of $1-1/e \approx 0.632$. In particular, Epstein et al. [12] gave an upper bound of $\frac{1}{1+\ln 2} \approx 0.591$ for this problem, recently improved by Huang et al. [21] to $2-\sqrt{2} \approx 0.585$. (Both bounds apply even to online algorithms with preemption; i.e., allowing edges to be removed from the matching in favor of a newly-arrived edge.) On the positive side, as pointed out by Buchbinder et al. [4], the edge arrival model has proven challenging, and results beating the $1/2$ competitive ratio were only achieved under various relaxations, including: random order edge arrival [19], bounded number of arrival batches [25], on trees, either with or without preemption

2575-8454/19/$31.00 ©2019 IEEE
DOI 10.1109/FOCS.2019.00011

[4, 29], and for bounded-degree graphs [4]. The above papers all asked whether there exists a randomized $({}^1\!/_2 + \Omega(1))$-competitive algorithm for adversarial edge arrivals (see also Open Question 17 in Mehta's survey [27]).

In this work, we answer this open question, providing it with a strong negative answer. In particular, we show that no online algorithm for *fractional* matching (i.e., an algorithm which immediately and irrevocably assigns values x_e to edge e upon arrival such that $\vec{x}$ is in the fractional matching polytope $\mathcal{P} = \{\vec{x} \geqslant \vec{0} \mid \sum_{e \ni v} x_e \leqslant 1 \ \forall v \in V\}$) is better than ${}^1\!/_2$ competitive. As any randomized algorithm induces a fractional algorithm with the same competitive ratio, this rules out any randomized online matching algorithm which is better than deterministic algorithms.

Theorem 1. *No fractional online algorithm is* $^1\!/_2 + \Omega(1)$ *competitive for online matching under adversarial edge arrivals, even in bipartite graphs.*

This result shows that the study of relaxed variants of online matching under edge arrivals is not only justified by the difficulty of beating the trivial bound for this problem, but rather by its *impossibility*.

General Vertex Arrivals: In the online matching problem under vertex arrivals, vertices are revealed one at a time, together with their edges to their previously-revealed neighbors. An online matching algorithm must decide immediately and irrevocably upon arrival of a vertex whether to match it (or keep it free for later), and if so, who to match it to. The one-sided bipartite problem studied by Karp et al. [23] is precisely this problem when all vertices of one side of a bipartite graph arrive first. As discussed above, for this one-sided arrival model, the problem is thoroughly understood (even down to lower-order error terms [13]). Wang and Wong [30] proved that general vertex arrivals are strictly harder than one-sided bipartite arrivals, providing an upper bound of $0.625 < 1 - {}^1\!/_e$ for the more general problem, later improved by Buchbinder et al. [4] to $\frac{2}{3+\phi^2} \approx 0.593$. Clearly, the general vertex arrival model is no harder than the online edge arrival model but is it *easier*? The answer is "yes" for *fractional* algorithms, as shown by combining our Theorem 1 with the 0.526-competitive fractional online matching algorithm under general vertex arrivals of Wang and Wong [30]. For *integral* online matching, however, the problem has proven challenging, and the only positive results for this problem, too, are for various relaxations, such as restriction to trees, either with or without preemption [4, 5, 29], for bounded-degree graphs [4], or (recently) allowing vertices to be matched during some known time interval [20, 21].

We elaborate on the last relaxation above. In the model recently studied by Huang et al. [20, 21] vertices have both arrival and departure times, and edges can be matched whenever both their endpoints are present. (One-sided vertex arrivals is a special case of this model with all online vertices departing immediately after arrival and offline vertices departing at ∞.) We note that any α-competitive online matching under general vertex arrivals is α-competitive in the less restrictive model of Huang et al. As observed by Huang et al., for their model an optimal approach might as well be greedy; i.e., an unmatched vertex v should always be matched at its departure time if possible. In particular, Huang et al. [20, 21], showed that the RANKING algorithm of Karp et al. achieves a competitive ratio of ≈ 0.567. For general vertex arrivals, however, RANKING (and indeed any maximal matching algorithm) is no better than ${}^1\!/_2$ competitive, as is readily shown by a path on three edges with the internal vertices arriving first. Consequently, new ideas and algorithms are needed.

The natural open question for general vertex arrivals is whether a competitive ratio of $({}^1\!/_2 + \Omega(1))$ is achievable by an *integral* randomized algorithm, without any assumptions (see e.g., [30]). In this work, we answer this question in the affirmative:

Theorem 2. *There exists a* $({}^1\!/_2 + \Omega(1))$*-competitive randomized online matching algorithm for general adversarial vertex arrivals.*

B. Our Techniques

Here we outline the techniques underlying our results.

Edge Arrivals: All prior upper bounds in the online literature [4, 12, 13, 21, 23] can be rephrased as upper bounds for *fractional* algorithms; i.e., algorithms which immediately and irrevocably assign each edge e a value x_e on arrival, so that $\vec{x}$ is contained in the fractional matching polytope, $\mathcal{P} = \{\vec{x} \geqslant \vec{0} \mid \sum_{e \ni v} x_e \leqslant 1 \ \forall v \in V\}$. With the exception of [4], the core difficulty of these hard instances is uncertainty about "identity" of vertices (in particular, which vertices will neighbor which vertices in the following arrivals). Our hardness instances rely on uncertainty about the "time horizon". In particular, the underlying graph, vertex identifiers, and even arrival order are known to the algorithm, but the number of edges of the graph to be revealed (to arrive) is uncertain. Consequently, an α-competitive algorithm must accrue high enough value up to each arrival time to guarantee a high competitive ratio at all points in time. As we shall show, for competitive ratio ${}^1\!/_2 + \Omega(1)$, this goal is at odds with the fractional matching constraints, and so such a competitive ratio is impossible. In particular, we provide a family of hard instances and formulate their prefix-competitiveness and matching constraints as linear constraints to obtain a linear program whose objective value bounds the optimal competitive ratio. Solving the obtained LP's dual, we obtain by weak duality the claimed upper bound on the optimal competitive ratio. See [2, 6, 14] for more examples of the use of LP duality for proofs of hardness results for online problems, first advocated by [2].

General Vertex Arrivals: Our high-level approach here will be to round online a fractional online matching algorithm's output, specifically that of Wang and Wong [30]. While this approach sounds simple, there are several obstacles to overcome. First, the fractional matching polytope is not integral in general graphs, where a fractional matching may have value, $\sum_e x_e$, some $3/2$ times larger than the optimal matching size. (For example, in a triangle graph with value $x_e = 1/2$ for each edge e.) Therefore, any general rounding scheme must lose a factor of $3/2$ on the competitive ratio compared to the fractional algorithm's value, and so to beat a competitive ratio of $1/2$ would require an online fractional matching with competitive ratio $> 3/4 > 1-1/e$, which is impossible. To make matters worse, even in bipartite graphs, for which the fractional matching polytope is integral and offline lossless rounding is possible [1, 17], *online* lossless rounding of fractional matchings is impossible, even under one-sided vertex arrivals [7].

Despite these challenges, we show that a slightly better than $1/2$-competitive fractional matching computed by the algorithm of [30] can be rounded online without incurring too high a loss, yielding $(1/2+\Omega(1))$-competitive randomized algorithm for online matching under general vertex arrivals.

To outline our approach, we first consider a simple method to round matchings online. When vertex v arrives, we pick an edge $\{u, v\}$ with probability $z_u = x_{uv}/\Pr[u \text{ free when } v \text{ arrives}]$, and add it to our matching if u is free.

