A Randomness Threshold for Online Bipartite Matching, via Lossless Online Rounding

Niv Buchbinder*[1], Joseph (Seffi) Naor†[2], and David Wajc‡[3]

[1]Tel Aviv University, niv.buchbinder@gmail.com
[2]Technion, naor@cs.technion.ac.il
[3]Stanford University, wajc@stanford.edu

Abstract

Over three decades ago, Karp, Vazirani and Vazirani (STOC'90) introduced the online bipartite matching problem. They observed that deterministic algorithms' competitive ratio for this problem is no greater than $1/2$, and proved that randomized algorithms can do better. A natural question thus arises: *how random is random*? i.e., how much randomness is needed to outperform deterministic algorithms? The RANKING algorithm of Karp et al. requires $\tilde{O}(n)$ random bits, which, ignoring polylog terms, remained unimproved. On the other hand, Pena and Borodin (TCS'19) established a lower bound of $(1 - o(1)) \log\log n$ random bits for any $1/2 + \Omega(1)$ competitive ratio.

We close this doubly-exponential gap, proving that, surprisingly, this lower bound is tight. In fact, we prove a *sharp threshold* of $(1 \pm o(1)) \log\log n$ random bits for the randomness necessary and sufficient to outperform deterministic algorithms for this problem, as well as its vertex-weighted generalization. Our work implies the same threshold for the advice complexity (nondeterminism) of these problems.

Similar to recent breakthrough results in the online matching literature, for the edge-weighted matching problem (Fahrbach et al. FOCS'20) and adwords without the small bids assumption (Huang et al. FOCS'20), our algorithms break the barrier of $1/2$ by randomizing matching choices over two neighbors. Unlike these works, our approach does not rely on the recently-introduced OCS machinery, nor the more established randomized primal-dual method. Instead, our work revisits a highly-successful online design technique, which was nonetheless under-utilized in the area of online matching, namely (lossless) online rounding of fractional algorithms. While this technique is known to be hopeless for online matching in general, we show that it is nonetheless applicable to carefully designed fractional algorithms with additional (non-convex) constraints.

*Supported in part by by Israel Science Foundation grant 2233/19 and United States - Israel Binational Science Foundation grant 2018352.

†Supported in part by by Israel Science Foundation grant 2233/19 and United States - Israel Binational Science Foundation grant 2018352.

‡Supported in part by NSF Award CCF1812919, ONR award N000141912550, and a gift from Cisco Research.

1 Introduction

We study the online bipartite matching problem and its vertex-weighted generalization. Online bipartite matching was introduced three decades ago by Karp, Vazirani, and Vazirani [35], who presented optimal deterministic and randomized algorithms for this problem, i.e., with competitive ratios $1/2$ and $1 - 1/e$, respectively. This problem and its extensions have been widely studied over the years; see Mehta [38] for a survey of earlier work, and for more recent developments see, e.g., [18, 22, 25, 29, 30, 31, 33, 34, 43]. This line of work has led to a deeper understanding of the power of computation in online settings, and has introduced a number of powerful algorithmic techniques. Most prominent among these techniques is the randomized primal-dual framework of Devanur, Jain, and Kleinberg [19], which has proven tremendously useful in recent years [22, 28, 29, 30, 31, 33, 34, 37, 50]. Despite the progress on many fundamental questions in the area, many open questions still remain, both on the results side, and on the techniques side. We address a number of such questions in this paper, focusing on the following two.

The Power of Randomness and Nondeterminism. A core goal of online algorithms research is determining the power of randomized algorithms compared to deterministic algorithms. A refinement of this question is determining the *amount* of randomness needed to outperform deterministic algorithms. A related question is studied in the *advice model*, introduced by Emek et al. [21]. In this model a deterministic online algorithm is equipped with an advice string, and the algorithm's guarantees for any input are with respect to the best advice string for the input. Non-deterministic advice bits are always at least as powerful as random bits, and in many cases are strictly more powerful [9, 10, 40, 45, 46, 47].

For online bipartite matching Karp et al. [35] proved that any deterministic online bipartite matching algorithm is at best $1/2$ competitive, and designed a $(1 - 1/e)$-competitive randomized algorithm, using $\log(n!) = O(n \log n)$ bits of randomness. The amount of randomness needed to outperform deterministic algorithms was slightly improved later by [20], who gave a $(1 - 1/e - \epsilon)$-competitive algorithm using $O_\epsilon(n)$ random bits. On the other hand, Pena and Borodin [45] showed that any algorithm which is $(1/2 + \epsilon)$-competitive for any constant $\epsilon > 0$ must use at least $(1 - o(1)) \log \log n$ advice bits, and hence at least that amount of random bits.[1] The best known bounds on the number of advice bits needed to outperform deterministic algorithms for the problem are $O(\log n)$, using exponential time, and $O(n)$, using polynomial time [9, 20, 45]. Closing these (doubly-)exponential gaps between upper and lower bounds on the amount of randomness and nondeterminism needed to outperform deterministic online bipartite matching algorithms has remained a tantalizing open question.

Question 1.1. *How much randomness/advice is needed to achieve a $(1/2 + \Omega(1))$ competitive ratio for online bipartite matching?*

The Power of (Online) Rounding. A very general and widely-used algorithmic framework for the design of randomized online algorithms is the following two-step approach: first, design an online algorithm for a fractional relaxation of the problem, and then randomly round the fractional solution in an online fashion. This approach has been highly successful, playing a pivotal role in the resolution of many fundamental online problems, such as the k-server problem [5, 12, 13, 36], weighted paging [1, 6], generalized caching [1, 7], Metrical task systems [8, 11, 16], online set cover [3], online weighted steiner tree [41], online edge coloring [17, 48], and more. However, this design pattern has not been as helpful for the extensively-studied class of online bipartite matching

[1] All logarithms in this paper are to base 2.

problems. This is surprising, given the integrality of the bipartite matching polytope, implying that fractional bipartite matchings can be rounded without loss. More formally, a *lossless* rounding scheme is a polytime randomized algorithm that given a fractional bipartite matching $\vec{x}$ outputs a matching $\mathcal{M}$ that satisfies

$$\Pr[e \in \mathcal{M}] = x_e \qquad \forall e \in E. \tag{1}$$

For the bipartite matching polytope such lossless rounding exists, and moreover can satisfy strong and useful negative correlation properties [27]. Lossless rounding implies that the output matching's value is as good as the fractional matching, for any linear objective. Therefore, a similar *online* rounding scheme combined with known fractional algorithms [14, 23, 51] would result in improved/optimal randomized algorithms for many problems in this area. Put otherwise, the lack of such rounding schemes poses a major obstacle for the design of optimal randomized algorithms for edge-weighted online matching [22], online matching with two-sided arrivals [25], and adwords without the small bid assumption [33]. Unfortunately, as pointed out by [18, 19] (and elaborated in Section 1.2), such *online* lossless rounding of fractional bipartite matchings is in general *impossible*. Nevertheless, in this work, we ask what can be salvaged from this seemingly hopeless approach of (lossless) online rounding for bipartite matching problems.

1.1 Our Main Contributions

In this paper we initiate the study of lossless online rounding of fractional bipartite matchings. As discussed above, such rounding is generally impossible. However, in this paper, we argue (and show) that despite this general impossibility, lossless online rounding is possible for competitive fractional matchings which are designed *with rounding in mind*. Indeed, since every randomized algorithm induces a fractional matching which, by construction, is losslessly roundable, it is natural to search for roundable fractional matching algorithms. We identify a non-trivial family of such roundable algorithms. In particular, we design fractional algorithms which:

1. Are better than $1/2$-competitive.
2. Are losslessly roundable, using polynomially-many random bits.
3. Are near-losslessly roundable, using (optimal) $(1 + o(1)) \cdot \log \log n$ random bits.

Describing the fractional algorithms we consider requires some notation, which we now provide. (For formal definitions, we refer the reader to Section 2.) The input to the online matching problem is a bipartite graph, with the left and right sides referred to as offline and online nodes, respectively. Each offline node i has weight $w_i > 0$ (in the unweighted version $w_i = 1$ for all i). Initially, only the n offline nodes are known. Online nodes arrive sequentially. At times $t = 1, 2, \dots,$ online node t arrives. An online matching algorithm must assign for each edge $(i, t) \in E$, immediately and irrevocably, a fractional value $x_{i,t} \in [0, 1]$ (or $x_{i,t} \in \{0, 1\}$, for integral algorithms). These values should satisfy that each node v has *fractional degree* at most one, i.e., $\sum_{e \ni v} x_e \leq 1$. So, for example, $x_i^{(t)} := \sum_{t' < t} x_{i,t'}$, the fractional degree of offline node i before time t, satisfies $x_i^{(t)} \leq 1$. The goal is to maximize the weighted value of the matching, $\sum_{i,t} w_i \cdot x_{i,t}$. We focus on *two-choice* algorithms, which, for any online node t set $x_{i,t} > 0$ for at most two offline nodes i. Such two-choice algorithms were previously used to obtain breakthrough results breaking the barrier of $1/2$ for other online bipartite matching problems [22, 34]. On the other hand, prior work shows that two-choice are not, in general, losslessly roundable online [18, 19].

The main new ingredient in our work is the following set of (non-convex) constraints for fractional two-choice algorithms, which we motivate and provide intuition for in Section 1.2.

Definition 1.2. *A two-choice fractional online matching algorithm $\mathcal{A}$ is* sound *if for every online node t with $P_t := \{i \mid x_{i,t} > 0\}$, the fractional matching $\vec{x}$ of $\mathcal{A}$ satisfies*

$$\sum_{i \in P_t} x_{i,t} \leq 1 - \prod_{i \in P_t} x_i^{(t)}. \tag{2}$$

If Equation (2) is met at equality for each t, we say that $\mathcal{A}$ is also maximal.

Our main technical contribution is a proof that sound two-choice algorithms can be rounded losslessly online.

Theorem 1.3 (Lossless Rounding)**.** *Let $\mathcal{A}$ be a sound two-choice online fractional algorithm, and its output be $\vec{x}$. Then, there exists a randomized online algorithm whose output matching $\mathcal{M}$ matches each edge (i,t) with probability*

$$\Pr[(i,t) \in \mathcal{M}] = x_{i,t}.$$

Moreover, if $\mathcal{A}$ is maximal, then this randomized online algorithm is implementable in poly-time.

Given the above rounding scheme, we ask whether such lossless (or near-lossless) rounding can be implemented with less randomness. We show that this is indeed possible for certain two-choice algorithms, which we refer to as *k-level b-bit algorithms* (see Section 3.3 for definitions).

Theorem 1.4 (Low-Randomness Rounding)**.** *Let $\mathcal{A}$ be a k-level b-bit fractional algorithm, with output $\vec{x}$. Then, there exists a polytime randomized online algorithm using $(1+o(1)) \cdot \log\log n + b \cdot 2^{k+1}$ random bits, whose output matching $\mathcal{M}$ matches each edge (i,t) with probability*

$$\Pr[(i,t) \in \mathcal{M}] = x_{i,t} \pm o(1).$$

The above rounding schemes are, of course, worthless without good accompanying online fractional solutions. Fortunately, designing (deterministic) fractional online algorithms is in general a much easier task than designing randomized algorithms. However, adding the extra (non-convex) constraints of Definition 1.2 adds a new dimension of complexity, and requires somewhat non-standard analysis (more on this later). We show that even with these extra constraints, designing online fractional algorithms with good competitive ratios is possible.

Theorem 1.5 (Fractional Algorithms)**.** *There exist sound two-choice fractional algorithms that are:*

- 0.532-*competitive for online bipartite matching.*
- *k-level 2^{k-1}-bit, and $(0.532 - O(2^{-2^k}))$-competitive for online bipartite matching.*
- 2-*level* 2-*bit, and* $^{11}/_{21} \approx 0.524$-*competitive for online vertex-weighted bipartite matching.*

A Sharp Threshold for Randomness and Nondeterminism. Our main quantitative result, obtained by combining theorems 1.4 and 1.5, are randomized algorithms with a random seed of $(1+o(1))\log\log n$ bits which are 0.532-competitive and 0.524-competitive for the unweighted and vertex-weighted online bipartite matching problem, respectively. Combing these upper bounds with the $(1-o(1))\log\log n$ advice bits (and hence random bits) lower bound of [45], this provides a sharp *threshold* for the randomness needed to outperform deterministic algorithms, and likewise for the advice complexity of this problem. To conclude, we obtain the following sharp answer to Question 1.1 and its vertex-weighted generalization.

Theorem 1.6. *$(1\pm o(1))\log\log n$ random (advice) bits are both necessary and sufficient to achieve a competitive ratio of $(1/2+\Omega(1))$ for online vertex-weighted bipartite matching.*

1.2 Techniques

In this section we give a high-level intuition and overview of our techniques for lossless rounding two-choice algorithms. We first briefly recall the motivation for such lossless rounding.

Motivation and challenges of lossless online rounding. The impossibility of lossless online rounding was first observed by Devanur et al. [19]. (We refer the unfamiliar reader to an illustrative example in Appendix A.) Correspondingly, prior work applying rounding to online matchings [18, 25, 44, 48] studied *lossy* rounding schemes. Given the delicate constants involved in online (fractional) matching problems, for which optimal competitive ratios are in the range $[1/2, 1-1/e]$, this approach may cause the rounding to lose some of the poignancy of the fractional results. For example, the lossy rounding of Gamlath et al. [25] rounds a family of fractional algorithms for online matching in general graphs which include a 0.526-competitive algorithm [51], but only obtains a randomized $1/2+\epsilon$ competitive ratio, for some small constant $\epsilon>0$. More importantly, lossy rounding cannot hope to obtain the competitive ratios which are known to be achievable fractionally and conjectured to be achievable randomly for many problems in the area [14, 22, 22, 23, 25, 34].

In this work we deviate from the lossy rounding approach, and show how to round well-structured (competitive) fractional algorithms online, *losslessly.* To this end, we revisit the design of fractional algorithms with the explicit objective of losslessly rounding their output online.

What fractional solutions are roundable? Suppose we have managed to losslessly round a fractional matching $\vec{x}$ until time t. Consequently, each offline node is matched before time t with probability $x_i^{(t)}$. Suppose for now that these probabilities are independent. Then, since t can only be matched to a neighbor in $P_t := \{i \mid x_{i,t}>0\}$ if at least one node in P_t is free at this point, we find that for lossless rounding to be possible, we have

$$\sum_{i\in P_t} x_{i,t} = \Pr[t \text{ matched}] \leq 1-\Pr[\text{all } P_t \text{ matched before time } t] = 1-\prod_{i\in P_t} x_i^{(t)}.$$

Therefore, we have that $\sum_{i\in P_t} x_{i,t} \leq 1-\prod_{i\in P_t} x_i^{(t)}$ is a *necessary condition* for lossless rounding in this scenario, where all offline nodes are matched independently. We say that our fractional matchings are *sound* precisely since this inequality is a necessary condition for lossless rounding. For two-choice algorithms (i.e., with $|P_t|\leq 2$), it is not hard to match edges of t with marginal probabilities $x_{i,t}$, assuming $\vec{x}$ is sound and the matched statuses of offline nodes are independent. Rounding becomes a challenge when there are (possibly) correlations between these offline nodes.

Dealing with correlations. In general, the matched status of offline nodes may depend on each other in intricate ways, based on the rounding so far. Positive correlations between nodes in P_t may cause the probability of all P_t being matched before time t to be strictly greater than $\prod_{i\in P_t} x_i^{(t)}$. This would rule out lossless rounding of maximal sound algorithms, as it would imply the following.

$$\Pr[t \text{ matched}] = \sum_{i\in P_t} x_{i,t} = 1 - \prod_{i\in P_t} x_i^{(t)} > 1 - \Pr[\text{all } P_t \text{ matched before time } t].$$

However, *negative dependence* between the matched status of offline nodes in P_t would still be consistent with matching edges of t with marginal probabilities $x_{i,t}$ output by a sound algorithm. Indeed, such dependence (and lossless rounding up to time t) would guarantee the necessary condition for lossless online rounding, namely $\Pr[t \text{ matched}] \leq 1 - \Pr[\text{ all } P_t \text{ matched before time } t]$:

$$\Pr[t \text{ matched}] = \sum_{i\in P_t} x_{i,t} \leq 1 - \prod_{i\in P_t} x_i^{(t)} \leq 1 - \Pr[\text{all } P_t \text{ matched before time } t].$$

Accordingly, the crux of our rounding algorithm for sound two-choice algorithms is in designing a way of matching edges of t that simultaneously: (1) guarantees the marginal probabilities, and (2) preserves pairwise negative correlations between the matched status of different nodes. The second requirement is challenging to prove directly, and we end up proving significantly stronger forms of negative correlation between offline nodes to obtain our guarantees.

