Edge-Weighted Online Bipartite Matching

MATTHEW FAHRBACH, Google Research, United States
ZHIYI HUANG, The University of Hong Kong, Hong Kong
RUNZHOU TAO, Columbia University, United States
MORTEZA ZADIMOGHADDAM, Google Research, United States

Online bipartite matching is one of the most fundamental problems in the online algorithms literature. Karp, Vazirani, and Vazirani (STOC 1990) gave an elegant algorithm for unweighted bipartite matching that achieves an optimal competitive ratio of $1 - 1/e$. Aggarwal et al. (SODA 2011) later generalized their algorithm and analysis to the vertex-weighted case. Little is known, however, about the most general edge-weighted problem aside from the trivial $1/2$-competitive greedy algorithm. In this article, we present the first online algorithm that breaks the long-standing $1/2$ barrier and achieves a competitive ratio of at least 0.5086. In light of the hardness result of Kapralov, Post, and Vondrák (SODA 2013), which restricts beating a $1/2$ competitive ratio for the more general monotone submodular welfare maximization problem, our result can be seen as strong evidence that edge-weighted bipartite matching is strictly easier than submodular welfare maximization in an online setting.

The main ingredient in our online matching algorithm is a novel subroutine called *online correlated selection* (OCS), which takes a sequence of pairs of vertices as input and selects one vertex from each pair. Instead of using a fresh random bit to choose a vertex from each pair, the OCS negatively correlates decisions across different pairs and provides a quantitative measure on the level of correlation. We believe our OCS technique is of independent interest and will find further applications in other online optimization problems.

CCS Concepts: • **Theory of computation** → **Online algorithms;**

Additional Key Words and Phrases: Factor-revealing linear program, free disposal, online bipartite matching, online correlated selection, primal-dual method

ACM Reference format:
Matthew Fahrbach, Zhiyi Huang, Runzhou Tao, and Morteza Zadimoghaddam. 2022. Edge-Weighted Online Bipartite Matching. *J. ACM* 69, 6, Article 45 (November 2022), 35 pages.
https://doi.org/10.1145/3556971

1 INTRODUCTION

Matchings are fundamental graph-theoretic objects that play an indispensable role in combinatorial optimization. For decades, there have been tremendous ongoing efforts to design more efficient algorithms for finding maximum matchings in terms of their cardinality and, more generally, their

Matthew Fahrbach was supported in part by an NSF Graduate Research Fellowship under grant DGE-1650044. Zhiyi Huang was supported in part by an RGC grant 17203.

Authors' addresses: M. Fahrbach and M. Zadimoghaddam, Google Research, 111 8th Ave, New York, NY 10011, USA; emails: {fahrbach, zadim}@google.com; Z. Huang, The University of Hong Kong, Pokfulam, Hong Kong; email: zhiyi@cs.hku.hk; R. Tao, Columbia University, 500 West 112th Street, New York, NY 10027, USA; email: runzhou.tao@columbia.edu.

0004-5411/2022/11-ART45
https://doi.org/10.1145/3556971

total weight. In particular, matchings in bipartite graphs have found countless applications in settings where it is desirable to assign entities from one set to those in another, for example, matching students to schools, physicians to hospitals, computing tasks to servers, and impressions in online media to advertisers. Due to the enormous growth of matching markets in digital domains, efficient online matching algorithms have become increasingly important. For example, search engine companies have created opportunities for online matching algorithms to have a massive impact in multibillion-dollar advertising markets. Motivated by these applications, we consider the problem of matching a set of impressions that arrive one by one to a set of advertisers that is known in advance. When an impression arrives, its edges to the advertisers are revealed and an irrevocable decision must be made about which advertiser to assign the impression. Karp et al. [38] provided an elegant online algorithm called Ranking to find matchings in unweighted bipartite graphs with a *competitive ratio* of $1 - 1/e$. They also proved that this is the best achievable competitive ratio. Aggarwal et al. [1] later generalized this algorithm to the vertex-weighted online bipartite matching problem and showed that a $1 - 1/e$ competitive ratio is still attainable.