If $\sum_u z_u \leqslant 1$, this allows us to pick at most one edge per vertex and have each edge $e = \{u, v\}$ be in our matching with the right marginal probability, x_e, resulting in a lossless rounding. Unfortunately, we know of no better-than-$1/2$-competitive fractional algorithm for which this rounding guarantees $\sum_u z_u \leqslant 1$.

However, we observe that, for the correct set of parameters, the fractional matching algorithm of Wang and Wong [30] makes $\sum_u z_u$ close to one, while still ensuring a better-than-$1/2$-competitive fractional solution. Namely, as we elaborate later in Section III-C, we set the parameters of their algorithm so that $\sum_u z_u \leqslant 1 + O(\varepsilon)$, while retaining a competitive ratio of $1/2 + O(\varepsilon)$. Now consider the same rounding algorithm with normalized probabilities: I.e., on v's arrival, sample a neighbor u with probability $z'_u = z_u/\max\{1, \sum_u z_u\}$ and match if u is free. As the sum of z_u's is slightly above one in the worst case, this approach does not drastically reduce the competitive ratio. But the normalization factor is still too significant compared to the competitive ratio of the fractional solution, driving the competitive ratio of the rounding algorithm slightly below $1/2$.

To account for this minor yet significant loss, we therefore augment the simple algorithm by allowing it, with small probability (e.g., say $\sqrt{\varepsilon}$), to sample a second neighbor u_2 for each arriving vertex v, again with probabilities proportional to z'_{u_2}: If the first sampled choice, u_1, is free, we match v to u_1. Otherwise, if the second choice, u_2, is free, we match v to u_2. What is the marginal probability that such an approach matches an incoming vertex v to a given neighbor u? Letting F_u denote the event that u is free when v arrives, this probability is precisely

$$\Pr[F_u] \cdot \left(z'_u + z'_u \cdot \sqrt{\varepsilon} \cdot \sum_w z'_w \cdot (1 - \Pr[F_w \mid F_u]) \right). \quad (1)$$

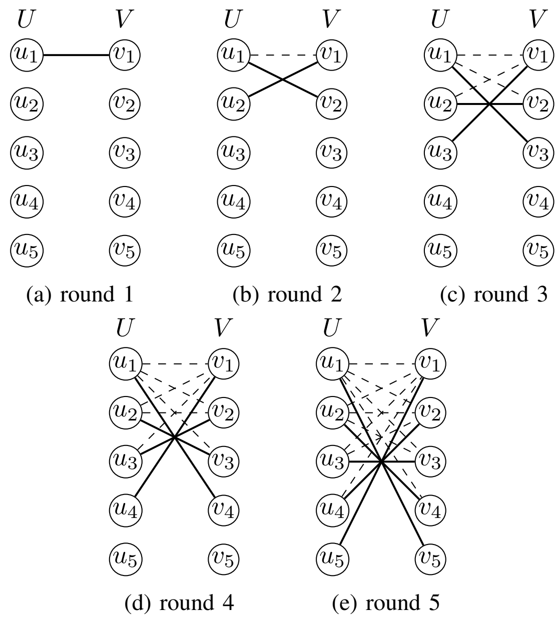
Here the first term in the parentheses corresponds to the probability that v matches to u via the first choice, and the second term corresponds to the same happening via the second choice (which is only taken when the first choice fails).

Ideally, we would like (1) to be at least x_{uv} for all edges, which would imply a lossless rounding. However, as mentioned earlier, this is difficult and in general impossible to do, even in much more restricted settings including one-sided bipartite vertex arrivals. We therefore settle for showing that (1) is at least $x_{uv} = \Pr[F_u] \cdot z_u$ for *most* edges (weighted by x_{uv}). Even this goal, however, is challenging and requires a nontrivial understanding of the correlation structure of the random events F_u. To see this, note that for example if the F_w events are perfectly positively correlated, i.e., $\Pr[F_w \mid F_u] = 1$, then the possibility of picking $e = \{u, v\}$ as a second edge does not increase this edge's probability of being matched *at all* compared to if we only picked a single edge per vertex. This results in e being matched with probability $\Pr[F_u] \cdot z'_u = \Pr[F_u] \cdot z_u/\sum_w z_w = x_{uv}/\sum_w z_w$, which does not lead to any gain over the $1/2$ competitive ratio of greedy. Such problems are easily shown not to arise if all F_u variables are independent or negatively correlated. Unfortunately, positive correlation does arise from this process, and so we the need to control these positive correlations.

The core of our analysis is therefore dedicated to showing that even though positive correlations do arise, they are, by and large, rather weak. Our main technical contribution consists of developing techniques for bounding such positive correlations. The idea behind the analysis is to consider the primary choices and secondary choices of vertices as defining a graph, and showing that after a natural pruning operation that reflects the structure of dependencies, most vertices are most often part of a very small connected component in the graph. The fact that connected components are typically very small is exactly what makes positive correlations weak and results in the required lower bound on (1) for most edges (in terms of x-value), which in turn yields our $1/2 + \Omega(1)$ competitive ratio.

II. Edge Arrivals

In this section we prove the asymptotic optimality of the greedy algorithm for online matching under adversarial edge arrivals. As discussed briefly in Section I, our main idea



(a) round 1 (b) round 2 (c) round 3

(d) round 4 (e) round 5

Figure 1: G_5, together with arrival order. Edges of current (prior) round are solid (dashed).

will be to provide a "prefix hardness" instance, where an underlying input and the arrival order is known to the online matching algorithm, but the prefix of the input to arrive (or "termination time") is not. Consequently, the algorithm must accrue high enough value up to each arrival time, to guarantee a high competitive ratio at all points in time. As we show, the fractional matching constraints rule out a competitive ratio of $1/2 + \Omega(1)$ even in this model where the underlying graph is known.

Theorem 3. *There exists an infinite family of bipartite graphs with maximum degree n and edge arrival order for which any online matching algorithm is at best $\left(\frac{1}{2} + \frac{1}{2n+2}\right)$-competitive.*

Proof: We will provide a family of graphs for which no fractional online matching algorithm has better competitive ratio. Since any randomized algorithm induces a fractional matching algorithm, this immediately implies our claim. The n^{th} graph of the family, $G_n = (U, V, E)$, consists of a bipartite graph with $|U| = |V| = n$ vertices on either side. We denote by $u_i \in U$ and $v_i \in V$ the i^{th} node on the left and right side of G_n, respectively. Edges are revealed in n discrete rounds. In round $i = 1, 2, \ldots, n$, the edges of a perfect matching between the first i left and right vertices arrive in some order. I.e., a matching of $u_1, u_2, \ldots, u_i$ and $v_1, v_2, \ldots, v_i$ is revealed. Specifically, edges (u_j, v_{i-j+1}) for all $i \geqslant j$ arrive. (See Figure 1 for example.) Intuitively, the difficulty for an algorithm attempting to assign high value to edges of OPT is that the (unique) maximum matching OPT changes every round, and no edge ever re-enters OPT.