Our negative correlation properties. Let $F_{i,t}$ be an indicator for offline node i being free (unmatched by the rounding algorithm) by time t, and let $F_{I,t} := \bigwedge_{i\in I} F_{i,t}$ be an indicator for all of offline node-set I being free at time t. In addition to satisfying the target marginal probabilities, our algorithm also satisfies the following strong negative correlation property

$$\Pr[F_{i,t} \mid F_{K,t}] \leq \Pr[F_{i,t} \mid F_{J,t}] \qquad \forall t,\, \forall i \in [n],\, \forall J \subseteq K \subseteq [n] \setminus \{i\} \text{ s.t.} \Pr[F_{K,t}] > 0.$$

In words, the probability of node i to be free at time t decreases when we condition on a larger set of other offline nodes being free at that time. This strong negative correlation property, while natural, is not shared by prior online bipartite matching algorithms.[2] If the fractional matching x is also maximal, our rounding algorithm satisfies the stronger invariant that at any time t, offline nodes in every set I are either matched independently, or at least one node in I must be matched, i.e., $\Pr[F_{I,t}] \in \left\{0, \prod_{i\in I} \Pr[F_{i,t}]\right\}$. We believe that the strong negative correlation properties of our rounding algorithms (both strong enough to imply exponential tail bounds, similarly to known offline lossless rounding schemes [26]) are of independent interest, and might find uses beyond this work.

Low-randomness implementation. A surprising useful property of our lossless online rounding scheme is that for particularly well-structured fractional algorithms, which we refer to as k-level algorithms (see Section 3.3), the number of random variables of our rounding algorithm which any node's matched status depends on is constant. This property, combined with the theory of small-bias distributions (see Section 2) allows us to implement our rounding scheme for these algorithms with little randomness. This low-randomness rounding, applied to the competitive k-level algorithms which we design in Section 4, underlies our qualitative results of Theorem 1.6.

Remark 1.7 (Generalization to d-choice algorithms)**.** *Given the above discussion, it is reasonable to suspect that Condition* (2) *is sufficient for lossless online rounding of d-choice algorithms for $d > 2$. Surprisingly, in Appendix B, we show that this condition is* not *sufficient for lossless online rounding even when $d = 3$. Our work therefore suggests the following question.*

Question 1.8. *What is a sufficient condition for (good) fractional algorithms to be (efficiently) losslessly roundable online?*

[2] E.g., the RANKING algorithm of Karp et al. [35] is easily seen to induce *positively* correlated variables $\{F_{i,t}\}_i$.

1.3 Further Related Work

Following the seminal work of Karp et al. [35], extensions and generalizations of the online bipartite matching problem were widely studied. In 2005, Mehta et al. [39] introduced the adwords problem and drew attention to the connection between online matching and its extensions to Internet advertising, which sparked a flurry of research in this field. Furthermore, [39] gave an (optimal) $(1 - 1/e)$-competitive algorithm for the adwords problem under the small-bid assumption; Buchbinder et al. [14] obtained the same result via the online primal-dual method, and also presented an optimal fractional algorithm for the problem (without assumptions). A similar $1 - {}^1\!/e$ competitive ratio was obtained for the vertex-weighted bipartite matching problem [2]. For edge-weighted matching, a $1 - {}^1\!/e$ fractional algorithm was given by [23]. In a celebrated recent result, the barrier of ${}^1\!/2$ for randomized (integral) algorithms was broken by Fahrbach et al. [22]. A similar first super-deterministic randomized algorithm was recently achieved for adwords without the small-bids assumption [34]. For both problems, as well as for online matching in more general arrivals ([25, 29, 31, 51]), online rounding has the potential to improve the state of the art further, and possibly even yield optimal competitive ratios, if lossless rounding is achievable.

2 Preliminaries

In the online bipartite matching problem, the underlying input is a bipartite graph $G = (L, R, E)$. Initially, only the n offline nodes (nodes), $L = [n]$, are known, as well as the weight $w_i > 0$ of each offline node $i \in L$. At times $t = 1, 2, \ldots$, online node $t \in R$ arrives, together with its edges.

A fractional online matching algorithm, at each time t, must decide, immediately and irrevocably, what values $x_{i,t}$ to assign edges $(i, t) \in E$, while guaranteeing that the *fractional degree* of each vertex $v \in L \cup R$ satisfies $\sum_{e \ni v} x_e \leq 1$. We denote by $x_i^{(t)} := \sum_{t'<t} x_{i,t'}$ the fractional degree of vertex i before time t, and by $x_t := \sum_i x_{i,t}$ the fractional degree of vertex t. Thus, a fractional online algorithm guarantees that $x_t \leq 1$ and $x_i^{(t)} \leq 1$ for any time t and offline vertex i. Thus, such an algorithm maintains a feasible solution to the fractional (vertex-weighted) matching problem, whose linear programming relaxation is given (together with its dual) in Figure 1.

Primal		Dual	
maximize	$\sum_{(i,t)\in E} w_i \cdot x_{it}$	minimize	$\sum_{i\in L} y_i + \sum_{t\in R} y_t$
subject to:		subject to:	
$\forall i \in L$:	$\sum_t x_{i,t} \leq 1$	$\forall (i,t) \in E$:	$y_i + y_t \geq w_i$
$\forall t \in R$:	$\sum_i x_{i,t} \leq 1$	$\forall i \in L$:	$y_i \geq 0$
$\forall (i,t) \in E$:	$x_{i,t} \geq 0$	$\forall t \in R$:	$y_t \geq 0$

Figure 1: The fractional vertex-weighted bipartite matching LP and its dual

The analysis of our fractional algorithms uses these LPs and the well-established online primal-dual method [15]. The challenge in the analysis is dealing with the non-convex Constraint (2).

Bounded Independence. A useful notion we will make use of is (δ, k)-dependence, introduced by Naor and Naor [42], generalizing k-wise independence.

Definition 2.1 ([42]). *Binary random variables* $Y_1, Y_2, \ldots, Y_m$ *are* (δ, k)-dependent *if for any subset of* k *or fewer indices,* $I \subseteq [m]$, $|I| \leq k$,

$$\sum_{\vec{v}\in\{0,1\}^{|I|}} \left| \Pr\left[\bigwedge_{i\in I} (Y_i = v_i)\right] - 2^{-|I|} \right| \leq \delta.$$

A $(0,k)$-dependent distribution is *k-wise independent*, satisfying that any subset of k or fewer variables is independent. More generally, a joint distribution $\vec{Y}$ is (δ,k)-dependent if for any subset $I \subseteq [m]$ of k or fewer variables, the total variation distance between the distribution on the variables indexed by I and the uniform distribution on $|I|$ i.i.d bernoulli($1/2$) variables is at most δ. Consequently, such a distribution satisfies the following.

Lemma 2.2. *Let $\mathcal{U}$ be the uniform distribution and let $\mathcal{D}$ be a (δ,k)-dependent distribution over m binary variables $Y_1, Y_2, \ldots, Y_m$. Then, for any event A which is determined by k or fewer random variables in $Y_1, Y_2, \ldots, Y_m$,*

$$\Pr_{\vec{Y}\sim\mathcal{U}}[A] - \delta \leq \Pr_{\vec{Y}\sim\mathcal{D}}[A] \leq \Pr_{\vec{Y}\sim\mathcal{U}}[A] + \delta.$$

Proof. Let $I \subseteq [m]$ be a set of k or fewer indices such that $\{Y_i \mid i \in I\}$ determine A, and let $S \subseteq 2^{|I|}$ be such that $A = \bigcup_{\vec{v}\in S}\left[\vec{Y} = \vec{v}\right]$. Then, by triangle inequality and definition of (δ,k)-dependence,

$$\left|\Pr_{\vec{Y}\sim\mathcal{D}}[A] - \Pr_{\vec{Y}\sim\mathcal{U}}[A]\right| \leq \sum_{\vec{v}\in S}\left|\Pr\left[\bigwedge_{i\in I}(Y_i = v_i)\right] - 2^{-|I|}\right| \leq \sum_{\vec{v}\in\{0,1\}^{|I|}}\left|\Pr\left[\bigwedge_{i\in I}(Y_i = v_i)\right] - 2^{-|I|}\right| \leq \delta. \qquad \square$$

A useful property of (δ,k)-dependent distributions is that such distributions can be specified—and constructed in polynomial time—using a small random seed [42, 49]. For completeness, a proof of this lemma, following the construction of Naor and Naor [42], is given in Appendix E.

Lemma 2.3. *For any $\delta > 0$, a (δ,k)-dependent joint distribution on n binary variables can be constructed using $\log\log n + O(k + \log(\frac{1}{\delta}))$ random bits. Moreover, after polytime preprocessing, each random variable in this distribution can be sampled in $O(k \cdot \log n)$ time.*

3 Rounding, and an FKG-like Inequality

In this section we show how to round losslessly two-choice sound fractional solutions online, proving Theorem 1.3 and Theorem 1.4. In Section 3.1 and Section 3.2 we present the lossless rounding algorithm, and prove its properties. In Appendix C an analysis of a special case of our algorithm for *maximal* 2-choice sound algorithms, which, while less general, has the advantage of being simpler to describe, having stronger negative correlation properties, and allowing for a polytime implementation. In Section 3.3 we then prove that this algorithm can be implemented with $o(1/n^2)$ additive loss per edge using $(1+o(1))\log\log n$ bits of randomness.

3.1 The Algorithm and its Invariants

In this section we design our lossless rounding algorithm. We are given a 2-choice fractional algorithm that satisfies property (2) meaning that for any online node t, $\sum_{i\in P_t} x_{i,t} \leq 1 - \prod_{i\in P_t} x_i^{(t)}$. Let $F_{i,t}$ be the event that offline node i is free (unmatched in $\mathcal{M}$) by time t, and let $F_{I,t} := \bigwedge_{i\in I} F_{i,t}$ be the event that all nodes in $I \subseteq [n]$ are free by time t. Our online rounding algorithm maintains the following two invariants for any time t.

$$\Pr[(i,t) \in \mathcal{M}] = x_{i,t} \qquad \forall (i,t) \in E \tag{3}$$

$$\Pr[F_{i,t} \mid F_{K,t}] \leq \Pr[F_{i,t} \mid F_{J,t}] \qquad \forall t,\ \forall i \in [n],\ \forall J \subseteq K \subseteq [n] \setminus \{i\} \text{ s.t. } \Pr[F_{K,t}] > 0 \tag{4}$$

The first condition is precisely losslessness, while the second monotonicity property is precisely **log-submodularity** of the function $f(I) := \Pr[F_{I,t}]$,

$$\log \Pr[F_{K+i,t}] - \log \Pr[F_{K,t}] \leq \log \Pr[F_{J+i,t}] - \log \Pr[F_{J,t}].$$

Equivalently, this is a (reverse) **FKG-like lattice condition** [24],

$$\Pr[F_{A\cap B,t}] \cdot \Pr[F_{A\cup B,t}] \leq \Pr[F_{A,t}] \cdot \Pr[F_{B,t}].$$

Invariant (4) implies negative pairwise correlation between the $F_{i,t}$ variables, i.e., $\Pr[F_{i,t}, F_{j,t}] \leq \Pr[F_{i,t}] \cdot \Pr[F_{j,t}]$ for all $i \neq j$, and hence between these variables' complements, $M_{i,t} := 1 - F_{i,t}$, i.e., $\Pr[M_{i,t}, M_{j,t}] \leq \Pr[M_{i,t}] \cdot \Pr[M_{j,t}]$. Therefore, combining Condition (2) with Invariants (3) and (4) we obtain the following bound on the probability of any online node t being matched.

$$\Pr[t \text{ matched}] \overset{(3)}{=} \sum_{i\in P_t} x_{i,t} \overset{(2)}{\leq} 1 - \prod_{i\in P_t} x_i^{(t)} \overset{(3)}{=} 1 - \prod_{i\in P_t} \Pr[M_{i,t}] \overset{(4)}{=} 1 - \Pr\left[\bigwedge_{i\in P_t} M_{i,t}\right].$$

The conclusion of this chain of inequalities, whereby $\Pr[t \text{ matched}] \leq 1 - \Pr\left[\bigwedge_{i\in P_t} M_{i,t}\right] = 1 - \Pr[\text{all nodes in } P_t \text{ matched before time } t]$, is a trivial necessary condition for any randomized matching algorithm. We show that the conditions we impose on our fractional solution, together with the invariants we maintain, allow us to inductively maintain these properties, while outputting a randomized matching, online.

We next describe formally the algorithm. Assume without loss of generality that online node t increases two neighbors: $1, 2$. Let $x_1 := x_1^{(t)}, x_2 := x_2^{(t)}$ be their fractional values, and $\Delta x_1 := x_{1,t}, \Delta x_2 := x_{2,t}$ be their change. By the properties of the fractional agorithm we are guaranteed that we have that $\Delta x_1 + \Delta x_2 \leq 1 - x_1 x_2$. Our pseudocode is given in Algorithm 1.

Algorithm 1 Online Lossless Rounding

for arrival of online node t **do**
  if t has less than two neighbors **then**
    add two dummy neighbors i with $\Delta x_i = 0$ and $x_i = 1$ ▷ used to simplify notation
  let $P_t := \{1, 2\}$ be the two neighbors of t of highest $\Delta x_i := x_{i,t}$, and let $x_i := x_i^{(t)}$
  let $p_{12} := \Pr[F_{\{1,2\},t}]$ ▷ assuming p_{12} is known
  let a_1, a_2, b_1, b_2 be solutions to Program (Prob-Program) with input $\Delta x1, \Delta x_2, x_1, x_2, p_{12}$
  if $1, 2$ are both free **then**
    match t to 1 with probability a_1 and to 2 with probability a_2
  else if a single $i \in \{1, 2\}$ is free **then**
    match t to i with probability b_i

Probability-Setting Program($\Delta x1, \Delta x_2, x_1, x_2, p_{12}$): (Prob-Program)

$$a_1 + a_2 \leq 1 \tag{5}$$

$$a_i \geq 0 \qquad \forall i = 1, 2 \tag{6}$$

$$b_i \leq 1 \qquad \forall i = 1, 2 \tag{7}$$

$$b_i \geq a_i \qquad \forall i = 1, 2 \tag{8}$$

$$b_i \leq \frac{a_i}{1 - a_{3-i}} \qquad \forall i = 1, 2 \tag{9}$$

$$a_i \cdot p_{12} + b_i \cdot (1 - x_i - p_{12}) = \Delta x_i \qquad \forall i = 1, 2 \tag{10}$$

We first show that the algorithm's steps at time t are well-defined, provided our claimed invariants hold until this time.