The edge-weighted case, however, is less understood. This is partly due to the fact that no competitive algorithm exists without an additional assumption. To see this, consider two instances of the edge-weighted problem, each with one advertiser and two impressions. The edge weight of the first impression is 1 in both instances, and the weight of the second impression is 0 in the first instance and W in the second instance, for some arbitrarily large W. An online algorithm cannot distinguish between the two instances when the first impression arrives. However, it has to decide whether to assign this impression to the advertiser. Not assigning it gives a competitive ratio of 0 in the first instance, and assigning it gives an arbitrarily small competitive ratio of $1/W$ in the second. This problem cannot be tackled unless assigning both impressions to the advertiser is an option.[1]

In display advertising, assigning more impressions to an advertiser than they paid for only makes them happier. In other words, we can assign multiple impressions to a given advertiser. However, instead of gaining the weights of all edges assigned to it, we gain only the maximum weight, that is, the objective equals the sum of the heaviest edge weight assigned to each advertiser. This is equivalent to allowing the advertiser to dispose of previously matched edges for free to make room for new, heavier edges. This assumption is known as the *free disposal model*. In the display advertising literature [14, 39], the free-disposal assumption is well received and widely used because of its natural economic interpretation. More generally, edge-weighted online bipartite matching in the free disposal model is a special case of monotone submodular welfare maximization, for which we can apply known $1/2$-competitive greedy algorithms [16, 41].

1.1 Our Contributions

Despite 30 years of research in online matching since the seminal work of Karp et al. [38] and more than a decade of efforts after Feldman et al. [14] introduced the free disposal model, deciding whether there exists an edge-weighted online bipartite matching algorithm that achieves a competitive ratio greater than $1/2$ has remained a tantalizing open problem. This article presents a new online algorithm and answers the question affirmatively, breaking the long-standing $1/2$ barrier (under free disposal).

Theorem 1.1. *There is a 0.5086-competitive algorithm for edge-weighted online bipartite matching.*

[1] The text addresses only deterministic algorithms. To rule out randomized algorithms, consider n instances, each with one advertiser and n impressions. In instance i, the first i impressions have weights $1, W, W^2, \ldots, W^{i-1}$ whereas the remaining impressions have zero weights. The probability that an online algorithm assigns impression i to the advertiser must be consistent for instances i to n; denote this by p_i. Since $\sum_{i=1}^{n} p_i \leq 1$, we have that $p_{i^*} \leq 1/n$ for some i^*. The algorithm is then at most $1/n$ competitive for instance i^*.

Given the hardness result of Kapralov et al. [36] that restricts beating a $1/2$ competitive ratio for monotone submodular welfare maximization assuming that **NP** $\neq$ **RP**, our algorithm shows that edge-weighted bipartite matching is strictly easier than submodular welfare maximization in an online setting.

From here, we will use the more formal terminologies of offline and online vertices in a bipartite graph instead of advertisers and impressions. One of our main technical contributions is a novel algorithmic ingredient called *online correlated selection* (OCS), which is an online subroutine that takes a sequence of pairs of vertices as input and selects one vertex from each pair. Instead of using a fresh random bit to make each of its decisions, the OCS asks to what extent the decisions across different pairs can be negatively correlated and guarantees that a vertex appearing in k pairs is selected at least once with probability strictly greater than $1-2^{-k}$. See Section 3 for a short introduction and Section 5 for the full details.

Given an OCS, we can achieve a better than $1/2$ competitive ratio for *unweighted* online bipartite matching with the following (barely) randomized algorithm. For each online vertex, either pick a pair of offline neighbors and let the OCS select one or choose one offline neighbor deterministically. Concretely, among the neighbors that have not been matched deterministically, find the least-matched ones, that is, those that have appeared in the least number of pairs so far. Pick two if there are at least two; otherwise, choose one deterministically. We analyze this algorithm in Appendix A.