Consider some α-competitive fractional algorithm $\mathcal{A}$. We call the edge of a vertex w in the (unique) maximum matching of the subgraph of G_n following round i the i^{th} edge of w. For $i \geqslant j$, denote by $x_{i,j}$ the value $\mathcal{A}$ assigns to the i^{th} edge of vertex u_j (and of v_{i-j+1}); i.e., to (u_j, v_{i-j+1}). By feasibility of the fractional matching output by $\mathcal{A}$, we immediately have that $x_{i,j} \geqslant 0$ for all i, j, as well as the following matching constraints for u_j and v_j. (For the latter, note that the i^{th} edge of v_{i-j+1} is assigned value $x_{i,j} = x_{i,i-(i-j+1)+1}$ and so the i^{th} edge of v_j is assigned value $x_{i,i-j+1}$).

$$\sum_{i=j}^{n} x_{i,j} \leqslant 1. \qquad (u_j \text{ matching constraint}) \tag{2}$$

$$\sum_{i=j}^{n} x_{i,i-j+1} \leqslant 1. \qquad (v_j \text{ matching constraint}) \tag{3}$$

On the other hand, as $\mathcal{A}$ is α-competitive, we have that after some k^{th} round – when the maximum matching has cardinality k – algorithm $\mathcal{A}$'s fractional matching must have value at least $\alpha \cdot k$. (Else an adversary can stop the input after this round, leaving $\mathcal{A}$ with a worse than α-competitive matching.) Consequently, we have the following competitiveness constraints.

$$\sum_{i=1}^{k} \sum_{j=1}^{i} x_{i,j} \geqslant \alpha \cdot k \qquad \forall k \in [n]. \tag{4}$$

Combining constraints (2), (3) and (4) together with the non-negativity of the $x_{i,k}$ yields the following linear program, LP(n), whose optimal value upper bounds any fractional online matching algorithm's competitiveness on G_n, by the above.

$$\begin{array}{lll} \text{maximize} & \alpha & \\ \text{subject to:} & \sum_{i=j}^{n} x_{i,j} \leqslant 1 & \forall j \in [n] \\ & \sum_{i=j}^{n} x_{i,i-j+1} \leqslant 1 & \forall j \in [n] \\ & \sum_{i=1}^{k} \sum_{j=1}^{i} x_{i,j} \geqslant \alpha \cdot k & \forall k \in [n] \\ & x_{i,j} \geqslant 0 & \forall i, j \in [n]. \end{array}$$

To bound the optimal value of LP(n), we provide a feasible solution its LP dual, which we denote by Dual(n). By weak duality, any dual feasible solution's value upper bounds the optimal value of LP(n), which in turn upper bounds the optimal competitive ratio. Using the dual variables ℓ_j, r_j for the degree constraints of the j^{th} left and right vertices respectively (u_j and v_j) and dual variable c_k for the competitiveness constraint of the k^{th} round, we get the following dual linear program. Recall here again that $x_{i,i-j+1}$ appears in the matching constraint of v_j, with dual variable r_j, and so $x_{i,j} = x_{i,i-(i-j+1)+1}$ appears in the same constraint for v_{i-j+1}.)

$$\begin{array}{lll} \text{minimize} & \sum_{j=1}^{n}(\ell_j + r_j) & \\ \text{subject to:} & \sum_{k=1}^{n} k \cdot c_k \geqslant 1 & \\ & \ell_j + r_{i-j+1} - \sum_{k=i}^{n} c_k \geqslant 0 & \forall i \in [n], j \in [i] \\ & \ell_j, r_j, c_k \geqslant 0 & \forall j,k \in [n]. \end{array}$$

We provide the following dual solution.

$$c_k = \frac{2}{n(n+1)} \qquad \forall k \in [n]$$

$$\ell_j = r_j = \begin{cases} \frac{n-2(j-1)}{n(n+1)} & \text{if } j \leqslant n/2+1 \\ 0 & \text{if } n/2+1 < j \leqslant n. \end{cases}$$

We start by proving feasibility of this solution. The first constraint is satisfied with equality. For the second constraint, as $\sum_{k=i}^{n} c_k = \frac{2(n-i+1)}{n(n+1)}$ it suffices to show that $\ell_j + r_{i-j+1} \geqslant \frac{2(n-i+1)}{n(n+1)}$ for all $i \in [n], j \in [i]$. Note that if $j > n/2+1$, then $\ell_j = r_j = 0 > \frac{n-2(j-1)}{n(n+1)}$. So, for all j we have $\ell_j = r_j \geqslant \frac{n-2(j-1)}{n(n+1)}$. Consequently, $\ell_j + r_{i-j+1} \geqslant \frac{n-2(j-1)}{n(n+1)} + \frac{n-2(i-j+1-1)}{n(n+1)} = \frac{2(n-i+1)}{n(n+1)}$ for all $i \in [n], j \in [i]$. Non-negativity of the ℓ_j, r_j, c_k variables is trivial, and so we conclude that the above is a feasible dual solution.

It remains to calculate this dual feasible solution's value. We do so for n even,[1] for which

$$\begin{aligned} \sum_{j=1}^{n}(\ell_j + r_j) &= 2 \cdot \sum_{j=1}^{n} \ell_j \\ &= 2 \cdot \sum_{j=1}^{n/2+1} \frac{n-2(j-1)}{n(n+1)} \\ &= \frac{1}{2} + \frac{1}{2n+2}, \end{aligned}$$

completing the proof. ∎

Remark 1. Recall that Buchbinder et al. [4] and Lee and Singla [25] presented better-than-1/2-competitive algorithms for bounded-degree graphs and bounded number of arrival batches. Our upper bound above shows that a deterioration of the competitive guarantees as the maximum degree and number of arrival batches increase (as in the algorithms of [4, 25]) is inevitable.

Remark 2. Recall that the *asymptotic* competitive ratio of an algorithm is the maximum c such that the algorithm always guarantees value at least $ALG \geqslant c \cdot OPT - b$ for some fixed $b > 0$. Our proof extends to this weaker notion of competitiveness easily, by revealing multiple copies of the hard family of Theorem 3 and letting x_{ik} denote the average of its counterparts over all copies.

[1] A similar bound and calculation for odd n holds. As it is unnecessary to establish the result of this theorem, we omit it.

III. General Vertex Arrivals

In this section we present a $(1/2 + \Omega(1))$-competitive randomized algorithm for online matching under general arrivals. As discussed in the introduction, our approach will be to round (online) a *fractional* online matching algorithm's output. Specifically, this will be an algorithm from the family of fractional algorithms introduced in [30]. In Section III-A we describe this family of algorithms. To motivate our rounding approach, in Section III-B we first present a simple lossless rounding method for a 1/2-competitive algorithm in this family. In Section III-C we then describe our rounding algorithm for a better-than-1/2-competitive algorithm in this family. Finally, in Section III-D we present the high level analysis of this rounding scheme. We defer the full proof that this rounding scheme yields a $(1/2 + \Omega(1))$-competitive algorithm to the full version [16].

A. Finding a fractional solution

In this section we revisit the algorithm of Wang and Wong [30], which beats the 1/2 competitiveness barrier for online fractional matching under general vertex arrivals. Their algorithm (technically, family of algorithms) applies the primal-dual method to compute both a fractional matching and a fractional vertex cover – the dual of the fractional matching relaxation. The LPs defining these dual problems are as follows.