Lemma 3.1. *Assuming invariants* (3) *and* (4) *hold before time* t*, then the algorithm's steps at time* t *are well-defined. In particular, Program* (Prob-Program) *is solvable (efficiently). Consequently,*

1. $a_1 + a_2 \leq 1$ *and* $a_i \geq 0$ *for all* $i = 1, 2$. *(Line 8 is well-defined)*
2. $b_i \in [a_i, 1] \subseteq [0, 1]$ *for all* $i = 1, 2$. *(Line 10 is well-defined)*

Proof. Properties 1 and 2 follow from constraints (5), (6), (7) and (8) of Program (Prob-Program). It remains to prove that this program is (efficiently) solvable, which we do using the following algorithm. Initially, we set $a_i, b_i \leftarrow \Delta x_i/(1-x_i)$ for both $i = 1, 2$. If $\sum_i \Delta x_i/(1-x_i) \leq 1$, we terminate, as this solution satisfies all the constraints of Program (Prob-Program), with the non-trivial constraints following from the fractional matching constraints implying $\Delta x_i/(1-x_i) \in [0, 1]$. Otherwise, for $i = 1, 2$, we decrease a_i and increase b_i while maintaining Equation (10), until $a_1 + a_2 = 1$ or $b_i = 1$.[3] We note that one of the two stopping conditions will occur. Indeed, if we set $a_i = 0$, then, since $a_{3-i} \leq \Delta x_{3-i}/(1-x_{3-i}) \leq 1$, we have that $a_1 + a_2 \leq 1$. We conclude that by the algorithm's termination, constraints (5) and (6) and (7) are satisfied. Moreover, by construction (of the algorithm), the equality constraint (10) is satisfied. From this, we obtain the following.

$$a_i \cdot \frac{p_{12}}{1-x_i} + b_i \cdot \frac{1-x_i-p_{12}}{1-x_i} = \frac{\Delta x_i}{1-x_i}. \tag{11}$$

Now, by invariants (3) and (4), we have that $p_{12} = Pr[F_{\{1,2\},t}] \leq Pr[F_{1,t}] \cdot Pr[F_{2,t}] = (1-x_1)(1-x_2)$. Therefore, $p_{12} \leq (1-x_i)$ and so Equation (11) implies that $\Delta x_i/(1-x_i)$ is a convex combination of a_i and b_i. Since we initialize $a_i, b_i \leftarrow \Delta x_i/(1-x_i)$, and decrease a_i while increasing b_i, we obtain $b_i \geq \Delta x_i/(1-x_i) \geq a_i$, implying Constraint (8). Finally, to prove Constraint (9), we show that if $\sum_i \Delta x_i/(1-x_i) > 1$, then $a_1 + a_2 = 1$, and so Constraint (9) follows from Constraint (7), since $a_i/(1-a_{3-i}) = 1$. Indeed, if $a_1 + a_2 > 1$ by the algorithm's termination, then we must have stopped both iterations of the loop decreasing a_i and increasing b_i after reaching $b_i = 1$. But then, we have

$$\begin{aligned}
\Delta x_1 + \Delta x_2 &= b_1 \cdot (1-x_1-p_{12}) + a_1 \cdot p_{12} + b_2 \cdot (1-x_2-p_{12}) + a_2 \cdot p_{12} \\
&= (1-x_1) + (1-x_2) + (a_1+a_2-2) \cdot p_{12} && b_1 = b_2 = 1 \\
&> (1-x_1) + (1-x_2) - p_{12} && a_1 + a_2 > 1 \\
&\geq (1-x_1) + (1-x_2) - (1-x_1)(1-x_2) && \text{inv. (3) and (4)} \\
&= 1 - x_1 \cdot x_2,
\end{aligned}$$

thus contradicting Condition (2), i.e., $\Delta x_1 + \Delta x_2 \leq 1 - x_1 \cdot x_2$. We conclude that this algorithm terminates with a feasible solution to (Prob-Program), and thus Algorithm 1 is well-defined. □

3.2 Lossless Rounding using Algorithm 1

So far, we have proven that assuming the claimed invariants—(3) and (4)—hold prior to time t, then Algorithm 1 is well defined. We now prove that if these invariants hold prior to time t, then they likewise hold prior to time $t+1$.

For our proof we will need the following simple corollary of Bayes' Law.

[3] This algorithm, which we state as a continuous algorithm, is trivial to discretize and implement in constant time.

Observation 3.2. *If* $(A, B) \perp (C, D)$*, (i.e.,* (A, B) *and* (C, D) *are independent), then*

$$\Pr[A, C \mid B, D] = \Pr[A \mid B] \cdot \Pr[C \mid D].$$

The special case of $\Pr[C] = 1$ *implies that if* $D \perp (A, B)$*, then* $\Pr[A \mid B, D] = \Pr[A \mid B]$.

Proof. By Bayes' Law, we have that

$$\Pr[A, C \mid B, D] = \frac{\Pr[A, B, C, D]}{\Pr[B, D]} = \frac{\Pr[A, B] \cdot \Pr[C, D]}{\Pr[B] \cdot \Pr[D]} = \Pr[A \mid B] \cdot \Pr[C \mid D]. \qquad \square$$

Lemma 3.3. *Algorithm 1 satisfies Invariants* (3) *and* (4).

Proof. First, we prove Invariant (3). Fix a time t. By construction (of the algorithm) we trivially have $\Pr[(i, t) \in \mathcal{M}] = 0$ for all i with $\Delta x_i = 0$. Now, let $P_t = \{1, 2\}$ be as in Line 4, and let $i \in P_t$. Then, by our choice of a_i, b_i, and Constraint (10), the probability i is matched to t is precisely

$$\begin{aligned} \Pr[(i, t) \in \mathcal{M}] &= \Pr[(i, t) \in \mathcal{M}, F_{P_t,t}] + \Pr[(i, t) \in \mathcal{M}, F_{i,t}, \overline{F_{3-i,t}}] \\ &= a_i \cdot p_{12} + b_i \cdot (p_1 - p_{12}) = \Delta x_i. \end{aligned}$$

We now turn to proving Invariant (4). We prove this invariant holds for all tuples (i, J, K, t), by induction on t. The invariant clearly holds for $t = 1$. Assume the invariant holds for time $t \geq 1$. We prove that this implies the same for time $t + 1$. For the inductive step, when wishing to prove Invariant (4) for the tuple $(i, J, K, t + 1)$, we may safely assume that both $\Pr[F_{K,t}] \neq 0$ and $\Pr[F_{K \cup \{i\},t}] \neq 0$ hold. Indeed, the converse would imply that $\Pr[F_{K \cup \{i\},t+1}] \leq \Pr[F_{K \cup \{i\},t}] = 0$, in which case Invariant (4) holds trivially for this tuple.

Let E_i denote the event that the algorithm does not match (i, t) at time t. We further denote by $C_1 \sim \mathrm{Ber}(a_1)$ and $C_2 \sim \mathrm{Ber}(a_2)$ the Bernoulli random variables corresponding to the probability of matching t to 1 and 2, respectively, in Line 8, if both 1 and 2 are free at time t. We can imagine our algorithm tosses these (correlated) coins regardless of the event $F_{\{1,2\},t}$, and only inspects these variables if the event $F_{\{1,2\},t}$ occurs. We note that the random variables C_1 and C_2 are independent of all events determined by random choices made by the algorithm until time t.

With this notation and these observations at hand, we now turn to proving the desired invariant holds for the tuple $(i, J, K, t + 1)$. There are five cases to consider, based on the inclusions between $P_t = \{1, 2\}$ and $K \cup \{i\}$, where if $i \in \{1, 2\}$, we assume without loss of generality that $i = 1$.

Case 1: $\{1, 2\} \cap (K \cup \{i\}) = \emptyset$. In this case $F_{I,t+1} \equiv F_{I,t}$ for all $I \subseteq K \cup \{i\}$, and so the invariant follows trivially from the inductive hypothesis.

Case 2: $\{1, 2\} \subseteq K$: By the inductive hypothesis, and independence of (C_1, C_2) from $F_{i,t}$, we obtain the desired inequality for the tuple $(i, J, K, t + 1)$.

$$\Pr[F_{i,t+1} \mid F_{J,t+1}] = \Pr[F_{i,t} \mid F_{J,t}] \geq \Pr[F_{i,t} \mid F_{K,t}] = \Pr[F_{i,t+1} \mid F_{K,t+1}].$$

Here, the equalities follow from $F_{i,t} \equiv F_{i,t+1}$ and $F_{i,t} \perp (C_1, C_2)$ together with Observation 3.2, while the inequality follows from the inductive hypothesis.

Case 3: $\{1, 2\} \cap K = \emptyset, i = 1$. For this case we rely on the probability of 1 not being matched decreasing when we condition on a larger set of offline nodes being free, as in the following inequality.

$$\Pr[E_1 \mid F_{K \cup \{1\},t}] \leq \Pr[E_1 \mid F_{J \cup \{1\},t}]. \tag{12}$$

Indeed, subtracting $\Pr[E_1 \mid F_{J\cup\{1\},t}]$ from both sides, expanding both terms using the law of total probability, we get

$$
\begin{aligned}
&\Pr[E_1 \mid F_{K\cup\{1\},t}] - \Pr[E_1 \mid F_{J\cup\{1\},t}] \\
=&\Pr[E_1 \mid F_{\{1,2\},t}] \cdot \Big(\Pr[F_{2,t} \mid F_{K\cup\{1\},t}] - \Pr[F_{2,t} \mid F_{J\cup\{1\},t}]\Big) \\
+&\Pr[E_1 \mid F_{1,t}, \overline{F_{2,t}}] \cdot \Big(\Pr[\overline{F_{2,t}} \mid F_{K\cup\{1\},t}] - \Pr[\overline{F_{2,t}} \mid F_{J\cup\{1\},t}]\Big) \\
=&((1-a_1)-(1-b_1)) \cdot (\Pr[F_{2,t} \mid F_{K\cup\{1\},t}] - \Pr[F_{2,t} \mid F_{J\cup\{1\},t}]) \ \le \ 0.
\end{aligned} \tag{13}
$$

Here, the second equality follows from $\Pr[E_1 \mid F_{\{1,2\},t}] = 1 - a_1$ and $\Pr[E_1 \mid F_{1,t}, \overline{F_{2,t}}] = 1 - b_1$ by definition. For any event A, $\Pr[F_{2,t} \mid A] + \Pr[\overline{F_{2,t}} \mid A] = 1$, implying

$$\Pr[F_{2,t} \mid F_{K\cup\{1\},t}] - \Pr[F_{2,t} \mid F_{J\cup\{1\},t}] = -(\Pr[\overline{F_{2,t}} \mid F_{K\cup\{1\},t}] - \Pr[\overline{F_{2,t}} \mid F_{J\cup\{1\},t}]).$$

Finally, Inequality (13) follows from Constraint (8) implying that $(1-a_1)-(1-b_1) \ge 0$, and by the inductive hypothesis together with the assumption that $\Pr[F_{K\cup\{1\},t}] \neq 0$ implying that $\Pr[F_{2,t} \mid F_{K\cup\{1\},t}] - \Pr[F_{2,t} \mid F_{J\cup\{1\},t}] \le 0$. We conclude that Equation (12) holds.

The desired inequality of Invariant (4) for the tuple $(i, J, K, t{+}1)$ then follows from Equation (12), the inductive hypothesis and the assumption that $\Pr[F_{K,t}] \neq 0$, implying

$$
\begin{aligned}
\Pr[F_{1,t+1} \mid F_{K,t+1}] &= \Pr[F_{1,t+1} \mid F_{K,t}] \\
&= \Pr[E_1 \mid F_{K\cup\{1\},t}] \cdot \Pr[F_{1,t} \mid F_{K,t}] \\
&\le \Pr[E_1 \mid F_{J\cup\{1\},t}] \cdot \Pr[F_{1,t} \mid F_{J,t}] && \text{I.H.} + (12) \\
&= \Pr[E_1, F_{1,t} \mid F_{J,t}] \\
&= \Pr[F_{1,t+1} \mid F_{J,t+1}].
\end{aligned}
$$

Case 4: $\{1,2\} \cap (K \setminus J) = \{2\}$, and $i = 1$. Independence of (C_1, C_2) from $(F_{1,t}, F_{K,t})$, and the inductive hypothesis yield the desired inequality for the tuple $(i, J, K, t+1)$, as follows.

$$
\begin{aligned}
\Pr[F_{1,t+1} \mid F_{K,t+1}] &= \Pr[\overline{C_1}, F_{1,t} \mid F_{K,t}, \overline{C_2}] \\
&= \Pr[\overline{C_1} \mid \overline{C_2}] \cdot \Pr[F_{1,t} \mid F_{K,t}] && (C_1, C_2) \perp (F_{1,t}, F_{K,t}) + 3.2 \\
&= \left(1 - \frac{a_1}{1-a_2}\right) \cdot \Pr[F_{1,t} \mid F_{K,t}] \\
&\le (1-b_1) \cdot \Pr[F_{1,t} \mid F_{K,t}] && (9) \\
&\le (1-b_1) \cdot \Pr[F_{1,t} \mid F_{J,t}] && \text{I.H.} \\
&\le \Pr[E_1 \mid F_{1,t}] \cdot \Pr[F_{1,t} \mid F_{J,t}] && a_1 \le b_1 \\
&= \Pr[F_{1,t+1} \mid F_{J,t}] \\
&= \Pr[F_{1,t+1} \mid F_{J,t+1}],
\end{aligned}
$$

where the last inequality relied on Constraint (8), whereby $a_1 \le b_1$, implying that

$$\Pr[E_1 \mid F_{1,t}] = (1-a_1) \cdot \Pr[F_{2,t} \mid F_{1,t}] + (1-b_1) \cdot \Pr[\overline{F_{2,t}} \mid F_{1,t}] \ge (1-b_1).$$

Case 5: $\{1,2\} \cap J = \{2\}$, and $i = 1$. Independence of (C_1, C_2) from $(F_{1,t}, F_{J,t}, F_{K,t})$, together with the inductive hypothesis, proves the desired inequality for the tuple $(i, J, K, t+1)$.

$$\begin{aligned}
\Pr[F_{1,t+1} \mid F_{K,t+1}] &= \Pr[\overline{C_1}, F_{1,t} \mid \overline{C_2}, F_{K,t}] \\
&= \Pr[\overline{C_1} \mid \overline{C_2}] \cdot \Pr[F_{1,t} \mid F_{K,t}] && (C_1, C_2) \perp (F_{1,t}, F_{K,t}) + 3.2 \\
&\leq \Pr[\overline{C_1} \mid \overline{C_2}] \cdot \Pr[F_{1,t} \mid F_{J,t}] && \text{I.H.} \\
&= \Pr[\overline{C_1}, F_{1,t} \mid \overline{C_2}, F_{J,t}] && (C_1, C_2) \perp (F_{1,t}, F_{J,t}) + 3.2 \\
&= \Pr[F_{1,t+1} \mid F_{J,t+1}].
\end{aligned}$$

□

Combining Lemma 3.1 and Lemma 3.3, we find that Algorithm 1 is well-defined throughout its execution. Moreover, we find that each edge is matched with the appropriate marginal probability prescribed by the fractional solution. In other words, we obtain the following.

Theorem 3.4. *Algorithm 1, when run on a fractional matching $\vec{x}$ satisfying Condition* (2)*, outputs a random matching $\mathcal{M}$ such that*

$$\Pr[(i,t) \in \mathcal{M}] = x_{i,t} \qquad \forall (i,t) \in E.$$

3.3 Small Random Seed for k-level Algorithms

The randomized algorithms derived from Algorithm 1 require (at least) polynomially-large random seeds. In this section we show that this is not really necessary, and essentially the same competitive ratio can be achieved using only a doubly-logarithmic random seed.

The need for a large random seed of our algorithms of the previous section is due to two reasons. The first one is because of precision issues: some of the probabilities in this algorithm can be arbitrarily small, and so these require arbitrarily-large random seeds. We overcome this first issue by explicitly restricting our attention to algorithms requiring only b bits of randomness to determine the random choices of Algorithm 4, as follows.

Definition 3.5. *A fractional algorithm $\mathcal{A}$ is b-*bit precise *if for each online node t with $P_t = \{1,2\} = \{i \mid x_{i,t} > 0\}$, the fractional matching $\vec{x}$ output by $\mathcal{A}$ satisfies*

$$\left\{ \frac{x_{i,t}}{1 - x_i^{(t)}}, \frac{1 - x_i^{(t)} - x_{i,t}}{(1-x_1)(1-x_2)} \right\} \in \left\{ \frac{a}{2^b} \;\middle|\; a \in \{0, 1, \ldots, 2^b\} \right\}.$$

The second, more fundamental reason, for the large random seed is our (implicit) requirement of complete independence between the random choices during each time step. For n random variables—one per arrival—this trivially requires at least n random bits. As we show, a significant saving over this amount of randomness can be obtained by considering small-bias distributions. For this, we will further restrict our attention to the following kind of two-choice algorithms.

Definition 3.6 (k-level Algorithm)**.** *A k-level algorithm has some $k+2$ possible values, denoted by $0 = z_0 < z_1 < \ldots < z_k < z_{k+1} = 1$, and maintains the invariant that each offline node has fractional degree equal to one of these z_i. At each step of the algorithm the fractions $x_1^{(t)}, x_2^{(t)}$ of at most two offline nodes $\{1,2\}$ are increased to $x_1^{(t+1)}, x_2^{(t+1)}$, with the following options:*

- **deterministic step:** *$x_1^{(t)}$ is increased to 1. ($x_2^{(t+1)} \leftarrow x_2^{(t)}$.)*
- **random step:** *$x_1^{(t)}, x_2^{(t)}$ are increased to $x_1^{(t+1)}, x_2^{(t+1)} > \max\{x_1^{(t)}, x_2^{(t)}\}$ (strict inequality).*

- **shift step:** $x_1^{(t)} = 0$ *and* $x_2^{(t)} \in (0,1)$ *are increased to* $x_1^{(t+1)} \leftarrow x_2^{(t)}, x_2^{(t+1)} \leftarrow 1$.