Primal-Matching

$$\begin{array}{lll} \text{maximize} & \sum_{e \in E} x_e & \\ \text{subject to:} & \sum_{u \in N(v)} x_{uv} \leqslant 1 & \forall u \in V \\ & x_e \geqslant 0 & \forall e \in E \end{array}$$

Dual-Vertex Cover

$$\begin{array}{lll} \text{minimize} & \sum_{u \in V} y_u & \\ \text{subject to:} & y_u + y_v \geqslant 1 & \forall e = \{u,v\} \in E \\ & y_u \geqslant 0 & \forall u \in V \end{array}$$

Before introducing the algorithm of [30], we begin by defining the fractional online vertex cover problem for vertex arrivals. When a vertex v arrives, if $N_v(v)$ denotes the previously-arrived neighbors of v, then for each $u \in N_v(v)$, a new constraint $y_u + y_v \geqslant 1$ is revealed, which an online algorithm should satisfy by possibly increasing y_u or y_v. Suppose v has its dual value set to $y_v = 1 - \theta$. Then all of its neighbors should have their dual increased to at least θ. Indeed, an algorithm may as well increase y_u to $\max\{y_u, \theta\}$. The choice of θ therefore determines an online fractional vertex cover algorithm. The increase in the dual cost due to the newly-arrived vertex v is thus $1 - \theta + \sum_{u \in N_v(v)} (\theta - y_u)^+$.[2] In [30] θ is chosen to upper bound this term by $1 - \theta + f(\theta)$ for some function $f(\cdot)$. The primal solution (fractional matching) assigns values x_{uv} so as to guarantee feasibility of $\vec{x}$ and a ratio of β between the primal and dual values of $\vec{x}$ and $\vec{y}$, implying $\frac{1}{\beta}$-competitiveness of this online fractional matching algorithm,

[2] Here and throughout the paper, we let $x^+ := \max\{0, x\}$ for all $x \in \mathbb{R}$.

by feasibility of $\vec{y}$ and weak duality. The algorithm, parameterized by a function $f(\cdot)$ and parameter β to be discussed below, is given formally in Algorithm 1. In the subsequent discussion, $N_v(u)$ denotes the set of neighbors of u that arrive before v.

Input : A stream of vertices $v_1, v_2, \ldots v_n$. At step i, vertex v_i and $N_{v_i}(v_i)$ are revealed.
Output: A fractional vertex cover solution $\vec{y}$ and a fractional matching $\vec{x}$.

Let $y_u \leftarrow 0$ for all u, let $x_{uv} \leftarrow 0$ for all u, v.
foreach *v in the stream* **do**
  maximize θ
  subject to: $\theta \leqslant 1$
  $\sum_{u \in N_v(v)} (\theta - y_u)^+ \leqslant f(\theta)$
  foreach $u \in N_v(v)$ **do**
    $x_{uv} \leftarrow \frac{(\theta - y_u)^+}{\beta} \left(1 + \frac{1-\theta}{f(\theta)}\right)$.
    $y_u \leftarrow \max\{y_u, \theta\}$.
  $y_v \leftarrow 1 - \theta$.

Algorithm 1: Online general vertex arrival fractional matching and vertex cover

Algorithm 1 is parameterized by a function f and a constant β. The family of functions considered by [30] are as follows.

Definition 4. *Let*

$$f_\kappa(\theta) := \left(\frac{1+\kappa}{2} - \theta\right)^{\frac{1+\kappa}{2\kappa}} \left(\theta + \frac{\kappa - 1}{2}\right)^{\frac{\kappa-1}{2\kappa}}.$$

We define $W := \{f_\kappa \mid \kappa \geqslant 1\}$.

As we will see, choices of β guaranteeing feasibility of $\vec{x}$ are related to the following quantity.

Definition 5. *For a given $f : [0,1] \to \mathbb{R}_+$ let*

$$\beta^*(f) := \max_{\theta \in [0,1]} 1 + f(1-\theta) + \int_\theta^1 \frac{1-t}{f(\theta)}\, d\theta.$$

For functions $f \in W$ this definition of $\beta^*(f)$ can be simplified to $\beta^*(f) = 1 + f(0)$, due to the observation (see [30, Lemmas 4,5]) that all functions $f \in W$ satisfy

$$\beta^*(f) = 1 + f(1-\theta) + \int_\theta^1 \frac{1-\theta}{f(\theta)}\, d\theta \qquad \forall \theta \in [0,1]. \quad (5)$$

As mentioned above, the competitiveness of Algorithm 1 for appropriate choices of f and β is obtained by relating the overall primal and dual values, $\sum_e x_e$ and $\sum_v y_v$. As we show (and rely on later), one can even bound individual vertices' contributions to these sums. In particular, for any vertex v's arrival time, each vertex's contribution to $\sum_e x_e$, can be bounded in terms of its dual variable's value by this point, y_u, as follows.

Lemma 6. *For any vertex $u, v \in V$, let y_u be the dual variable of u prior to arrival of v. Then the fractional degree just before v arrives, $x_u := \sum_{w \in N_v(u)} x_{uw}$, is bounded as follows:*

$$\frac{y_u}{\beta} \leqslant x_u \leqslant \frac{y_u + f(1-y_u)}{\beta}.$$

Broadly, the lower bound on x_u is obtained by lower bounding the increase x_u by the increase to y_u/β after each vertex arrival, while the upper bound follows from a simplification of a bound given in [30, Invariant 1] (implying feasibility of the primal solution), which we simplify using (5). See the full version ([16]) for a full proof.

Another observation we will need regarding the functions $f \in W$ is that they are decreasing.

Observation 7. *Every function $f \in W$ is non-increasing in its argument in the range $[0,1]$.*

Proof: As observed in [30], differentiating (5) with respect to z yields $-f'(1-z) - \frac{1-z}{f(z)} = 0$, from which we obtain $f(z) \cdot f'(1-z) = z - 1$. Replacing z by $1-z$, we get $f(1-z) \cdot f'(z) = -z$, or $f'(z) = -\frac{z}{f(1-z)}$. As $f(z)$ is positive for all $z \in [0,1]$, we have that $f'(z) < 0$ for all $z \in [0,1]$. ■

The next lemma of [30] characterizes the achievable competitiveness of Algorithm 1.

Lemma 8 ([30])**.** *Algorithm 1 with function $f \in W$ and $\beta \geqslant \beta^*(f) = 1 + f(0)$ is $\frac{1}{\beta}$ competitive.*

Wang and Wong [30] showed that taking $\kappa \approx 1.1997$ and $\beta = \beta^*(f_\kappa)$, Algorithm 1 is ≈ 0.526 competitive. In later sections we show how to round the output of Algorithm 1 with f_κ with $\kappa = 1 + 2\varepsilon$ for some small constant ε and $\beta = 2 - \varepsilon$ to obtain a $(1/2 + \Omega(1))$-competitive algorithm. But first, as a warm up, we show how to round this algorithm with $\kappa = 1$ and $\beta = \beta^*(f_1) = 2$.

B. Warmup: a 1/2-competitive randomized algorithm

In this section we will round the 1/2-competitive fractional algorithm obtained by running Algorithm 1 with function $f(\theta) = f_1(\theta) = 1 - \theta$ and $\beta = \beta^*(f) = 2$. We will devise a lossless rounding of this fractional matching algorithm, by including each edge e in the final matching with a probability equal to the fractional value x_e assigned to it by Algorithm 1. Note that if v arrives after u, then if F_u denotes the event that u is free when v arrives, then edge $\{u, v\}$ is matched by an online algorithm with probability $\Pr[\{u,v\} \in M] = \Pr[\{u,v\} \in M \mid F_u] \cdot \Pr[F_u]$. Therefore, to match each edge $\{u,v\}$ with probability x_{uv}, we need $\Pr[\{u,v\} \in M \mid F_u] = x_{uv} / \Pr[F_u]$. That is, we must match $\{u,v\}$ with probability $z_u = x_{uv} / \Pr[F_u]$ conditioned on u being free. The simplest way of doing so (if possible) is to pick an edge $\{u,v\}$ with the above probability z_u always, and to match it only if u is free. Algorithm 2 below does just this, achieving a lossless rounding of this fractional algorithm. As before, $N_v(u)$ denotes the set of neighbors of u that arrive before v.

Input : A stream of vertices $v_1, v_2, \ldots, v_n$. At step i, vertex v_i and $N_{v_i}(v_i)$ are revealed.
Output: A matching M.
1 Let $y_u \leftarrow 0$ for all u, let $x_{uv} \leftarrow 0$ for all u, v.
2 Let $M \leftarrow \emptyset$.
3 **foreach** *v in the stream* **do**
4   Update y_u's and x_{uv}'s using Algorithm 1 with $\beta = 2$ and $f = f_1$.
5   **foreach** $u \in N_v(v)$ **do**
6         $z_u \leftarrow \frac{x_{uv}}{\Pr[u \text{ is free when } v \text{ arrives}]}$.
          `//` $z_u = x_{uv}/(1 - y_u)$`, as shown later`
7   Sample (at most) one neighbor $u \in N_v(v)$ according to z_u.
8   **if** *a free neighbor u is sampled* **then**
9   Add $\{u, v\}$ to M.

Algorithm 2: Online vertex arrival warmup randomized fractional matching

Algorithm 2 is well defined if for each vertex v's arrival, z is a probability distribution; i.e., $\sum_{u \in N_v(v)} z_u \leqslant 1$. The following lemma asserts precisely that. Moreover, it asserts that Algorithm 2 matches each edge with the desired probability.

Lemma 9. *Algorithm 2 is well defined, since for every vertex v on arrival, z is a valid probability distribution. Moreover, for each v and $u \in N_v(v)$, it matches edge $\{u, v\}$ with probability x_{uv}.*

Proof: We prove both claims in tandem for each v, by induction on the number of arrivals. For the base case (v is the first arrival), the set $N_v(v)$ is empty and thus both claims are trivial. Consider the arrival of a later vertex v. By the inductive hypothesis we have that each vertex $u \in N_v(v)$ is previously matched with probability $\sum_{w \in N_v(u)} x_{wu}$. But by our choice of $f(\theta) = f_1(\theta) = 1 - \theta$ and $\beta = 2$, if w arrives after u, then y_u and θ at arrival of w satisfy $x_{uw} = \frac{(\theta - y_u)^+}{\beta} \cdot \left(1 + \frac{1-\theta}{f(\theta)}\right) = (\theta - y_u)^+$. That is, x_{uw} is precisely the increase in y_u following arrival of w. On the other hand, when u arrived we have that its dual value y_u increased by $1 - \theta = \sum_{v' \in N_u(u)} (\theta - y_{v'})^+ = \sum_{v' \in N_u(u)} x_{uv'}$. To see this last step, we recall first that by definition of Algorithm 1 and our choice of $f(\theta) = 1 - \theta$, the value θ on arrival of v is chosen to be the largest $\theta \leqslant 1$ satisfying

$$\sum_{u \in N_v(v)} (\theta - y_u)^+ \leqslant 1 - \theta. \tag{6}$$

But the inequality (6) is an equality whether or not $\theta = 1$ (if $\theta = 1$, both sides are zero). We conclude that $y_u = \sum_{v' \in N_v(u)} x_{uv'}$ just prior to arrival of v. But then, by the inductive hypothesis, this implies that $\Pr[u \text{ free when } v \text{ arrives}] = 1 - y_u$ (yielding an easily-computable formula for z_u). Consequently, by (6) we have that when v arrives z is a probability distribution, as

$$\begin{aligned}
\sum_{u \in N_v(v)} z_u &= \sum_{u \in N_v(v)} \frac{(\theta - y_u)^+}{1 - y_u} \\
&\leqslant \sum_{u \in N_v(v):\, y_u \leqslant \theta} \frac{(\theta - y_u)^+}{1 - \theta} \\
&= \sum_{u \in N_v(v)} \frac{(\theta - y_u)^+}{1 - \theta} \\
&\leqslant 1.
\end{aligned}$$

Finally, for u to be matched to a latter-arriving neighbor v, it must be picked, and free when v arrives. These events happen independently with probabilities $\frac{x_{uv}}{\Pr[u \text{ is free when } v \text{ arrives}]}$ and $\Pr[u \text{ is free when } v \text{ arrives}]$ respectively. Consequently, each edge $\{u, v\}$ is indeed matched with probability precisely

$$\Pr[\{u, v\} \in M] = x_{uv}.$$

■

In the next section we present an algorithm which allows to round better-than-1/2-competitive algorithms derived from Algorithm 1.

C. An improved algorithm

In this section, we build on Algorithm 2 and show how to improve it to get a $(1/2 + \Omega(1))$ competitive ratio.

There are two concerns when modifying Algorithm 2 to work for a general function from the family W. The first is how to compute the probability that a vertex u is free when vertex v arrives, in Line 6. In the simpler version, we inductively showed that this probability is simply $1 - y_u$, where y_u is the dual value of u as of v's arrival (see the proof of Lemma 9). With a general function f, this probability is no longer given by a simple formula. Nevertheless, it is easily fixable: We can either use Monte Carlo sampling to estimate the probability of u being free at v's arrival to a given inverse polynomial accuracy, or we can in fact exactly compute these probabilities by maintaining their marginal values as the algorithm progresses. In what follows, we therefore assume that our algorithm can compute these probabilities exactly.

The second and more important issue is with the sampling step in Line 7. In the simpler algorithm, this step is well-defined as the sampling probabilities indeed form a valid distribution: I.e., $\sum_{u \in N_v(v)} z_u \leqslant 1$ for all vertices v. However, with a general function f, this sum can exceed one, rendering the sampling step in Line 7 impossible. Intuitively, we can normalize the probabilities to make it

Input : A stream of vertices $v_1, v_2, \ldots, v_n$. At step i, vertex v_i and $N_{v_i}(v)$ are revealed.
Output: A matching M.

1 Let $y_u \leftarrow 0$ for all u, let $x_{uv} \leftarrow 0$ for all u, v.
2 Let $M \leftarrow \emptyset$.
3 **foreach** *v in the stream* **do**
4   Update y_u's and x_{uv}'s using Algorithm 1 with $\beta = 2 - \varepsilon$ and $f = f_{1+2\varepsilon}$.
5   **foreach** $u \in N_v(v)$ **do**
        /* Compute Pr[u is free when v arrives] as explained in Section III-C */
6         $z_u \leftarrow \frac{x_{uv}}{\Pr[u \text{ is free when } v \text{ arrives}]}$.
7   **foreach** $u \in N_v(v)$ **do**
8         $z'_u \leftarrow z_u / \max\left\{1, \sum_{u \in N_v(v)} z_u\right\}$.
9   Pick (at most) one $u_1 \in N_v(v)$ with probability z'_{u_1}.
10   **if** $\sum_{u \in N_v(v)} z_u > 1$ **then**
11   With probability $\sqrt{\varepsilon}$, pick (at most) one $u_2 \in N_v(v)$ with probability z'_{u_2}.
        /* Probability of dropping edge $\{u, v\}$ can be computed using (7). */
12   Drop u_2 with minimal probability ensuring $\{u_2, v\}$ is matched with probability at most $x_{u_2 v}$.
13   **if** *a free neighbor u_1 is sampled* **then**
14   Add $\{u_1, v\}$ to M.
15   **else if** *a free neighbor u_2 is sampled* **then**
16   Add $\{u_2, v\}$ to M.