We show that when rounding such fractional algorithms using Algorithm 1, the following holds: for each edge (i,t), the event that (i,t) is matched in the random matching $\mathcal{M}$ output by Algorithm 4 depends on a bounded number of random choices of this algorithm. To this end, we denote by $A_{i,t} \sim Bernoulli(a_i^t)$ the random variable corresponding to the random choice in Line 6 of Algorithm 4 and by $B_{i,t} \sim Bernoulli(b_i^t)$ the random variables of Line 11, where a_i^t, b_i^t are the solution to Program (Prob-Program) used by the algorithm at time t. We prove the following.

Lemma 3.7. *The event* $[(i,t) \in \mathcal{M}]$ *is determined by at most* 2^{k+2} *random variables* $\{A_{i',t'}, B_{i',t'}\}_{i',t'}$.

Proof. We say an offline node i has level ℓ before time t if $x_i^{(t)} = z_\ell$. We prove by induction on all times t and on the level $\ell \leq k$ of node i before time t that $F_{i,t}$ is determined by at most $2^{\ell+1} - 2$ random variables. From this we find that $\mathbb{1}[(i,t) \in \mathcal{M}] = F_{i,t} - F_{i,t+1}$ is determined by at most $2^{k+1} - 2 + 2^{k+1} - 2 \leq 2^{k+2}$ random variables the set in $S := \{A_{i',t'}, B_{i',t'}\}_{i',t'}$.

For nodes at level 0, we have that $F_{i,t} \equiv 1$. Consequently, since at time $t = 1$ all offline nodes i are at level $\ell = 0$, the variables $F_{i,t}$ are determined by $0 = 2^0 - 1$ random variables. Now, consider a time t where the level of i increases, depending on what kind of step caused the increase to level ℓ. If this increase is due to a <u>deterministic step</u> (i.e., $\ell = k$), then $F_{i,t+1} \equiv 0$ is deterministic, by Invariant (3), and so this variable depends on $0 \leq 2^{k+1} - 2$ random variables in S. Suppose next that the level increase of i is due to a <u>random step</u>, with j the second neighbor of t whose level increases at time t. Then we have that $F_{i,t+1} = F_{i,t} \cdot \left(F_{j,t} \cdot \overline{A_{i,t}} + \overline{F_{j,t}} \cdot \overline{B_{i,t}}\right)$. Consequently, since both i and j had level strictly lower than the new level ℓ of i, we have by the inductive hypothesis that $F_{i,t+1}$ is determined by $2 + 2 \cdot (2^\ell - 2) = 2^{\ell+1} - 2$ random variables in S.

Finally, if the level of i increased to level ℓ due to a <u>shift step</u>, then if $\ell = k$, as argued before, $F_{i,t+1} \equiv 0$, and therefore $F_{i,t+1}$ is a function of zero variables. Otherwise, the level of the other node j whose level increased was ℓ before, while the previous level of i was zero. Consequently, by Invariant (3), we have that t is matched with probability one. That is, $(F_{i,t} - F_{i,t+1}) + (F_{j,t} - F_{j,t+1}) = 1$. But, similarly, by Invariant (3) we have that $F_{j,t+1} \equiv 0$, and since i was at level zero before time t, we have that $F_{i,t} \equiv 1$. Putting the above together, we find that $F_{i,t+1} = F_{j,t}$. Consequently, by the inductive hypothesis, since j previously had level ℓ, we have that $F_{i,t+1} = F_{j,t}$ is the function of at most $2^{\ell+1} - 2$ random variables in S. □

We can now rely on our analysis for Algorithm 4 using independent random variables to analyze the same algorithm when using $(\delta, b \cdot 2^{k+1})$-dependent binary variables to sample variables $A_{i,t}, B_{i,t}$. In particular, we can show that such a random seed yields an essentially lossless rounding.

Theorem 3.8. *Let* $\mathcal{M}$ *be the random matching output by Algorithm 4 when rounding a b-bit precision restricted k-level algorithm, using a distribution* $\mathcal{D}$ *over* $(\delta, b \cdot 2^{k+2})$*-dependent binary variables for the random choices. Then,*

$$\Pr_{\mathcal{D}}[(i,t) \in \mathcal{M}] = x_{i,t} \pm \delta \qquad \forall (i,t) \in E.$$

Proof. By definition of b-bit precisions, the probabilities of Algorithm 4 can be specified using b binary variables. On the other hand, by Lemma 3.7, each event $[(i,t) \in \mathcal{M}]$ is determined by 2^{k+2} random variables, or $b \cdot 2^{k+2}$ random binary variables. Now, if we denote by $\mathcal{U}$ the uniform distribution, then by Theorem 3.4 and Lemma 2.2 we obtain the desired result,

$$\Pr_{\mathcal{D}}[(i,t) \in \mathcal{M}] = \Pr_{\mathcal{U}}[(i,t) \in \mathcal{M}] \pm \delta = x_{i,t} \pm \delta.$$

□

Since there are n offline nodes and each node can change levels at most k times, the total number of random variable $A_{i,t}, B_{i,t}$ is bounded by $O(nk)$. Hence, by Lemma 2.3 and the above lemma, all k-level algorithms can be rounded with loss $\delta = \frac{1}{\log\log n} = o(1)$, using $(1 + o(1)) \log\log n + 2^{k+2}b$ random bits, if we take. This concludes Theorem 1.4.

We note that using standard k-wise independence, one can round a k-level b-bit algorithm without *any* loss (even $o(1)$) using $O(2^{k+2}b \cdot \log(nk))$ bits of randomness.

Finally, we note that an efficient implementation of Algorithm 1 with perfect independence implies an implementation with similar running time, only slowed down by the time to sample from a $(\delta, b \cdot 2^{k+2})$-dependent distributions. So, for example, Lemma C.5 yields polytime implementations of Algorithm 1 when applied to the k-level fractional matching algorithms we design in the sequel.

4 Online Roundable Fractional Matching Algorithms

In this section we prove Theorem 1.5, by designing two-choice maximally-independent fractional algorithms with high competitive ratios.

4.1 The Bounded Water-Level Algorithm

In this section we design a two-choice maximally-independent fractional algorithm for unweighted matching. The algorithm simply picks the two offline vertices with smallest fractional degree. It then applies "water level" on these two vertices (i.e., it raises the fractional degree of the neighbors of lowest degree among the pair) until it is maximally independent (i.e., until (2) holds with equality). The formal description appears as Algorithm 2.

Algorithm 2 Restricted Water Level

initially, set $\vec{x} \leftarrow \vec{0}$.
for arrival of online node t **do**
  if t has less than two neighbors **then**
    add two dummy neighbors i with $x_i^{(t)} = 1$ ▷ used to simplify notation
  let $0 \leq x_1^{(t)} \leq x_2^{(t)} \leq \cdots \leq x_k^{(t)} \leq 1$ be the fractional degrees of neighbors of t
  Let $x_f = \frac{x_1^{(t)} + x_2^{(t)} + 1 - x_1^{(t)} \cdot x_2^{(t)}}{2}$. Set $x_{1,t} \leftarrow x_f - x_1^{(t)}$, $x_{2,t} \leftarrow x_f - x_2^{(t)}$

First, we note that Algorithm 2 is well-defined, as it outputs a feasible fractional matching.

Observation 4.1. *Algorithm 2 outputs a feasible fractional matching $\vec{x}$, with $x_{i,t} = 0$ for all dummy nodes i and online nodes t.*

Proof. We show by induction on t that $\vec{x}$ satisfies the fractional matching constraints for all nodes. We thus assume that for all offline nodes $x_i^{(t)} \in [0, 1]$. First,

$$x_2^{(t)} \leq x_2^{(t)} + \frac{(1 - x_2^{(t)}) \cdot (1 + x_1^{(t)})}{2} = \frac{x_1^{(t)} + x_2^{(t)} + 1 - x_1^{(t)} \cdot x_2^{(t)}}{2} = 1 - \frac{(1 - x_1^{(t)})(1 - x_2^{(t)})}{2} \leq 1,$$

meaning that $x_1^{(t)} \leq x_2^{(t)} \leq x_f \leq 1$. This proves that for all i, $x_i^{(t)} \leq x_i^{(t+1)} \leq 1$ (and $x_{i,t} \geq 0$). In particular, if $x_2^{(t)} = 1$ then $x_f = 1$ and so we do not increase dummy vertices. Finally,

$$x_{1,t} + x_{2,t} = 2x_f - (x_1^{(t)} + x_2^{(t)}) = 1 - x_1^{(t)} \cdot x_2^{(t)} \leq 1.$$ □

We now turn to analyzing the competitive ratio of this algorithm. We prove the following:

Lemma 4.2. *Let $g : [0,1] \to [0,1]$ be a twice differentiable function that is monotone increasing, convex and bijective in $[0,1]$ (and so, in particular, satisfy $g(0) = 0$ and $g(1) = 1$). Then, Algorithm 2 is α_g-competitive, where*

$$\alpha_g := \min_{x\in[0,1]} \frac{1-x^2}{1-3g(x)+2g\left(x+\frac{1-x^2}{2}\right)}. \tag{14}$$

We first prove the following Claim.

Claim 4.3. *Let $g : [0,1] \to [0,1]$ be a twice differentiable function that is monotone increasing, convex and bijective in $[0,1]$ Then, for any $x \in [0,1]$, we have*

$$g'(x) \le g'(1) \le \frac{1}{\alpha_g}.$$

Proof. The first inequality follows from f being convex and twice differentiable. For the second inequality, we note that the RHS can be written as

$$\begin{aligned}\alpha_g = \max_{x\in[0,1]} \frac{1-3g(x)+2g\left(x+\frac{1-x^2}{2}\right)}{1-x^2} &= \max_{x\in[0,1]} \left(\frac{1-g(x)}{1-x^2} + \frac{g\left(x+\frac{1-x^2}{2}\right)-g(x)}{\frac{1-x^2}{2}}\right) \\ &\ge \max_{x\in[0,1]} \frac{g\left(x+\frac{1-x^2}{2}\right)-g(x)}{\frac{1-x^2}{2}} \\ &\ge g'(1),\end{aligned}$$

where the first inequality follows follows from $1-g(x) \ge 0$ for $x \in [0,1]$ and so $\frac{1-g(x)}{1-x^2} \ge 0$, while the second inequality follows by considering $x \to 1$.[4] □

We next prove that Algorithm 2 is α_g-competitive.

Proof. The proof relies on dual fitting. Let g be a function that satisfies the conditions of the lemma. We use the function $g(\cdot)$ to assign dual values to offline node with fractional degree x_i. When an online node t arrives, we denote by $0 \le x_1 \le x_2 \le \cdots \le x_k \le 1$ the fractional degrees of its neighbors (including dummy neighbors). Algorithm 2 increases the fractional degree of neighbors $i = \{1,2\}$ of t to $x_f := \frac{x_1+x_2+1-x_1x_2}{2} = \hat{x} + \frac{1-x_1x_2}{2}$, where $\hat{x} := \frac{x_1+x_2}{2}$. (Note that this is indeed an increase, as by Observation 4.1, for $i = 1,2$, we have $x_f - x_i = x_{i,t} \ge 0$, and so $x_f \ge x_i$.) We set the dual of the online node at time y_t to $1-g(x_2)$, while maintaining the invariant that each offline node i with fractional degree x_i has dual value $y_i = g(x_i)$. This satisfies the dual constraint for each edges (i,t), due to the monotonicity of y, implying $1-g(x_2)+g(x_i) \ge 1$ for all $x_i \ge x_2$, and due to the final fractional degree of 1 and 2 satisfying $x_f \ge x_2$.

[4] For the reader uncomfortable thinking of $\lim_{x\to1} \frac{g\left(x+\frac{1-x^2}{2}\right)-g(x)}{\frac{1-x^2}{2}}$ as being of the form $\lim_{x\to1}\lim_{h\to0} \frac{g(x+h)-g(x)}{h}$, since $h = (1-x^2)/2$ depends on x here, consider applying L'hôpital's rule, which yields

$$\lim_{x\to1} \frac{g\left(x+\frac{1-x^2}{2}\right)-g(x)}{\frac{1-x^2}{2}} = \lim_{x\to1} \frac{y'\left(x+\frac{1-x^2}{2}\right)\cdot(1-x)-g'(x)}{-x} = g'(1) + \frac{g'(1)-g'(1)}{-1} = g'(1).$$

We show that the primal gain is at least α_g times the dual cost, which implies the lemma, by weak duality. Let $\hat{x}_f = \hat{x} + \frac{1-\hat{x}^2}{2}$. By the AM-GM inequality we have:

$$\hat{x}_f = \hat{x} + \frac{1-\hat{x}^2}{2} \le \hat{x} + \frac{1-x_1x_2}{2} = x_f.$$

We now show that the dual and primal changes satisfy $\Delta D - \frac{1}{\alpha_g}\cdot \Delta P \le 0$. Indeed,

$$\begin{aligned}
\Delta D - \frac{1}{\alpha_g}\cdot \Delta P &= 1 - g(x_2) + (g(x_f) - g(x_1)) + (g(x_f) - g(x_2)) - \frac{1}{\alpha_g}\cdot 2(x_f - \hat{x}) \\
&\le 1 - g(\hat{x}) + 2g(x_f) - 2g(\hat{x}) - \frac{1}{\alpha_g}\cdot 2(x_f - \hat{x}) \qquad (15)\\
&\le 1 - g(\hat{x}) + 2g(\hat{x}_f) - 2g(\hat{x}) - \frac{1}{\alpha_g}\cdot 2(\hat{x}_f - \hat{x}) \qquad (16)\\
&\le \max_{x\in[0,1]} \left\{1 - 3g(x) + 2g\left(x + \frac{1-x^2}{2}\right) - \frac{1}{\alpha_g}\cdot(1-x^2)\right\} = 0.
\end{aligned}$$

Inequality (15) follows by convexity of y implying $2g(\hat{x}) \le g(x_1) + g(x_2)$, and $x_2 \ge \hat{x}$ together with y being monotone increasing. Inequality (16) follows from $\hat{x}_f \le x_f$ and the function $g(x) - \frac{1}{\alpha_g}\cdot x$ being monotone decreasing in x, as $g'(x) - \frac{1}{\alpha_g} \le 0$ for all $x \in [0,1]$, by Claim 4.3. The final equality follows by the definition of α_g. □

Finally, we prove the upper and lower bounds on the competitive ratio of the algorithm.

Theorem 4.4. *Algorithm 2 is α-competitive, where α is ≈ 0.532 and at most ≈ 0.536.*

Proof. For the lower bound we note that the function $g(x) := \frac{a^x-1}{a-1}$ for $a = 1.6$ satisfies the conditions of Lemma 4.2, and achieves a value of $\alpha_g \approx 0.532$. For the upper bound we design a bad example that shows that the competitive ratio of the algorithm is at most $\sum_{i\ge 0}\frac{1}{3}\cdot\left(\frac{2}{3}\right)^i\cdot\left(1 - 2^{-2^i+1}\right) \approx 0.536$.