Algorithm 3: A randomized online matching algorithm under general vertex arrivals.

a proper distribution, but by doing so, we end up losing some amount from the approximation guarantee. We hope to recover this loss using a second sampling step, as we mentioned in Section I-B and elaborate below.

Suppose that, instead of $\beta = 2$ and $f = f_1$ (i.e., the function $f(\theta) = 1 - \theta$), we use $f = f_{1+2\varepsilon}$ and $\beta = 2 - \varepsilon$ to define x_{uv} and y_u values. As we show later in this section, for an ε sufficiently small, we then have $\sum_{u \in N_v(v)} z_u \leqslant 1 + O(\varepsilon)$, implying that the normalization factor is at most $1 + O(\varepsilon)$. However, since the approximation factor of the fractional solution is only $1/2 + O(\varepsilon)$ for such a solution, (i.e., $\sum_{\{u,v\} \in E} x_{uv} \geqslant (1/\beta) \cdot \sum_{u \in V} y_u$), the loss due to normalization is too significant to ignore.

Now suppose that we allow arriving vertices to sample a second edge with a small (i.e., $\sqrt{\varepsilon}$) probability and match that second edge if the endpoint of the first sampled edge is already matched. Consider the arrival of a fixed vertex v such that $\sum_{u \in N_v(v)} z_u > 1$, and let z'_u denote the normalized z_u values. Further let F_w denote the event that vertex w is free (i.e, unmatched) at the arrival of v. Then the probability that v matches u for some $u \in N_v(v)$ using either of the two sampled edges is

$$\Pr[F_u] \cdot \left(z'_u + z'_u \sqrt{\varepsilon} \cdot \sum_{w \in N_v(v)} z'_w \cdot (1 - \Pr[F_w \mid F_u]) \right), \tag{7}$$

which is the same expression from (1) from Section I-B, restated here for quick reference. Recall that the first term inside the parentheses accounts for the probability that v matches u via the first sampled edges, and the second term accounts for the probability that the same happens via the second sampled edge. Note that the second sampled edge is used only when the first one is incident to an already matched vertex and the other endpoint of the second edge is free. Hence we have the summation of conditional probabilities in the second term, where the events are conditioned on the other endpoint, u, being free. If the probability given in (7) is x_{uv} for all $\{u, v\} \in E$, we would have the same guarantee as the fractional solution x_{uv}, and the rounding would be lossless. This seems unlikely, yet we can show that the quantity in (7) is at least $(1 - \varepsilon^2) \cdot x_{uv}$ for most (not by number, but by the total fractional value of x_{uv}'s) of the edges in the graph, showing that our rounding is *almost* lossless. We postpone further discussion of the analysis to Section III-D where we highlight the main ideas before proceeding with the formal proof.

Our improved algorithm is outlined in Algorithm 3. Up until Line 6, it is similar to Algorithm 2 except that it uses $\beta = 2 - \varepsilon$ and $f = f_{1+2\varepsilon}$ where we choose $\varepsilon > 0$ to be any constant small enough such that the results in the analysis hold. In Line 8, if the sum of z_u's exceeds one we normalize the z_u to obtain a valid probability distribution z'_u. In Line 9, we sample the first edge incident to an arriving vertex v. In Line 11, we sample a second edge incident to the same vertex with probability $\sqrt{\varepsilon}$ if we had to scale down z_u's in Line 8. Then in Line 12, we drop the sampled second edge with the minimal probability to ensure that no edge $\{u, v\}$ is matched with probability more than x_{uv}. Since (7) gives the exact probability of $\{u, v\}$ being matched, this probability of dropping an edge $\{u, v\}$ can be computed by the algorithm. However, to compute this, we need the conditional probabilities $\Pr[F_w \mid F_u]$, which again can be estimated using Monte Carlo sampling[3]. In the subsequent lines, we match v to a chosen free neighbor (if any) among its chosen neighbors, prioritizing its first choice.

For the purpose of analysis we view Algorithm 3 as constructing a greedy matching on a directed acyclic graph (DAG) H_τ defined in the following two definitions.

[3]It is also possible to compute them exactly if we allow the algorithm to take exponential time.

Definition 10 (Non-adaptive selection graph G_τ)**.** *Let τ denote the random choices made by the vertices of G. Let G_τ be the DAG defined by all the arcs (v, u_1), (v, u_2) for all vertices $v \in V$. We call the arcs (v, u_1)* primary *arcs, and the arcs (v, u_2) the* secondary *arcs.*

Definition 11 (Pruned selection graph H_τ)**.** *Now construct H_τ from G_τ by removing all arcs (v, u) (primary or secondary) such that there exists a* primary *arc (v', u) with v' arriving before v. We further remove a secondary arc (v, u) if there is a primary arc (v, u); i.e., if a vertex u has at least one incoming primary arc, remove all incoming primary arcs that came after the first primary arc and all secondary arcs that came after or from the same vertex as the first primary arc.*

It is easy to see that the matching constructed by Algorithm 3 is a greedy matching constructed on H_τ based on order of arrival and prioritizing primary arcs. The following lemma shows that the set of matched vertices obtained by this greedy matching does not change much for any change in the random choices of a single vertex v, which will prove useful later on. It can be proven rather directly by an inductive argument showing the size of the symmetric difference in matched vertices in G_τ and $G_{\tau'}$ does not increase after each arrival besides the arrival of v, whose arrival clearly increases this symmetric difference by at most two. See full version for details.

Lemma 12. *Let G_τ and $G_{\tau'}$ be two realizations of the random digraph where all the vertices in the two graphs make the same choices except for* one *vertex v. Then the number of vertices that have different* matched *status (free/matched) in the matchings computed in H_τ and $H_{\tau'}$ at any point of time is at most two.*

D. High-Level Description of Analysis

In this section we outline the analysis of Algorithm 3, highlighting its main ideas. See the full version for full details [16].

As described in Section III-C, the main difference compared to the simpler $1/2$-competitive algorithm is the change of the construction of the fractional solution, which in turn makes the rounding more complex. In particular, we may have at the arrival of a vertex v that $\sum_{u \in N_v(v)} z_u > 1$. The majority of the analysis is therefore devoted to such "problematic" vertices since otherwise, if $\sum_{u \in N_v(v)} z_u \leqslant 1$, the rounding is lossless due to the same reasons as described in the simpler setting of Section III-B. We now outline the main ideas in analyzing a vertex v with $\sum_{u \in N_v(v)} z_u > 1$. Let F_w be the event that vertex w is free (i.e., unmatched) at the arrival of v. Then, as described in Section III-C, the probability that we select edge $\{u, v\}$ in our matching is the minimum of x_{uv} (because of the pruning in Line 12), and

$$\Pr[F_u] \cdot \left(z'_u + z'_u \sqrt{\varepsilon} \cdot \sum_{w \in N_v(v)} z'_w \cdot (1 - \Pr[F_w \mid F_u]) \right).$$

By definition, $\Pr[F_u] \cdot z_u = x_{uv}$, and the expression inside the parentheses is at least z_u (implying $\Pr[\{u, v\} \in M] = x_{uv}$) if

$$1 + \sqrt{\varepsilon} \cdot \sum_{w \in N_v(v)} z'_w \cdot (1 - \Pr[F_w \mid F_u]) \geqslant \frac{z_u}{z'_u}. \tag{8}$$

To analyze this inequality, we first use the structure of the selected function $f = f_{1+2\varepsilon}$ and the selection of $\beta = 2 - \varepsilon$ to show that if $\sum_{w \in N_v(v)} z_w > 1$ then several structural properties hold (see full version). In particular, there are absolute constants $0 < c < 1$ and $C > 1$ (both independent of ε) such that for small enough (constant) $\varepsilon > 0$

1) $\sum_{w \in N_v(v)} z_w \leqslant 1 + C\varepsilon$;
2) $z_w \leqslant C\sqrt{\varepsilon}$ for every $w \in N_v(v)$; and
3) $c \leqslant \Pr[F_w] \leqslant 1 - c$ for every $w \in N_v(v)$.