The bad example consists of a bipartite graph with $n = 3^k$ nodes on either side, with a perfect matching. The online nodes arrive in rounds, as follows. At the beginning of round $i < k$, a subset of the offline nodes is active, and they all have the same fractional degree. The online nodes of a round each have three distinct neighbors among the active offline nodes. In every such three-tuple of offline nodes (neighboring a common online node in round i), one node is not matched at all. This node is chosen to be de-activated. A simple proof by induction shows that the number of active offline nodes in round $i = 0, 1, 2, \dots, k-1$ is $\left(\frac{2}{3}\right)^i \cdot n$, while the fractional degree of active nodes in round i is $1 - 2^{-2^i+1}$. Therefore, the nodes which are de-activated in round i only accrue a gain of $1 - 2^{-2^i+1}$. As these de-activated nodes in round i are a third of the active nodes in this round, we find that the total gain of the algorithm from nodes de-activaed in round i is $\frac{1}{3}\cdot\left(\frac{2}{3}\right)^i n\cdot\left(1 - 2^{-2^i+1}\right)$. Finally, in the last round, each of the $\left(\frac{2}{3}\right)^k\cdot n = 2^k = o(n)$ active nodes has one distinct online neighbor, and so each of these offline nodes gets a gain of one. The nodes of this last round guarantee the existence of a perfect matching in G, consisting of the edges of the last round, together with an edge between every online node t and its offline neighbor de-activated in the round t arrived in. We conclude that the algorithm's competitive ratio is at most

$$\inf_k \sum_{i=0}^{k} \frac{1}{3}\cdot\left(\frac{2}{3}\right)^i\cdot(1 - 2^{-2^i+1}) + \frac{2^k}{3^k} \approx 0.536$$

□

4.2 The k-Level Unweighted Algorithm

In this section we design a fractional k-level algorithm for the unweighted matching problem. The algorithm uses levels $z_0 = 0 < z_1 < z_2 < \ldots < z_k < 1$, where $z_i = z_{i-1} + \frac{1-z_{i-1}^2}{2}$. Solving the recursion yields $z_i := 1 - 2^{-2^i+1}$.[5] The algorithm is as follows:

Algorithm 3 The k-Level Algorithm

- initially, set $\vec{x} \leftarrow \vec{0}$
- **for** arrival of online node t **do**
  - **if** t has less than two neighbors **then**
    - add two dummy neighbors i with $x_i^{(t)} = 1$ ▷ used to simplify notation
  - let $0 \leq x_1^{(t)} \leq x_2^{(t)} \leq \cdots \leq x_k^{(t)} \leq 1$ be the fractional degrees of neighbors of t
  - **if** $x_1^{(t)} < x_2^{(t)}$ **or** $x_1^{(t)} = z_k$ **then**
    - set $x_1^{(t+1)} \leftarrow 1$ ▷ $x_{1,t} \leftarrow 1 - x_1^{(t)}$
  - **else if** $x_1^{(t)} = x_2^{(t)} = z_i < z_k$ **then**
    - set $x_1^{(t+1)} = x_2^{(t+1)} \leftarrow z_{i+1}$ ▷ $x_{1,t} = x_{2,t} \leftarrow z_{i+1} - z_i$

The following observation shows that Algorithm 3 indeed satisfies Definitions 3.5 and 3.6, and so can be rounded with less randomness.

Observation 4.5. *Algorithm 3 is a k-level independent maximal and 2^{k-1}-bit precise.*

Proof. It is easy to verify that the steps of the algorithm satisfy the k-level requirements in Definition 3.6. Also, since $z_i = z_{i-1} + \frac{1-z_{i-1}^2}{2}$ it is easy to see that the algorithm is maximal independent. Finally, we need to prove that the algorithm is b-bit precise (satisfies Definition 3.5). We should prove that at any time t:

$$\left\{ \frac{x_{i,t}}{1-x_i^{(t)}}, \frac{1-x_i^{(t)}-x_{i,t}}{(1-x_1)(1-x_2)} \right\} \in \left\{ \frac{a}{2^b} \mid a \in \{0,1,\ldots,2^b\} \right\}.$$

As each time t, the first term is either 1 or of the form: $\frac{z_i - z_{i-1}}{1-z_{i-1}} = \frac{1-z_{i-1}^2}{2(1-z_{i-1})} = \frac{1+z_{i-1}}{2} = 1-2^{-2^{i-1}}$, where $i \leq k$. The second term is always of the form $\frac{1-z_i}{(1-z_{i-1})^2} = \frac{1}{2}$. Therefore the algorithm is b-bit precise for $b = 2^{k-1}$. □

4.2.1 Warm-up: Analysis of the 2-level Algorithm

As a warm-up, we analyze the algorithm when $k = 2$. In this case the algorithm uses only two levels: $z_1 = \frac{1}{2}, z_2 = \frac{7}{8}$. The algorithm has the following 4 cases:

- $x_1^{(t)} = x_2^{(t)} = 0$: set $x_1^{(t+1)} = x_2^{(t+1)} \leftarrow \frac{1}{2}$.
- $x_1^{(t)} = x_2^{(t)} = \frac{1}{2}$: set $x_1^{(t+1)} = x_2^{(t+1)} \leftarrow \frac{7}{8}$.
- $x_1^{(t)} = x_2^{(t)} = \frac{7}{8}$: set $x_1^{(t+1)} \leftarrow 1$.
- $x_1^{(t)} < x_2^{(t)}$: set $x_1^{(t+1)} \leftarrow 1$.

[5] Indeed, by induction on i, we have that $z_{i+1} = z_i + \frac{1-z_i^2}{2} = \frac{(1-z_i)^2}{2} = \frac{(1-(1-2^{-2^i+1}))^2}{2} = (1-2^{-2^{i+1}+1})$.

Theorem 4.6. *The fractional* 2*-level algorithm is* $1/2 + 1/36 \approx 0.527$*-competitive.*

Proof. The analysis is via a dual fitting argument. Let $\{1, 2\}$ be the two vertices that were increased, and for simplicity we denote by x_1, x_2 their fractional degree at time t. We use the following (optimized) values for the dual nodes: $y_1 = y(\frac{1}{2}) = 17/38, y_2 = y(\frac{7}{8}) = 67/76$. These are the values of the offline nodes at the corresponding levels. To guarantee that the dual solution is feasible, we set the dual value of node t to $1 - y(x_2)$. This satisfies the dual constraints of all edges (i, t), and as the dual values are only increasing the dual constraints remain satisfied. We next analyze the four cases, proving that in each one the ratio between the values of the primal and dual changes is at most $1 + \frac{17}{19}$. This concludes the proof.

Case 1 $(x_1 = 0, x_2 = 0)$**:** $x_1 = x_2 \leftarrow \frac{1}{2}$. The value of the online node can be set to $1 - y(0) = 1$. Thus,

$$\frac{\Delta D}{\Delta P} = \frac{1 + y_1 + y_1}{1} = 1 + \frac{17}{19}.$$

Case 2 $(x_1 = \frac{1}{2}, x_2 = \frac{1}{2})$**:** $x_1 = x_2 \leftarrow \frac{7}{8}$. The value of the online node can be set to $\min\{1 - y_1\}$. Thus,

$$\frac{\Delta D}{\Delta P} = \frac{1 - y_1 + 2(y_2 - y_1)}{2 \cdot \frac{3}{8}} = 1 + \frac{17}{19}.$$

Case 3 $(x_1 = \frac{7}{8}, x_2 = \frac{7}{8})$**:** We set $x_1 \leftarrow 1$ and we can set the value of the online node to $1 - y_2$. Thus,

$$\frac{\Delta D}{\Delta P} = \frac{1 - y_2 + 1 - y_2}{\frac{1}{8}} = 1 + \frac{17}{19}.$$

Case 4 $(x_1 < x_2)$**:** We set $x_1 \leftarrow 1$ and the value of the online node can be set to $1 - y(x_2)$. There are several possible cases here (some are easily dominated by others in terms of competitiveness). $(x_1, x_2) \in \{(0, \frac{1}{2}), (0, \frac{7}{8}), (0, 1), (\frac{1}{2}, \frac{7}{8}), (\frac{1}{2}, 1), (\frac{7}{8}, 1)\}$. The worst ratio is obtained when $x_1 = 0, x_2 = \frac{1}{2}$. Thus,

$$\frac{\Delta D}{\Delta P} = \frac{1 - y(x_2) + 1 - y(x_1)}{1 - x_1} \leq \frac{1 - y_1 + 1}{1} = 1 + \frac{21}{38}.$$ □

4.2.2 Analysis of the k-level Algorithm

In this section we give a general analysis of the k-level algorithm. We prove the following Lemma.

Lemma 4.7. *Let* $g : [0, 1] \to [0, 1]$ *be a twice differentiable function that is monotone increasing, convex and bijective in* $[0, 1]$ *(and so, in particular, satisfy* $g(0) = 0$ *and* $g(1) = 1$*). Then, Algorithm 2 is* $\left(\alpha_g - O(2^{-2^k})\right)$*-competitive, where*

$$\alpha_g := \min_{x \in [0,1]} \left\{ \frac{1 - x^2}{1 - 3g(x) + 2g\left(x + \frac{1-x^2}{2}\right)}, \frac{1 - x}{2 - g(x) - g(x + \frac{1-x^2}{2})} \right\}. \quad (17)$$

Proof. The proof relies on dual fitting. Let g be a function that satisfies the conditions of the lemma. We use the function $g(\cdot)$ to assign dual values to offline node with fractional degree x_i. We assign a dual value $y_i = g(x_i)$. For any online node t whose neighbors' fractional degrees at time t are $x_1^{(t)} \leq x_2^{(t)} \leq \ldots$, we set $y_t = 1 - g(x_2^{(t)})$. This dual solution is trivially feasible. Now, consider ΔP and ΔD following an online node's arrival. We will show that $\Delta P/\Delta D$ is at least α_g for all arrivals except for a small fraction of arrivals, when weighted by their contribution to P. This will prove the competitive ratio. There are three cases to consider. **Case 1:** $x_1^{(t)} = x_2^{(t)} = z_m < z_k$**.** In this case we have that the primal gain and dual change are

$$\Delta P = 1 - z_m^2,$$
$$\Delta D = 2(g(z_{m+1}) - g(z_m)) + 1 - g(z_m).$$

By the definition of α_g, and since $z_{m+1} = z_m + \frac{(1-z_m)^2}{2}$, we have

$$\Delta P/\Delta D = \frac{1 - z_m^2}{1 - 3g(z_m) + 2g\left(z_m + \frac{1-z_m^2}{2}\right)} \geq \min_{x\in[0,1]} \frac{1 - x^2}{1 - 3g(x) + 2g\left(x + \frac{1-x^2}{2}\right)} \geq \alpha_g.$$

Case 2: $x_1^{(t)} = z_m < x_2^{(t)} < 1$**, with** $m < k$**.** In this case we have that the primal gain and dual cost are

$$\Delta P = 1 - z_m,$$
$$\Delta D \leq 1 - g(z_m) + 1 - g(z_{m+1}) = 2 - g(z_m) - g(z_{m+1}) = 2 - g(z_m) - g(z_m + \frac{1 - z_{i-1}^2}{2}).$$

By the definition of α_g,

$$\Delta P/\Delta D = \frac{1 - z_m}{2 - g(z_m) - g(z_m + \frac{1-z_{i-1}^2}{2})} \geq \min_{x\in[0,1]} \frac{1 - x}{2 - g(x) - g(x + \frac{1-x^2}{2})} \geq \alpha_g.$$

Case 3: $x_1^{(t)} = z_k$**.** In this case we have that the primal gain and dual change are

$$\Delta P = 1 - z_k,$$
$$\Delta D = 1 - g(z_k) + 1 - g(z_k) = 2 - 2g(z_k).$$

Every time we get to case 3 the fraction of a new offline node becomes 1. Hence the total dual cost in all these steps, D', is at most $2(1 - g(z_k))P$, where P is the final primal solution.

Summing up over all steps, we get that:

$$P \geq \alpha_g \cdot (D - D') = \alpha_g \cdot (D - 2(1 - g(z_k))P) \geq \alpha_g \cdot \left(D - \frac{2(1 - z_k)}{\alpha_g} P\right) = \alpha_g \cdot D - 2(1 - z_k)P,$$

where the last inequality follows since by Claim 4.3 for each $g \in \mathcal{F}$, we have $\frac{1-g(z_k)}{1-z_k} = g'(\hat{x}) \leq g'(1) \leq \frac{1}{\alpha_g} \leq \frac{1}{\alpha'_g}$, where $0 < \hat{x} < 1$. Overall, we get $P \geq \frac{\alpha_g}{1+2(1-z_k)} D = \frac{\alpha_g}{1+2^{-2^k+2}} D = \left(\alpha_g - O(2^{-2^k})\right) D$. □

The proof of the competitive ratio of the k-level algorithm follows the same argument as Theorem 4.4. In particular, the same function g can be used to show the competitive ratio, and the adversarial sequence that shows the upper bound is the same.

Theorem 4.8. *Algorithm 3 is* $\left(\alpha_g - O(2^{-2^k})\right)$*-competitive with* $\alpha \in [0.532, 0.536]$.

4.3 A 2-level Fractional Vertex-Weighted Algorithm

In this section we design a two-choice 2-level fractional algorithm for the more general *vertex-weighted* problem, where offline nodes have a weight associated with them, and we wish to output a matching of maximum weight. We prove the following:

Theorem 4.9. *There exists a fractional 2-level* $^{11}/_{21} \approx 0.524$*-competitive vertex-weighted online matching algorithm.*

Proof. The algorithm draws ideas from the 2-level algorithm for the unweighted case in Section 4.2.1, but is more involved, due to the offline weights adding another dimension of asymmetry. As in the unweighted case, the algorithm has two possible levels for the offline nodes: $z_1 = \frac{1}{2}, z_2 = \frac{7}{8}$. Let $w_i > 0$ be the weight of node i. At time t, let $w_1(1 - x_1^{(t)}) \geq w_2(1 - x_2^{(t)}) \geq \ldots, \geq w_k(1 - x_k^{(t)})$ be the neighbors of t sorted by their dual slack. We again assume wlog of generality that there are at least two neighbors. Otherwise, we add dummy neighbors with $x_i^{(t)} = 1$ that will not change the behavior of the algorithm or the analysis. Let $\{1, 2\}$ be the two offline nodes with the maximal slack (not necessarily sorted). By normalizing, we assume wlog of generality that $w_1 = 1$. We use the following (optimized) numbers: $y_1 = \frac{5}{11}, y_2 = \frac{79}{88}$. The algorithm is defined by the following cases:

- $x_1 = 0, x_2 = 0$: If $w_2 \leq \frac{1}{1-y_1}$ then set $x_1 = x_2 = \frac{1}{2}$, otherwise set $x_1 \leftarrow 1$.
- $x_1 = \frac{1}{2}, x_2 = \frac{1}{2}$: If $w_2 \leq \frac{1-y_1}{1-y_2}$ then set $x_1 = x_2 = \frac{7}{8}$, otherwise set $x_1 \leftarrow 1$.
- $x_1 = \frac{7}{8}, x_2 = \frac{7}{8}$: If $w_2 \leq 1$ then set $x_1 \leftarrow 1$, otherwise set $x_2 \leftarrow 1$.
- $x_1 = 0, x_2 = \frac{1}{2}$: If $w_2 \leq \frac{3}{2}$ then set $x_1 \leftarrow 1$, otherwise set $x_1 \leftarrow \frac{1}{2}, x_2 \leftarrow 1$.
- $x_1 = 0, x_2 = \frac{7}{8}$: If $w_2 \leq 5.5$ then set $x_1 \leftarrow 1$, otherwise set $x_1 \leftarrow \frac{7}{8}, x_2 \leftarrow 1$.
- $x_1 = \frac{1}{2}, x_2 = \frac{7}{8}$: If $w_2 \leq 4$ then set $x_1 \leftarrow 1$, otherwise set $x_2 \leftarrow 1$.
- $x_1 < 1, x_2 = 1$: set $x_1 \leftarrow 1$.

As in Observation 4.5 it is not hard to verify the following.

Observation 4.10. *The algorithm is k-level independent maximal and* 2*-bit precise.*

Analysis. The analysis proceeds via a dual fitting argument. Let $\{1, 2\}$ be the two vertices with the highest $w_i(1 - x_i^{(t)})$. We note that $\{1, 2\}$ are not numbered according to their slack, but rather according to their fractional degree, as in the algorithm's description. We assume wlog that $w_1 = 1$. We use dual values $y_1 = y(\frac{1}{2}) = \frac{5}{11}, y_2 = y(\frac{7}{8}) = \frac{79}{88}$. We note that if we give the online node a value of $\max\{\min\{1 - y(x_1^t), w_2(1 - y(x_2^t)\}, 1 - y(x_1^{t+1}), w_2(1 - y(x_2^{t+1})\}$, it will satisfy all dual constraints for edges (i, t) at time t (and hence at all future times). We therefore have that the obtained dual's cost upper bounds the optimal matching's weight. To that end, we prove the following claim, whose (rather tedious) proof is deferred to Appendix D.

Claim 4.11. *The changes to the vertex-weighted algorithm's dual and primal values at each time t satisfy*

$$(\Delta P)_t \geq \frac{11}{21} \cdot (\Delta D)_t.$$

Summing up over all time steps, we have that by Claim 4.11 and weak duality, the primal gain (i.e., the fractional matching's value) is at least

$$P = \sum_t (\Delta P)_t \geq \frac{11}{21} \cdot \sum_t (\Delta D)_t = \frac{11}{21} \cdot D \geq \frac{11}{21} \cdot OPT. \qquad \square$$

5 Conclusion and Open Questions

In this work we present a sharp threshold for the amount of randomness or advice needed to beat the optimal $1/2$ ratio of deterministic algorithms for online (vertex-weighted) matching. We obtain our results by presenting a family of competitive fractional algorithms which are losslessly roundable online. We are hopeful that our work will spur follow-ups exploring the overlooked technique of lossless online rounding in the context of online matching and related problems. Indeed, we note that any randomized algorithm trivially induces a roundable fractional algorithm (i.e., the algorithm which maintains as its solution the marginal probabilities $x_{i,t} = \Pr[(i,t) \in \mathcal{M}]$). Can the search for roundable fractional matching algorithms shed more light on the space of randomized algorithms for online (matching) problems?

Beyond this meta-question, our work suggests a number of further concrete questions. What is the optimal competitive ratio achievable using $(1 + o(1)) \log \log n$ bits of randomness? Can $1 - 1/e - o(1)$ be achieved, or is there another randomness transition for competitiveness above some ratio $\alpha > 0.532$? Achieving significantly better bounds would require switching to multiple-choice algorithms. As we show, new conditions are needed for lossless roundability of multiple-choice algorithms. What are such conditions which allow for improved approximation ratios? Can we find similar conditions for other problems in the space of online matching and online allocation? As outlined in the introduction, progress on this line of work has the potential to resolve longstanding questions in the area. We see this work as a first step in this direction.

Acknowledgements. We thank Amin Saberi for useful discussions.

Appendix

A Impossibility of Online Lossless Rounding

In this short section we briefly present an example demonstrating the impossibility of online lossless rounding. This example can be seen as a prefix of the example discussed in Devanur et al. [19] when discussing the impossibility of lossless online rounding, which is itself a special case of the lower bound of Cohen and Wajc [18].

Example 1. We consider a bipartite graph with simple a two-choice fractional matching assigning values $x_{i,t} \leftarrow 1/2$ to all edges $(i,t) \in E$, and show that this matching cannot be rounded losslessly. The first two online vertices neighbor offline vertex sets $\{1,2\}$ and $\{3,4\}$, respectively. A lossless online rounding scheme must match each edge (i,t) with probability $x_{i,t} = 1/2$, would match both offline vertices with probability one. Consequently, both these online vertices are matched with probability one. A simple averaging argument shows that for some pair $(i,j) \in \{1,2\} \times \{3,4\}$, the probability that they are both matched after these two online vertices arrive is at least $\frac{1}{4}$. Next, if another online vertex arrives that has these two vertices as neighbors, it can be matched with probability at most $\frac{3}{4}$, or strictly less than the fractional solution, which matches it to an extent of one. That is, this two-choice fractional matching cannot be rounded losslessly.

B Beyond Two Choices – Challenges

In this section we discuss our focus on two-choice algorithms, and challenges in generalizing our characterization of online roundable fractional matchings.

By a lower bound of Huang and Tao [32], two-choice fractional algorithms are at best $5/9$-competitive. Consequently, two-choice algorithms cannot yield optimal $1 - 1/e (> 5/9)$ competitive ratios for the problems we study and their generalizations. However, two-choice algorithms shed light on the technical challenge of lossless online rounding for online matching problems. Moreover, we show that these algorithms are sufficient to resolve the open question of the amount of randomness (and nondeterminism) needed to outperform deterministic algorithms' competitive ratio of $1/2$ for online bipartite matching. This mirrors the recent breakthroughs for online edge-weighted matching and adwords without the small-bids assumption [22, 34], which are also achieved via such two-choice algorithms (though using completely different techniques.)

A natural follow-up direction is to study lossless online roundability for *multiple-choice* algorithms. (i.e., algorithms with $|P_t| = |\{i \mid x_{i,t} > 0\}| > 2$ for some online nodes t). A natural generalization (and strengthening) of Condition (2) to multiple-choice algorithms, for which $P_t := \{i \mid x_{i,t} > 0\}$ need not satisfy $|P_t| \le 2$, is the following.

$$\sum_{i\in I} x_{i,t} \le 1 - \prod_{i\in I} x_i^{(t)} \qquad \forall I \subseteq P_t. \tag{18}$$

Similarly to our discussion for two-choice algorithms, this conditioned can be shown to be necessary condition for lossless online rounding in some scenarios. Perhaps surprisingly, we show that unlike for two-choice algorithms, this natural generalization of Condition (2) to multiple-choice algorithms is *not sufficient* for lossless online rounding.

Lemma B.1. *There exists a three-choice fractional matching algorithm $\mathcal{A}_f$ whose output $\vec{x}$ satisfies Condition* (18)*, such that for any randomized online matching algorithm $\mathcal{A}$, there exists a graph on which the fractional matching of $\mathcal{A}_f$ is strictly greater than the expected matching size of $\mathcal{A}$. That is, $\mathcal{A}_f$ is not losslessly roundable.*

Proof. Our proof goes via Yao's Lemma [52]. We consider the following distributions over graphs of maximum degree three, on which any fractional or randomized algorithm are trivially three-choice algorithms. We label the offline nodes $i_{1,1}, i_{1,2}, i_{2,1}, i_{2,2}, i_{3,1}, i_{3,2}$. Each online node $t \in [3]$ neighbors the two offline nodes $i_{t,1}, i_{t,2}$. In addition, we have two online nodes, 4 and 5, which neighbor a random node in each of the pairs $\{i_{t,1}, i_{t,2}\}_{t\in[3]}$. The three-choice fractional matching we consider is the following:

$$x_{i,t} = \begin{cases} 0 & (i,t) \notin E \\ \frac{1}{2} & (i,t) \in E, i = i_{t,j}, t \in [3] \\ \frac{7}{24} = \frac{1-\left(\frac{1}{2}\right)^3}{3} & (i,t) \in E, t = 4 \\ \frac{6965}{41472} = \frac{1-\left(\frac{19}{24}\right)^3}{3} & (i,t) \in E, t = 5. \end{cases}$$

We note that $\vec{x}$ is a three-choice fractional matching satisfying Condition (18). We note moreover that $x_{i,t} > \frac{1}{6}$ for $(i,t) \in E$ and $t = 5$. Consequently, this fractional matching has value at least $\sum_{i,t} x_{i,t} > 3 + \frac{7}{8} + \frac{1}{2}$. We now proceed to show that every randomized algorithm $\mathcal{A}$ outputs a matching $\mathcal{M}$ of expected size $\mathbb{E}[\mathcal{M}] = 3 + \frac{7}{8} + \frac{1}{2}$. That is, $\mathbb{E}[\mathcal{M}] < \sum_{i,t} x_{i,t}$.

Consider a deterministic algorithm $\mathcal{A}'$ run on an input drawn from the above distribution. By simple exchange arguments, due to the one-sided vertex arrivals, we may safely assume that $\mathcal{A}'$ is greedy, and matches whenever presented with an online node with at least one free neighbor [35]. Therefore, precisely one node in each pair $\{i_{t,1}, i_{t,2}\}$ is matched two online node t, and therefore the number of free nodes in the neighborhood of 4 and 5 before time 4 is distributed $Y \sim Bin(3, 1/2)$. Consequently, the expected number of online nodes matched among 4 and 5 is precisely $\mathbb{E}[\max\{2,Y\}] = 1\cdot\Pr[Y=1] + 2\cdot\Pr[Y\geq 2] = \frac{3}{8}+1 = \frac{7}{8}+\frac{1}{2}$. We conclude that the matching $\mathcal{M}$ output by the deterministic algorithm $\mathcal{A}'$ on the above distribution has expected size strictly less than the value of the fractional matching output by $\mathcal{A}_f$, namely

$$\mathbb{E}[|\mathcal{M}|] = 3 + \frac{7}{8} + \frac{1}{2} < \sum_{i,t} x_{i,t}.$$

By Yao's Lemma, for each randomized algorithm $\mathcal{A}$, outputs a matching on one of the graphs in the support of the above distribution, whose expected size is strictly smaller than the fractional matching of the three-choice algorithm $\mathcal{A}_f$, whose output satisfies Condition (18). ☐

Corollary B.2. *Condition* (18) *is not sufficient to round multiple-choice algorithms losslessly.*

C Lossless Online Rounding: A Special Case

In this section we study a special case of our rounding algorithm for the case the input 2-choice sound algorithm is also maximal, i.e., satisfies Condition (2) at equality. We also show that this obtained algorithm can be implemented efficiently, i.e., in polynomial time.

C.1 The Algorithm

In this section we show that any *maximal* two-choice sound fractional solution $\vec{x}$ can be rounded while preserving the marginal probabilities of all edges, with even stronger negative correlation properties than for non-maximal such algorithms. As before, we denote by $F_{I,t}$ the probability that a set of offline vertices $I \subseteq [n]$ is free at time t, and use $F_{i,t}$ as shorthand for $F_{\{i\},t}$.

$$\Pr\left[F_{i,t}\right] = 1 - x_i^{(t)} \qquad \forall i = 1,\ldots,n. \tag{19}$$

$$\Pr\left[F_{I,t}\right] \in \left\{0, \prod_{i\in I}(1-x_i^{(t)})\right\} \qquad \forall I \subseteq \{1,\ldots,n\}. \tag{20}$$

We call a subset I of offline nodes *negative* (at time t) if $\Pr\left[F_{I,t}\right]=0$. Otherwise, we say it is *independent*, noting that in this case $\Pr\left[F_{I,t}\right] = \prod_{i\in I}(1-x_i) = \prod_{i\in I}\Pr\left[F_{i,t}\right]$. We note that this second name is apt, since for any independent set I at time t, the variables $\{F_{i,t} \mid i \in I\}$ are indeed independent as observed in the following. Before proving this fact, we make the following simple observations.

Observation C.1. *A set that is negative at time t remains negative at all times $t' \geq t$.*

Observation C.2. *If a set I is negative at time t, then all supersets of I are negative at time t.*

By Observation C.2, If I is independent then all subsets of I are independent at time t. Therefore, for any set $J \subseteq I$, we have that $\Pr[F_{J,t}] = \prod_{j\in J}\Pr[F_{j,t}]$. Consequently, for any disjoint subsets K, J of I, by the inclusion-exclusion principle and Invariant (19), we have that

$$\begin{aligned}\Pr[F_{K,t}, \overline{F_{J,t}}] &= \sum_{r=0}^{|J|}(-1)^r \sum_{J'\subseteq J, |J'|=r} \Pr[F_{K\cup J',t}] = \sum_{r=0}^{|J|}(-1)^r \sum_{J'\subseteq J,|J'|=r}\ \prod_{j\in K\cup J'}(1-x_k^{(t)}) \\ &= \prod_{k\in K}(1-x_k^{(t)})\cdot\prod_{j\in J} x_j^{(t)} = \prod_{k\in K}\Pr[F_{k,t}]\cdot\prod_{j\in J}\left(1-\Pr[F_{j,t}]\right).\end{aligned}$$

Observation C.3. *If set I is independent at time t, then the variables $\{F_{i,t} \mid i \in I\}$ are independent.*

Our rounding algorithm for maximal sound algorithms is a special case of Algorithm 1. In particular, it is obtained from the solution to Program (Prob-Program) setting $b_i = 1$ and $a_1 = \frac{1-x_2-\Delta x_2}{(1-x_1)(1-x_2)}$ and $a_2 = \frac{1-x_1-\Delta x_1}{(1-x_1)(1-x_2)}$. Our reason to present this special case is threefold: (1) it is very simple to describe, (ii) its analysis is more elegant than Algorithm 1, and (iii) it provides stronger negative correlation properties, which yield efficient polynomial-time implementation of this rounding scheme. This last point proves particularly useful, given that the fractional matching algorithms we apply our rounding schemes are all maximal.

As asserted above, our rounding scheme of this section is very simple to describe. It examines whether the (at most) two vertices whose fractional value increased are negative or independent, and which of the offline vertices is available to be matched. Then, it carefully decides probabilistically how to match the new online vertex. The formal definition of the algorithm is the following.

Algorithm 4 Lossless Rounding: The Maximal Case

```
1: for arrival of online node t do
2:     if t has less than two neighbors then
3:         add two dummy neighbors i with Δx_i = 0 and x_i = 1        ▷ used to simplify notation
4:     let P_t := {1,2} be the two neighbors of t of highest Δx_i := x_{i,t}, and let x_i := x_i^(t)
5:     if {1,2} are negative then
6:         match t to its sole free neighbor i (if any) with probability Δx_i/(1−x_i)
7:     if {1,2} are independent then
8:         if only one i ∈ {1,2} is free then
9:             match t to i
10:        if both 1 and 2 are free then
11:            match t to node 1 with prob. (1−x_2−Δx_2)/((1−x_1)(1−x_2)) and to node 2 with prob. (1−x_1−Δx_1)/((1−x_1)(1−x_2))
```

We observe that the algorithm is well defined (and dummy vertices are never matched). Indeed, as the fractional solution guarantees that $\Delta x_i \in [0, 1 - x_i]$, we get that $\frac{\Delta x_i}{1-x_i} \in [0,1]$ and $\frac{1-x_i-\Delta x_i}{(1-x_1)(1-x_2)} \geq 0$. On the other hand, since $\Delta x_1 + \Delta x_2 = 1 - x_1 x_2$, we have in Line 11 that $\frac{1-x_2-\Delta x_2}{(1-x_1)(1-x_2)} + \frac{1-x_1-\Delta x_1}{(1-x_1)(1-x_2)} = 1$. The final equality also implies that $1, 2$ cannot both be free after time t, i.e., $\Pr[F_{\{1,2\},t+1}] = 0$.

We now turn to proving the key lemma in the analysis of Algorithm 4, namely, that this algorithm maintains the above desired invariants.

Lemma C.4. *Algorithm 4 preserves invariants* (19) *and* (20).

Proof. We prove both invariants in tandem, by induction on t. Both invariants clearly hold for $t = 1$. Assume the invariants hold for time $t \geq 1$. We prove that this implies the same for time $t+1$. For simplicity, we use the shorthand $x_i := x_i^{(t)}$ and $x'_i := x_i^{(t+1)} = x_i + \Delta x_i$, where $\Delta x_i := x_{i,t}$.

Proof of Invariant (19): We prove that $\Pr[(i,t) \in \mathcal{M}] = x_{i,t}$ for each edge $(i,t) \in E$, which implies Invariant (19) by linearity of expectation. We prove the claim for $i = 1$ (the proof for $i = 2$ is symmetric). If $\{1,2\}$ are negative, then, by the inductive hypothesis,

$$\Pr[(1,t) \in M] = \Pr[F_{1,t}, \overline{F_{2,t}}] \cdot \frac{\Delta x_1}{1-x_1} = \Pr[F_{1,t}] \cdot \frac{\Delta x_1}{1-x_1} = \Delta x_1.$$

If $\{1,2\}$ are independent, then by the inductive hypothesis, $\Pr[F_{1,t}, F_{2,t}] = (1-x_1)(1-x_2)$ and similarly, $\Pr[F_{1,t}, \overline{F_{2,t}}] = (1-x_1)x_2 = x_2 - x_1x_2$. Consequently, since $\Delta x_1 + \Delta x_2 = 1 - x_1x_2$, we have

$$\Pr[(1,t) \in M] = \Pr[F_{1,t}, \overline{F_{2,t}}] + \Pr[F_{1,t}, F_{2,t}] \cdot \frac{1 - x_2 - \Delta x_2}{(1-x_1)(1-x_2)} = x_2 - x_1x_2 + 1 - x_2 - \Delta x_2 = \Delta x_1.$$

Proof of Invariant (20)**.** By Observation C.1, a set I that ever becomes negative stays negative. Therefore, we only need to consider the case that I (before the current step) is independent, i.e., $\Pr[F_{I,t}] = \prod_{i\in I}(1-x_i)$. We prove that before the arrival of node $t+1$, we have

$$\Pr[F_{I,t+1}] \in \left\{0, \prod_{i\in I}(1-x_i')\right\}.$$

The case $\{1,2\} \cap I = \emptyset$: Then $F_{i,t} = F_{i,t+1}$ and $x_i = x_i'$ for each $i \in I$, and so trivially

$$\Pr[F_{I,t+1}] = \Pr[F_{I,t}] = \prod_{i\in I}(1-x_i) = \prod_{i\in I}(1-x_i').$$

The case $\{1,2\} \subseteq I$: As observed above, after time t at least one of $1,2$ must be matched, i.e., $\Pr[F_{\{1,2\},t+1}] = 0$. Consequently, $\Pr[F_{I,t+1}] \le \Pr[F_{\{1,2\},t+1}] = 0$.