The first property implies that the right-hand-side of (8) is at most $1 + C\varepsilon$; and the second property implies that v has at least $\Omega(1/\sqrt{\varepsilon})$ neighbors and that each neighbor u satisfies $z'_u \leqslant z_u \leqslant C\sqrt{\varepsilon}$.

For simplicity of notation, we assume further in the high-level overview that v has exactly $1/\sqrt{\varepsilon}$ neighbors and each $u \in N_v(v)$ satisfies $z'_u = \sqrt{\varepsilon}$. Inequality (8) would then be implied by

$$\sum_{w \in N_v(v)} (1 - \Pr[F_w \mid F_u]) \geqslant C\,. \tag{9}$$

To get an intuition why we would expect the above inequality to hold, it is instructive to consider the unconditional version:

$$\sum_{w \in N_v(v)} (1 - \Pr[F_w]) \geqslant c|N_v(v)| = c/\sqrt{\varepsilon} \gg C\,,$$

where the first inequality is from the fact that $\Pr[F_w] \leqslant 1 - c$ for any neighbor $w \in N_v(v)$. The large slack in the last inequality, obtained by selecting $\varepsilon > 0$ to be a sufficiently small constant, is used to bound the impact of conditioning on the event F_u. Indeed, due to the large slack, we have that (9) is satisfied if the quantity $\sum_{w \in N_v(v)} \Pr[F_w | F_u]$ is not too far away from the same summation with unconditional probabilities, i.e., $\sum_{w \in N_v(v)} \Pr[F_w]$. Specifically, it is sufficient to show

$$\sum_{w \in N_v(v)} (\Pr[F_w | F_u] - \Pr[F_w]) \leqslant c/\sqrt{\varepsilon} - C\,. \tag{10}$$

We do so by bounding the correlation between the events F_u and F_w in a highly non-trivial manner, which constitutes the heart of our analysis. The main challenges are that events F_u and F_w can be positively correlated and that, by conditioning on F_u, the primary and secondary choices of different vertices are no longer independent.

We overcome the last difficulty by replacing the conditioning on F_u by a conditioning on the component in H_τ

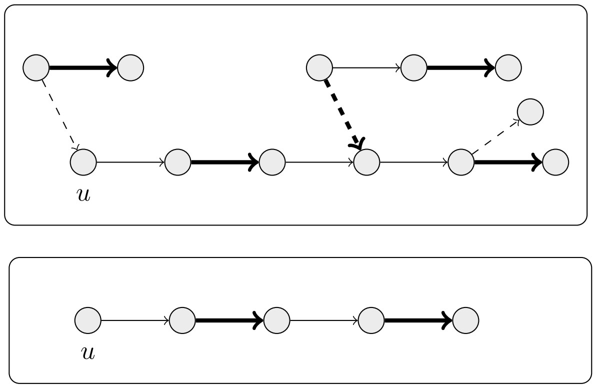


Figure 2: Two examples of the component of H_τ containing u. Vertices are depicted from right to left according to their arrival order. Primary and secondary arcs are solid and dashed, respectively. The edges that take part in the matching are thick.

(at the time of v's arrival) that includes u. As explained in Section III-C, the matching output by our algorithm is equivalent to the greedy matching constructed in H_τ and so the component containing u (at the time of v's arrival) determines F_u. But how can this component look like, assuming the event F_u? First, u cannot have any incoming primary arc since then u would be matched (and so the event F_u would be false). However, u could have incoming secondary arcs, assuming that the tails of those arcs are matched using their primary arcs. Furthermore, u can have an outgoing primary and possibly a secondary arc if the selected neighbors are already matched. These neighbors can in turn have incoming secondary arcs, at most one incoming primary arc (due to the pruning in the definition of H_τ), and outgoing primary and secondary arcs; and so on. In Figure 2, we give two examples of the possible structure, when conditioning on F_u, of u's component in H_τ (at the time of v's arrival). The left example contains secondary arcs, whereas the component on the right is arguably simpler and only contains primary arcs.

An important step in our proof is to prove that, for most vertices u, the component is of the simple form depicted to the right with probability almost one. That is, it is a path P consisting of primary arcs, referred to as a primary path that further satisfies:

(i) it has length $O(\ln(1/\varepsilon))$; and
(ii) the total z-value of the arcs in the blocking set of P is $O(\ln(1/\varepsilon))$. Informally, the blocking set contains those arcs that if appearing as primary arcs in G_τ would cause arcs of P to be pruned (or blocked) from H_τ.

Let $\mathcal{P}$ be the primary paths of above type that appear with positive probability as u's component in H_τ. Further let EQ_P be the event that u's component equals P. Then we show (for most vertices) that $\sum_{P\in\mathcal{P}} \Pr[\mathrm{EQ}_P \mid F_u]$ is almost one. For simplicity, let us assume here that the sum is equal to one. Then by the law of total probability and since $\sum_{P\in\mathcal{P}} \Pr[\mathrm{EQ}_P \mid F_u] = 1$, we have that, denoting the set $N_v(v)$ by S,

$$\begin{aligned}&\sum_{w\in S} (\Pr[F_w \mid F_u] - \Pr[F_w])\\ &= \sum_{P\in\mathcal{P}} \Pr[\mathrm{EQ}_P \mid F_u] \cdot \sum_{w\in S} (\Pr[F_w \mid F_u, \mathrm{EQ}_P] - \Pr[F_w])\,.\end{aligned}$$

But since the component P containing u determines F_u (i.e., it determines whether or not u is matched), the above can be simplified to

$$\begin{aligned}&\sum_{w\in N_v(v)} (\Pr[F_w \mid F_u] - \Pr[F_w])\\ &= \sum_{P\in\mathcal{P}} \Pr[\mathrm{EQ}_P \mid F_u] \cdot \sum_{w\in N_v(v)} (\Pr[F_w \mid \mathrm{EQ}_P] - \Pr[F_w])\,.\end{aligned}$$