The case $1 \in I$ and $2 \notin I$ (the opposite case is symmetric): Let E_1 be the event that 1 is not matched to t. There are two sub-cases to consider.
$\{1,2\}$ are negative. By Observation C.2, $I \cup \{2\}$ is also negative, and so by independence of I,

$$\Pr\left[\overline{F_{2,t}}, F_{I,t}\right] = \Pr[F_{I,t}] - \Pr[F_{2,t}, F_{I,t}] = \Pr[F_{I,t}] = \prod_{i\in I}(1-x_i).$$

Consequently, since $x_1' = x_1 + \Delta x_1$, we have that

$$\begin{aligned}\Pr[F_{I,t+1}] &= \Pr[E_1 \mid F_{1,t}, F_{2,t}] \cdot \Pr[F_{2,t}, F_{I,t}] + \Pr\left[E_1 \mid F_{1,t}, \overline{F_{2,t}}\right] \cdot \Pr\left[\overline{F_{2,t}}, F_{I,t}\right] \\ &= 0 + \left(1 - \frac{\Delta x_1}{1-x_1}\right) \cdot \prod_{i\in I}(1-x_i) = \prod_{i\in I}(1-x_i').\end{aligned}$$

$\{1,2\}$ is independent: If $I \cup \{2\}$ are also independent, then $\Pr[F_{2,t}, F_{I,t}] = \prod_{i\in I\cup\{2\}}(1-x_i)$. Therefore, since $\Pr[E_1 \mid F_{1,t}, F_{2,t}] = \frac{1-x_1-\Delta x_1}{(1-x_1)(1-x_2)}$, and again using $x_i' = x_1 + \Delta x_1$, we have that

$$\begin{aligned}\Pr[F_{I,t+1}] &= \Pr[E_1 \mid F_{1,t}, F_{2,t}] \cdot \Pr[F_{2,t}, F_{I,t}] + \Pr\left[E_1 \,\middle|\, F_{1,t}, \overline{F_{2,t}},\right] \cdot \Pr\left[\overline{F_{2,t}}, F_{I,t}\right] \\ &= \frac{1-x_1-\Delta x_1}{(1-x_1)(1-x_2)} \cdot (1-x_2) \cdot \prod_{i\in I}(1-x_i) + 0 = \prod_{i\in I}(1-x_i').\end{aligned}$$

Finally, we address the case that $\{1,2\}$ is independent and $I \cup \{2\}$ is negative. In this case, either 2 is not matched before time t, and some node in I must be matched, or 2 is matched before time t and the algorithm matches 1 in Line 9. Put otherwise, we have that I becomes negative, as

$$\Pr[F_{I,t+1}] = \Pr[E_1 \mid F_{1,t}, F_{2,t}] \cdot \Pr[F_{2,t}, F_{I,t}] + \Pr\left[E_1 \mid F_{1,t}, \overline{F_{2,t}}\right] \cdot \Pr\left[\overline{F_{2,t}}, F_{I,t}\right] = 0. \qquad \square$$

From Invariant (19), together with linearity of expectation, we have that $\Pr[(i,t) \in \mathcal{M}] = x_{i,t}$ for each edge (i,t). That is, we obtain our main technical result: an online lossless rounding scheme for two-choice independently-maximal fractional algorithms. To conclude Theorem 1.3 it remains to give a polytime implementation of this algorithm.

An Efficient Implementation. Algorithm 4 requires knowledge of whether or not pairs $\{1,2\}$ are negative at time t. That is, it must distinguish between $\Pr[F_{\{1,2\},t}] = 0$ and $\Pr[F_{\{1,2\},t}] = \Pr[F_{1,t}] \cdot \Pr[F_{2,t}]$. This can be easily done in exponential time by maintaining the entire probability space. In Appendix C.2 we give a more efficient implementation, by explicitly keeping track only of pairwise correlations.

Lemma C.5. *Algorithm 4 can be implemented in $O(n)$ time per online node arrival.*

C.2 An Efficient Implementation of Algorithm 4

In this section we present an efficient implementation of Algorithm 4 when run with independent random variables.

The only non-trivial part of an implementation of Algorithm 4 is determining whether or not pairs $\{1,2\}$ are negative. That is, we need to distinguish between $\Pr[F_{1,t}, F_{2,t}] = 0$ and $\Pr[F_{1,t}, F_{2,t}] = (1 - x_1^{(t)}) \cdot (1 - x_2^{(t)})$. This is trivial to check if $x_1^{(t)} = 1$ or $x_2^{(t)}$, since Invariant (20) implies that any pair (and indeed, any set) I containing a vertex i with fractional degree $x_i^{(t)} = 1$ is negative at time t, since $\Pr[F_{I,t}] \in \{0, \prod_i (1 - x_i^{(t)})\} = \{0\}$. We therefore focus on pairs which are not trivially negative, as in the following definition.

Definition C.6. *A set $I \subseteq [n]$ is* strictly negative *if it is negative and $x_i^{(t)} \neq 1$ for all $i \in I$.*

Now, determining whether a pair $\{1,2\}$ is strictly negative can be easily implemented in exponential time, by considering the decision tree defined by the algorithm. A much more efficient implementation is possible, however, as we now show.

Recall that by Observation C.2, if a set I contains a pair of nodes which are negative, then I must itself be negative. The following lemma, which will prove useful in order to implement our algorithm efficiently, shows that the converse is also true for strictly negative sets and pairs. That is, any strictly negative set I has a strictly negative "witness" consisting of a pair of nodes in I.

Lemma C.7. *A set of offline nodes I, $|I| \geq 2$ with $x_i^{(t)} \neq 1$ for all $i \in I$ is strictly negative if and only if it contains a pair $J \subseteq I$, $|J| = 2$ which is itself strictly negative.*

Proof. The "if" direction follows from Observation C.2 and definition of strict negativity. We prove the "only if" direction for all sets I by induction on t. The claim holds vacuously at time $t = 1$, at which point there are no negative sets. For the inductive step, consider some such set I with $x_i^{(t+1)} \neq 1$ for all $i \in I$ and time $t+1$.

If I was (strictly) negative by time t, then there exists a pair $J \subseteq I$, $|J| = 2$ which is strictly negative by time t, and by Observation C.1, both I and J remain negative at time $t+1$. Therefore, J is the desired strictly negative pair at time $t+1$ contained in I.

Now, suppose I was not strictly negative at time t, but it is at time $t+1$. Denote by $\{1,2\}$ the neighbors of t with non-zero probability of being matched to t. (Note that these must indeed be a pair, since if t can only be matched to at most one node 1, this node must reach fractional degree $x_1 + \Delta x_1 = \frac{x_1 + x_2 + 1 - x_1 x_2}{2} = 1$, and so all sets that become negative at time t are not strictly negative.) Inspecting the proof of Lemma C.4, we find that either $I \supseteq \{1,2\}$, in which case $J = \{1,2\}$ is the desired pair, or (wlog) $I \cap \{1,2\} = \{1\}$, and we have that $\{1,2\}$ is independent by time t and $I \cup \{2\}$ is (strictly) negative by time t. Then, by the inductive hypothesis, we have that $I \cup \{2\}$ contains a (strictly) negative pair J' by time t. Since I is not negative by time t, Observation C.2 implies that the pair J' cannot be a subset of I, and since $\{1, 2\}$ is also not negative at time t, we know that $J' \neq \{1, 2\}$, and so $J' = \{2, i\}$ for some $i \in I$. That is, we have that $\Pr[F_{i,t}, F_{2,t}] = 0$, and therefore $F_{i,t} \leq \overline{F_{2,t}}$. (In words, if i is free, 2 must be matched.) Consequently, we find that the pair $\{1, i\} \subseteq I$ becomes negative, since

$$\Pr[F_{1,t+1}, F_{i,t+1}] \leq \Pr[F_{1,t+1}, F_{i,t}] \leq \Pr[F_{1,t+1}, \overline{F_{2,t}}] = 0,$$

where the last inequality relies on $\{1, 2\}$ previously being independent, and so $\Pr[F_{1,t+1} \mid \overline{F_{2,t}}] = 0$. We conclude that if I satisfying $x_i^{(t+1)} \neq 1$ for all $i \in I$ is strictly negative at time $t + 1$, then there exists some pair $J \subseteq I$ which is itself strictly negative at time $t + 1$. □

For any offline node i which has $x_i^{(t)} \neq 1$, we denote the all offline nodes j such that the pair $\{i, j\}$ is strictly negative by time t by

$$S_i^{(t)} := \{j \mid \{i, j\} \text{ are strictly negative by time } t\}.$$

The following lemmas characterize the changes to these sets from time t to $t+1$, allowing for simple maintenance of these sets over time.

Lemma C.8. *If t has a single neighbor 1 with non-zero probability of being matched to t, then $S_1^{(t+1)} = \emptyset$ and $S_i^{(t+1)} = S_i^{(t)} \setminus \{1\}$ for all $i \neq 1$, while $S_1^{(t+1)} = \emptyset$.*

Proof. Follows from 1 reaching fractional degree $x_1^{(t+1)} = 1$ in this case, and therefore 1 no longer belongs to any strictly negative set, while for all other nodes $i \neq 1$, we have that $F_{i,t} \equiv F_{i,t+1}$, and so all pairs $\{i, j\} \not\ni 1$ are strictly negative at time $t + 1$ if and only if they are strictly negative at time t. □

Lemma C.9. *Let t be an online node with non-zero probability of being matched to nodes in $\{1, 2\}$. Then, we have*

$$S_i^{(t+1)} = \begin{cases} S_1^{(t)} \cup S_2^{(t)} \cup \{2\} & i = 1 \\ S_2^{(t)} \cup S_1^{(t)} \cup \{1\} & i = 2 \\ S_i^{(t)} \cup \{2\} & i \in S_1^{(t)} \\ S_i^{(t)} \cup \{1\} & i \in S_2^{(t)} \\ S_i^{(t)} & i \notin \{1, 2\} \cup S_1^{(t)} \cup S_2^{(t)}. \end{cases}$$

Proof. We note that for independently-maximal fractional algorithms no node reaches fractional degree $x_i^{(t+1)} = 1$ at time t. Therefore, by Observation C.1, for all i, we have that $S_i^{(t+1)} \supseteq S_i^{(t)}$. We will show that our expression for $S_i^{(t+1)} \setminus S_i^{(t)}$ is precisely the set of all other nodes j such that $\{i, j\}$ is strictly negative at time $t + 1$ but not at time t.

Consider a pair $I = \{i, j\}$ which was not strictly negative at time t, but became strictly negative at time $t + 1$. In particular, by monotonicity of x_i over time, this implies I must have been independent at time t. By the proof of Lemma C.4, this implies that one of two cases must hold:

1. $I = \{1, 2\}$.
2. $1 \in I$ and $2 \notin I$ (or vice versa) and $\{1, 2\}$ is independent at time t.

For the former case, this implies that $\Pr[F_{1,t+1}, F_{2,t+1}] = 0$, and therefore $1 \in S_2^{(t+1)}$ and $2 \in S_1^{(t+1)}$. For the latter case, consider a node $i \in S_2^{(t)}$. That is, some node i such that $\{i, 2\}$ is negative, and so $\Pr[F_{i,t}, F_{2,t}] = 0$. This implies that $F_{i,t} \leq \overline{F_{2,t}}$. (In words, if i is free, 2 must be matched.) Consequently, we find that the pair $\{1, i\} \subseteq I$ becomes negative, since

$$\Pr[F_{1,t+1}, F_{i,t+1}] \leq \Pr[F_{1,t+1}, F_{i,t}] \leq \Pr[F_{1,t+1}, \overline{F_{2,t}}] = 0,$$

where the last inequality relies on $\{1, 2\}$ being independent at time t, and so $\Pr[F_{1,t+1} \mid \overline{F_{2,t}}] = 0$. And indeed, we have that $S_2^{(t)} \subseteq S_1^{(t+1)}$. (Symmetrically, we have that $S_1^{(t)} \subseteq S_2^{(t+1)}$.)

We conclude that our expression for $S_i^{(t+1)}$ is correct. $\square$

The two preceding lemmata yield a simple linear-time algorithm for maintaining the negative pairs (by maintaining the sets $S_i^{(t)}$, in addition to the sets $x_i^{(t)}$), which by the preceding discussion yields an efficient implementation of our algorithm.

Lemma C.5. *Algorithm 4 can be implemented in $O(n)$ time per online node arrival.*

D Deferred Proofs of Section 4

In this section we provide of lemmas and theorems deferred from Section 4, restated for ease of reference.

D.1 The Weighted Algorithm

Claim 4.11. *The changes to the vertex-weighted algorithm's dual and primal values at each time t satisfy*

$$(\Delta P)_t \geq \frac{11}{21} \cdot (\Delta D)_t.$$

Proof. Dropping the subscript t, since it will be clear from context, what we wish to prove is that for any online time t and each of the cases in the algorithm's definition we have that $\frac{\Delta D}{\Delta P} \leq 1 + \frac{10}{11}$. We next analyze all cases, showing that in all cases the change in the primal value divided by the dual cost is at most $1 + \frac{10}{11}$.

Case 1 ($x_1 = 0, x_2 = 0$): In this case we can assume wlog that $w_1 = 1$ and $w_2 \geq 1$. If $w_2 \leq \frac{1}{1-y_1} = \frac{38}{19}$ then, $x_1 = x_2 = \frac{1}{2}$. The value of the online node can be at most $\max\{1, w_2(1 - y_1)\} \leq 1$. Thus,

$$\frac{\Delta D}{\Delta P} = \frac{1 + y_1 + w_2 y_1}{\frac{1}{2}(1 + w_2)} \leq \frac{1 + y_1 + y_1}{1} = 1 + \frac{10}{11}.$$

If $w_2 \geq \frac{1}{1-y_1}$ we set $x_2 = 1$ and we may set the value of the online node to w_2. In this case,

$$\frac{\Delta D}{\Delta P} = \frac{1 + w_2}{w_2} \leq \frac{1 + \frac{1}{1-y_1}}{\frac{1}{1-y_1}} = 2 - y_1 = 1 + \frac{6}{11}.$$

Case 2 ($x_1 = \frac{1}{2}, x_2 = \frac{1}{2}$): In this case we can assume wlog that $w_1 = 1$ and $w_2 \geq 1$. If $w_2 \leq \frac{1-y_1}{1-y_2} = \frac{16}{3} \approx 5.3$ then, $x_1 = x_2 = \frac{7}{8}$. The value of the online node can be set to at most $\max\{1 - y_1, w_2(1 - y_1)\} \leq 1 - y_1$. Thus,

$$\frac{\Delta D}{\Delta P} = \frac{1 - y_1 + y_2 - y_1 + w_2(y_2 - y_1)}{\frac{3}{8}(1 + w_2)} \leq \frac{1 - y_1 + y_2 - y_1 + 1(y_2 - y_1)}{\frac{3}{8}(1 + 1)} = 1 + \frac{10}{11}.$$

If $w_2 \geq \frac{1-y_1}{1-y_2}$ we set $x_2 = 1$ and the value of the online node can be set to $1 - y_1$. In this case,

$$\frac{\Delta D}{\Delta P} = \frac{1 - y_1 + w_2(1 - y_1)}{\frac{1}{2}w_2} \leq \frac{1 - y_1 + \frac{(1-y_1)^2}{1-y_2}}{\frac{1}{2}\frac{1-y_1}{1-y_2}} = \frac{1 - y_2 + 1 - y_1}{0.5} = 1 + \frac{13}{44}.$$

Case 3 ($x_1 = \frac{7}{8}, x_2 = \frac{7}{8}$): In this case we can assume wlog that $w_1 = 1$ and $w_2 \geq 1$. We set $x_2 = 1$. Thus,

$$\frac{\Delta D}{\Delta P} = \frac{1 - y_2 + w_2(1 - y_2)}{\frac{1}{8}w_2} \leq \frac{2 - 2y_2}{\frac{1}{8}} = 1 + \frac{7}{11}.$$

Case 4 ($x_1 = 0, x_2 = \frac{1}{2}$): Assume that $w_1 = 1$ If $w_2 \leq \frac{3}{2}$ then we set $x_1 = 1$. In this case $w_2(1 - y_1) \leq 1$ and so the online node can get value of $w_2(1 - y_1)$. Thus,

$$\frac{\Delta D}{\Delta P} = \frac{w_2(1 - y_1) + 1}{1} \leq \frac{\frac{3}{2}(1 - y_1) + 1}{1} = 1 + \frac{9}{11}.$$

If $w_2 \geq \frac{3}{2}$ then we set $x_1 = \frac{1}{2}, x_2 = 1$. In this case we may set the online node to 1. Thus,

$$\frac{\Delta D}{\Delta P} = \frac{1 + y_1 + w_2(1 - y_1)}{\frac{1}{2}(1 + w_2)} \leq \frac{1 + y_1 + \frac{3}{2}(1 - y_1)}{\frac{1}{2}(1 + \frac{3}{2})} = 1 + \frac{9}{11}.$$

Case 5 ($x_1 = 0, x_2 = \frac{7}{8}$): Assume that $w_1 = 1$ If $w_2 \leq 5.5$ then we set $x_1 = 1$. In this case $w_2(1 - y_2) \leq 1$ and we may set the online node to $w_2(1 - y_2)$. Thus,

$$\frac{\Delta D}{\Delta P} = \frac{w_2(1 - y_2) + 1}{1} \leq \frac{5.5(1 - y_2) + 1}{1} = 1 + \frac{9}{16}.$$

If $w_2 \geq 5.5$ then we set $x_1 = \frac{7}{8}, x_2 = 1$. In this case we may set the online node to 1. Thus,

$$\frac{\Delta D}{\Delta P} = \frac{1 + y_2 + w_2(1 - y_2)}{\frac{7}{8} + \frac{1}{8}w_2} \leq \frac{1 + y_2 + 5.5(1 - y_2)}{\frac{7}{8} + \frac{5.5}{8}} \approx 1.57.$$

Case 6 ($x_1 = \frac{1}{2}, x_2 = \frac{7}{8}$): Assume that $w_1 = 1$. If $w_2 \leq 4$ then we set $x_1 = 1$. In this case $w_2(1 - y_2) \leq 1$ and we may set the online node to $w_2(1 - y_2)$. Thus,

$$\frac{\Delta D}{\Delta P} = \frac{w_2(1 - y_2) + 1 - y_1}{\frac{1}{2}} \leq \frac{4(1 - y_2) + 1 - y_1}{\frac{1}{2}} = 1 + \frac{10}{11}.$$

If $w_2 \geq 4$ then we set $x_2 = 1$. In this case we may set the online node to $1 - y_1$. Thus,

$$\frac{\Delta D}{\Delta P} = \frac{1 - y_1 + w_2(1 - y_2)}{\frac{1}{8}w_2} \leq \frac{1 - y_1 + 4(1 - y_2)}{\frac{4}{8}} = 1 + \frac{10}{11}.$$

Case 7 ($x_1 < 1, x_2 = 1$): In this case we may set the online node to 0. Then,

$$\frac{\Delta D}{\Delta P} = \frac{1 - y(x_1)}{1 - x_1} \leq \frac{12}{11}.$$

□

E Construction of Small-Bias Probability Spaces

In this section we prove Lemma 2.3 for the sake of completeness, as it pertains to the construction of (δ, k)-dependent distributions with $\delta > 0$ that we use.

Lemma 2.3. *For any $\delta > 0$, a (δ, k)-dependent joint distribution on n binary variables can be constructed using $\log\log n + O(k + \log(\frac{1}{\delta}))$ random bits. Moreover, after polytime preprocessing, each random variable in this distribution can be sampled in $O(k \cdot \log n)$ time.*

We describe a construction suggested by Naor and Naor [42]. Consider a distribution $\mathcal{D}$ over $\{0,1\}^n$ random variables.

Definition E.1. *The* bias *of a subset $S \subseteq \{1, \dots, n\}$ for a distribution $\mathcal{D}$ is*

$$\mathrm{bias}_{\mathcal{D}}(S) := \left| \Pr_{\mathcal{D}} \left[\sum_{i \in S} x_i \equiv 0 \pmod 2 \right] - \Pr_{\mathcal{D}} \left[\sum_{i \in S} x_i \equiv 1 \pmod 2 \right] \right|.$$

We say that $\mathcal{D}$ is *k-wise ϵ-biased* if for every $S \subseteq \{1, \dots, n\}$ of size at most $|S| \leq k$, we have $\mathrm{bias}_{\mathcal{D}}(S) \leq \epsilon$. It is shown by Naor and Naor [42] that random variables that are k-wise ϵ-biased are also (δ, k)-dependent for $\delta = 2^{\frac{k}{2}} \cdot \epsilon$.

A construction by [4] for uniform k-wise random variables goes as follows. Let $v_1, \dots, v_n \in \{0,1\}^h$ be vectors that are linearly k-wise independent over GF[2], with $h = \frac{k}{2} \log n$. Such vectors are known to exist, and moreover can be constructed in polynomial time (e.g., rows of a parity check matrix of a BCH code). Choose $r \in \{0,1\}^h$ uniformly at random and define $x_i := \langle v_i, r \rangle$, for $i = 1, \dots, n$. It is easy to see that the resulting random variables $x_1, x_2, \dots, x_n$ over $\{0,1\}^n$ are k-wise independent.

To construct k-wise ϵ-biased random variables, Naor and Naor [42] use the same construction as above, except that r is now sampled from an ϵ-biased distribution over h random variables (instead of a uniformly i.i.d. source).

Claim E.2. *The n random variables constructed by sampling r from an ϵ-biased source are k-wise ϵ-biased random variables.*

Proof. For every subset S of cardinality at most $|S| \leq k$, we have

$$\sum_{i \in S} x_i = \sum_{i \in S} \langle v_i, r \rangle = r \cdot \sum_{i \in S} v_i = r \cdot M_S,$$

where $M_S \triangleq \sum_{i \in S} v_i$. Since the vectors are k-wise independent, $M_S \neq \vec{0}$. Denoting by $I \subseteq [h]$ the set of indices i for which $(M_S)_i = 1$, we have

$$\begin{aligned}
\mathrm{bias}(S) &= \left| \Pr \left[\sum_{i \in S} x_i \equiv 0 \pmod 2 \right] - \Pr \left[\sum_{i \in S} x_i \equiv 1 \pmod 2 \right] \right| \\
&= \left| \Pr \left[r \cdot M_S \equiv 0 \pmod 2 \right] - \Pr \left[r \cdot M_S \equiv 1 \pmod 2 \right] \right| \\
&= \left| \Pr \left[\sum_{i \in I} r_i \equiv 0 \pmod 2 \right] - \Pr \left[\sum_{i \in I} r_i \equiv 1 \pmod 2 \right] \right| \\
&\leq \epsilon.
\end{aligned}$$

□

Thus, the number of ϵ-biased random variables required for the construction is $h = k \log n$. The cardinality of a sample space of an ϵ-biased distribution over h variables constructed by [42] is linear in the number of random variables. Specifically, it is of size $O(h/\epsilon^3)$. Therefore, we need $\log(O(h/\epsilon^3))$ random bits to sample uniformly at random from this distribution, from which we obtain r. Finally, in order to compute any $x_i = \langle v_i, r \rangle$, we only spend $O(h) = O(k \cdot \log n)$ time. As mentioned before, the vectors v_i can be computed in polynomial time [42]. This concludes the proof of Lemma 2.3, since to obtain obtain (δ, k)-dependence, we need $\delta \leq 2^{\frac{k}{2}} \cdot \epsilon$, and so picking $\epsilon = \delta \cdot 2^{-\frac{k}{2}}$ yields a (δ, k)-dependent distribution, using $\log(O(h/\epsilon^3)) = \log \log n + O(k + \log(1/\delta)) + O(1)$ random bits.

References

[1] Adamaszek, A., Czumaj, A., Englert, M., and Räcke, H. 2012. An $o(\log k)$-competitive algorithm for generalized caching. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 1681–1689.

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

[3] Alon, N., Awerbuch, B., Azar, Y., Buchbinder, N., and Naor, J. 2009. The online set cover problem. *SIAM Journal on Computing (SICOMP) 39,* 2, 361–370.

[4] Alon, N., Babai, L., and Itai, A. 1986. A fast and simple randomized parallel algorithm for the maximal independent set problem. *J. Algorithms 7,* 4, 567–583.

[5] Bansal, N., Buchbinder, N., Madry, A., and Naor, J. 2011. A polylogarithmic-competitive algorithm for the k-server problem. In *Proceedings of the 52nd Symposium on Foundations of Computer Science (FOCS)*. 267–276.

[6] Bansal, N., Buchbinder, N., and Naor, J. 2012a. A primal-dual randomized algorithm for weighted paging. *Journal of the ACM (JACM) 59,* 4, 1–24.

[7] Bansal, N., Buchbinder, N., and Naor, J. 2012b. Randomized competitive algorithms for generalized caching. *SIAM Journal on Computing (SICOMP) 41,* 2, 391–414.

[8] Bansal, N., Buchbinder, N., and Naor, J. S. 2010. Metrical task systems and the k-server problem on hsts. In *Proceedings of the 37th International Colloquium on Automata, Languages and Programming (ICALP)*. 287–298.

[9] Böckenhauer, H.-J., Komm, D., Královič, R., and Rossmanith, P. 2014. The online knapsack problem: Advice and randomization. *Theoretical Computer Science (TCS) 527*, 61–72.

[10] Boyar, J., Favrholdt, L. M., Kudahl, C., Larsen, K. S., and Mikkelsen, J. W. 2017. Online algorithms with advice: a survey. *ACM Computing Surveys (CSUR) 50,* 2, 1–34.

[11] Bubeck, S., Cohen, M. B., Lee, J. R., and Lee, Y. T. 2019. Metrical task systems on trees via mirror descent and unfair gluing. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, T. M. Chan, Ed. SIAM, 89–97.

[12] Bubeck, S., Cohen, M. B., Lee, Y. T., Lee, J. R., and Mądry, A. 2018. K-server via multiscale entropic regularization. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC)*. 3–16.

[13] Buchbinder, N., Gupta, A., Molinaro, M., and Naor, J. 2019. k-servers with a smile: online algorithms via projections. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 98–116.

[14] Buchbinder, N., Jain, K., and Naor, J. S. 2007. Online primal-dual algorithms for maximizing ad-auctions revenue. In *Proceedings of the 15th Annual European Symposium on Algorithms (ESA)*. 253–264.

[15] Buchbinder, N. and Naor, J. S. 2009. The design of competitive online algorithms via a primal: dual approach. *Foundations and Trends® in Theoretical Computer Science (TCS) 3,* 2–3, 93–263.

[16] Coester, C. and Lee, J. R. 2019. Pure entropic regularization for metrical task systems. In *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA*, A. Beygelzimer and D. Hsu, Eds. Proceedings of Machine Learning Research Series, vol. 99. PMLR, 835–848.

[17] 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)*. 1–25.

[18] 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.

[19] 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.

[20] Dürr, C., Konrad, C., and Renault, M. 2016. On the power of advice and randomization for online bipartite matching. In *Proceedings of the 24th Annual European Symposium on Algorithms (ESA)*. 37:1–37:16.

[21] Emek, Y., Fraigniaud, P., Korman, A., and Rosén, A. 2011. Online computation with advice. *Theoretical Computer Science (TCS) 412,* 24, 2642–2656.

[22] Fahrbach, M., Huang, Z., Tao, R., and Zadimoghaddam, M. 2020. Edge-weighted online bipartite matching. In *Proceedings of the 61st Symposium on Foundations of Computer Science (FOCS)*. To appear.

[23] 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.

[24] Fortuin, C. M., Kasteleyn, P. W., and Ginibre, J. 1971. Correlation inequalities on some partially ordered sets. *Communications in Mathematical Physics 22,* 2, 89–103.

[25] Gamlath, B., Kapralov, M., Maggiori, A., Svensson, O., and Wajc, D. 2019. Online matching with general arrivals. In *Proceedings of the 60th Symposium on Foundations of Computer Science (FOCS)*. 26–38.

[26] Gandhi, R., Khuller, S., Parthasarathy, S., and Srinivasan, A. 2002. Dependent rounding in bipartite graphs. In *Proceedings of the 43rd Symposium on Foundations of Computer Science (FOCS)*. 323–332.

[27] 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.

[28] Gupta, A., Krishnaswamy, R., Kumar, A., and Panigrahi, D. 2017. Online and dynamic algorithms for set cover. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing (STOC)*. 537–550.

[29] Huang, Z., Kang, N., Tang, Z. G., Wu, X., Zhang, Y., and Zhu, X. 2020a. Fully online matching. *Journal of the ACM (JACM) 67,* 3, 1–25.

[30] Huang, Z., Tang, Z. G., Wu, X., and Zhang, Y. 2018. 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.

[31] Huang, Z., Tang, Z. G., Wu, X., and Zhang, Y. 2020b. Fully online matching ii: Beating ranking and water-filling. In *Proceedings of the 61st Symposium on Foundations of Computer Science (FOCS)*. To appear.

[32] Huang, Z. and Tao, R. 2019. Understanding zadimoghaddam's edge-weighted online matching algorithm: Unweighted case. *arXiv preprint arXiv:1910.02569*.

[33] Huang, Z. and Zhang, Q. 2020. Online primal dual meets online matching with stochastic rewards: Configuration lp to the rescue. In *Proceedings of the 52nd Annual ACM Symposium on Theory of Computing (STOC)*. 1153–1164.

[34] Huang, Z., Zhang, Q., and Zhang, Y. 2020c. Adwords in a panorama. In *Proceedings of the 61st Symposium on Foundations of Computer Science (FOCS)*. To appear.

[35] 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.

[36] Lee, J. R. 2018. Fusible hsts and the randomized k-server conjecture. In *Proceedings of the 59th Symposium on Foundations of Computer Science (FOCS)*. 438–449.

[37] Levin, R. and Wajc, D. 2021. Streaming submodular matching meets the primal-dual method. In *Proceedings of the 32nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. To appear.

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

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

[40] Mikkelsen, J. W. 2016. Randomization can be as helpful as a glimpse of the future in online computation. In *Proceedings of the 43rd International Colloquium on Automata, Languages and Programming (ICALP)*. 39:1–39:14.

[41] Naor, J., Panigrahi, D., and Singh, M. 2011. Online node-weighted steiner tree and related problems. In *Proceedings of the 52nd Symposium on Foundations of Computer Science (FOCS)*. 210–219.

[42] Naor, J. S. and Naor, M. 1993. Small-bias probability spaces: Efficient constructions and applications. *SIAM Journal on Computing (SICOMP) 22,* 4, 838–856.

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

[44] Papadimitriou, C., Pollner, T., Saberi, A., and Wajc, D. 2021. Online stochastic max-weight bipartite matching: Beyond prophet inequalities. In *Proceedings of the 22nd ACM Conference on Economics and Computation (EC).*

[45] Pena, N. and Borodin, A. 2019. On extensions of the deterministic online model for bipartite matching and max-sat. *Theoretical Computer Science (TCS) 770*, 1–24.

[46] Renault, M. P. and Rosén, A. 2015. On online algorithms with advice for the k-server problem. *Theoretical Computer Science (TCS) 56,* 1, 3–21.

[47] Renault, M. P., Rosén, A., and van Stee, R. 2015. Online algorithms with advice for bin packing and scheduling problems. *Theoretical Computer Science (TCS) 600*, 155–170.

[48] Saberi, A. and Wajc, D. 2021. The greedy algorithm is\emph {not} optimal for on-line edge coloring.

[49] Ta-Shma, A. 2017. Explicit, almost optimal, epsilon-balanced codes. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, H. Hatami, P. McKenzie, and V. King, Eds. ACM, 238–251.

[50] Tang, Z. G., Wu, X., and Zhang, Y. 2020. Towards a better understanding of randomized greedy matching. In *Proceedings of the 52nd Annual ACM Symposium on Theory of Computing (STOC)*. 1097–1110.

[51] 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.

[52] Yao, A. C.-C. 1977. Probabilistic computations: Toward a unified measure of complexity. In *Proceedings of the 18th Symposium on Foundations of Computer Science (FOCS)*. 222–227.