The proof is then completed by analyzing the term inside the parentheses for each primary path $P \in \mathcal{P}$ separately. As we prove in the full version, the independence of primary and secondary arc choices of vertices is maintained after conditioning on EQ_P.[4] Furthermore, we show that there is a bijection between the outcomes of the unconditional and the conditional distributions, so that the expected number of vertices that make different choices under this pairing can be upper bounded by roughly the length of the path plus the z-value of the edges in the blocking set. So, for a path P as above (for which these edges' total z-value is $O(\ln(1/\varepsilon))$), we have that the expected number of vertices that make different choices in the paired outcomes is $O(\ln(1/\varepsilon))$ which. By Lemma 12, this implies that the expected number of vertices that change matched status if we condition on EQ_P is also upper bounded by $O(\ln(1/\varepsilon))$. In other words, we have for every path $P \in \mathcal{P}$ as above that

$$\begin{aligned}&\sum_{w\in N_v(v)} (\Pr[F_w \mid \mathrm{EQ}_P] - \Pr[F_w])\\ &\leqslant \sum_{w\in V} (\Pr[F_w \mid \mathrm{EQ}_P] - \Pr[F_w])\\ &= O(\ln(1/\varepsilon)),\end{aligned}$$

which implies (10) for a small enough choice of ε. This completes the overview of the main steps in the analysis. The main difference in the formal proof is that not all vertices satisfy that their component is a short primary path with probability close to 1. To that end, we define the notion of *good* vertices, which are the vertices that are very unlikely to have long directed paths of primary arcs rooted at them. These are exactly the vertices v for which we can perform the above analysis for most neighbors u (in the proof of

[4]To be precise, conditioning on a primary path P with a so-called termination certificate T (see the full version). In the overview, we omit this detail and consider the event $\mathrm{EQ}_{P,T}$ (instead of EQ_P) in the formal proof.

the “key lemma”) implying that the rounding is almost lossless for v. Then, we show using a rather simple charging scheme that most of the vertices in the graph are good. Finally, putting everything together yields Theorem 2. See full version for details [16].

IV. Conclusion and Open Questions

In this work we resolve the open questions of whether the greedy algorithm is optimal for online matching under edge and vertex arrivals. There are still many questions left unanswered. We mention a few here.

For bipartite graphs with one-sided vertex arrivals the RANKING algorithm of Karp et al. [23] yields an optimal competitive ratio. In the same work, Karp et al. asked whether RANKING is optimal in general graphs. While there are several ways one could generalize their algorithm to an algorithm for general graphs, it seems that all would result in a greedy algorithm with some random tie breaking. For general vertex arrivals, it is easy to see that this would result in a competitive ratio of $1/2$. Our algorithm therefore shows that RANKING is sub-optimal for general vertex arrivals. What is the optimal competitive ratio achievable for this arrival model?

Next, for edge arrivals, our impossibility result further motivates the study of this arrival model under relaxations. One particularly appealing question is whether one can outperform greedy with *preemption* (see e.g., [5, 12, 29]). The power of preemption also still remains to be determined for edge-weighted matching under bipartite one-sided vertex arrivals [15].

Acknowledgements: The work of Michael Kapralov was supported in part by ERC Starting Grant 759471. The work of Ola Svensson was supported by the Swiss National Science Foundation project 200021-184656 “Randomness in Problem Instances and Randomized Algorithms”. This work done in part while David Wajc was visiting EPFL. The work of David Wajc was supported in part by NSF grants CCF-1527110, CCF-1618280, CCF-1814603, CCF-1910588, NSF CAREER award CCF-1750808 and a Sloan Research Fellowship.

References

[1] Ageev, A. A. and Sviridenko, M. I. 2004. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization 8,* 3, 307–328.

[2] Azar, Y., Cohen, I. R., and Roytman, A. 2017. Online lower bounds via duality. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA).* 1038–1050.

[3] Birnbaum, B. and Mathieu, C. 2008. On-line bipartite matching made simple. *ACM SIGACT News 39,* 1, 80–87.

[4] Buchbinder, N., Segev, D., and Tkach, Y. 2018. Online algorithms for maximum cardinality matching with edge arrivals. *Algorithmica*, 1–19.

[5] Chiplunkar, A., Tirodkar, S., and Vishwanathan, S. 2015. On randomized algorithms for matching in the online preemptive model. In *Proceedings of the 23rd Annual European Symposium on Algorithms (ESA).* 325–336.

[6] Cohen, I. R., Peng, B., and Wajc, D. 2019. Tight bounds for online edge coloring. In *Proceedings of the 60th Symposium on Foundations of Computer Science (FOCS).* To Appear.

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

[8] Devanur, N. R., Jain, K., and Kleinberg, R. D. 2013. Randomized primal-dual analysis of ranking for online bipartite matching. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA).* 101–107.

[9] Eden, A., Feldman, M., Fiat, A., and Segal, K. 2018. An economic-based analysis of ranking for online bipartite matching. *arXiv preprint arXiv:1804.06637*.

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

[11] Edmonds, J. 1965b. Paths, trees, and flowers. *Canadian Journal of mathematics 17,* 3, 449–467.

[12] Epstein, L., Levin, A., Segev, D., and Weimann, O. 2013. Improved bounds for online preemptive matching. In *Proceedings of the 30th International Symposium on Theoretical Aspects of Computer Science (STACS).* 389.

[13] Feige, U. 2018. Tighter bounds for online bipartite matching. *arXiv preprint arXiv:1812.11774*.

[14] Feldkord, B., Feldotto, M., Gupta, A., Guruganesh, G., Kumar, A., Riechers, S., and Wajc, D. 2018. Fully-dynamic bin packing with little repacking. In *Proceedings of the 45th International Colloquium on Automata, Languages and Programming (ICALP).* 51:1–51:24.

[15] Feldman, J., Korula, N., Mirrokni, V., Muthukrishnan, S., and Pál, M. 2009. Online ad assignment with free disposal. In *Proceedings of the 5th Conference on Web and Internet Economics (WINE).* 374–385.

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

[17] Gandhi, R., Khuller, S., Parthasarathy, S., and Srinivasan, A. 2006. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM (JACM) 53,* 3, 324–360.

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

[19] GURUGANESH, G. P. AND SINGLA, S. 2017. Online matroid intersection: Beating half for random arrival. In *Proceedings of the 19th Conference on Integer Programming and Combinatorial Optimization (IPCO)*. 241–253.

[20] HUANG, Z., KANG, N., TANG, Z. G., WU, X., ZHANG, Y., AND ZHU, X. 2018a. How to match when all vertices arrive online. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC)*. 17–29.

[21] HUANG, Z., PENG, B., TANG, Z. G., TAO, R., WU, X., AND ZHANG, Y. 2019. Tight competitive ratios of classic matching algorithms in the fully online model. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2875–2886.

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

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

[24] KUHN, H. W. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly 2,* 1-2, 83–97.

[25] LEE, E. AND SINGLA, S. 2017. Maximum matching in the online batch-arrival model. In *Proceedings of the 19th Conference on Integer Programming and Combinatorial Optimization (IPCO)*. 355–367.

[26] LOVÁSZ, L. AND PLUMMER, M. D. 2009. *Matching theory*. Vol. 367. American Mathematical Society.

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

[28] SCHRIJVER, A. 2003. *Combinatorial optimization: polyhedra and efficiency*. Vol. 24. Springer Science & Business Media.

[29] TIRODKAR, S. AND VISHWANATHAN, S. 2017. Maximum matching on trees in the online preemptive and the incremental dynamic graph models. In *Proceedings of the 23rd International Computing and Combinatorics Conference (COCOON)*. 504–515.

[30] WANG, Y. AND WONG, S. C.-W. 2015. Two-sided online bipartite matching and vertex cover: Beating the greedy algorithm. In *Proceedings of the 42nd International Colloquium on Automata, Languages and Programming (ICALP)*. 1070–1081.