Although the competitive ratio of the just described algorithm is far worse than the optimal $1 - 1/e$ ratio by Karp et al. [38], it benefits from improved generalizability. To extend this algorithm to the *edge-weighted* problem, we need a reasonable notion of "least-matched" offline neighbors. Suppose that one neighbor's heaviest edge weight is either 1 or 4 each with probability $1/2$ (over the randomness of the algorithm), another neighbor's heaviest edge is 2 with certainty, and their edge weights with the current online vertex are both 3. Which one is less matched? To tackle this, we use the online primal-dual framework for matching problems by Devanur et al. [9] along with an alternative formulation of the edge-weighted online bipartite matching problem by Devanur et al. [8]. In short, we account for the contribution of each offline vertex by weight levels. At each weight level, we consider the probability that the heaviest edge matched to the vertex has weight of at least this level. This is the complementary cumulative distribution function (CCDF) of the heaviest edge weight; hence, we call this the *CCDF viewpoint*. Then, for each offline neighbor, we utilize the dual variables to compute an offer at each weight level should the current online vertex be matched to it. The neighbor with the largest net offer aggregating over all weight levels is considered the "least matched." We introduce the online primal-dual framework and the CCDF viewpoint in Section 2. We formally present our edge-weighted matching algorithm in Section 4, followed by its analysis. In Appendix B, we include hard instances showing that the competitive ratio of our algorithm is nearly tight.

We draw connections between the online primal-dual algorithm in this article and the original algorithm of Fahrbach and Zadimoghaddam [12] in Appendix C. The algorithms share two key ideas. First, the original algorithm uses a pairing mechanism to safely make adaptive decisions based on potential past assignments. This is the initial version of negative correlation, that is, the mechanism is a primitive OCS, and it is crucial for making their analysis of the conditional probabilities tractable. Second, neither algorithm is fully adaptive—both algorithms operate in the expected state over all of their possible branches instead of conditioning on the outcomes of their internal randomness. This greatly limits the amount of randomness in the algorithms and means that most of the variables are deterministic quantities governed solely by the input graph and arrival order of the online vertices.

1.2 Related Work

While the online weighted bipartite matching algorithms literature is extensive, most of these works achieve competitive ratios greater than $1/2$ by assuming that offline vertices have large capacities or that some stochastic information about the online vertices is known in advance. In this section, we list the most relevant works and refer interested readers to the excellent survey of Mehta [44]. There have also been several significant advances in more general settings, including new arrival models and non-bipartite graphs [2, 17, 18, 24, 25, 29].

Large Capacities. The capacity of an offline vertex is the number of online vertices that can be assigned to it. Exploiting the large-capacity assumption to beat $1/2$ dates back 2 decades to Kalyanasundaram and Pruhs [35]. Feldman et al. [14] provided a $(1 - 1/e)$-competitive algorithm for Display Ads, which is equivalent to edge-weighted online bipartite matching assuming large capacities. Under analogous assumptions, the same competitive ratio was obtained for AdWords [6, 45], in which offline vertices have a budget constraint on the total weight that can be assigned to them rather than the number of impressions. From a theoretical point of view, one of the main goals in the online matching literature is to develop algorithms with a competitive ratio greater than $1/2$ without making any assumption on the capacities of offline vertices.

Stochastic Arrivals. If we have knowledge about the arrival patterns of online vertices, we can often leverage this information to design better algorithms. Typical stochastic assumptions include assuming that the online vertices are drawn from some known or unknown distribution [4, 10, 15, 22, 26, 32, 37, 43] or that they arrive in a random order [7, 13, 20, 21, 28, 33, 42]. These works achieve a $1 - \varepsilon$ competitive ratio if the large-capacity assumption holds in addition to the stochastic assumptions, or at least $1 - 1/e$ for arbitrary capacities. Hybrid models that mix adversarial and stochastic assumptions have also been studied and are known to be very powerful in practice [11, 46]. Korula et al. [40] showed that the greedy algorithm is 0.505-competitive for the more general problem of submodular welfare maximization if the online vertices arrive in a random order, without any assumption on the capacities. The analysis is later simplified and improved to 0.509-competitive by Buchbinder et al. [5]. The random order assumption is particularly justified because Kapralov et al. [36] proved that beating $1/2$ for submodular welfare maximization in the oblivious adversary model implies that $\mathbf{NP} = \mathbf{RP}$.

Subsequent Work. Several articles featuring work done concurrent with the work in this article have explored the open problems in Section 6 about improving OCS. Gao et al. [19] showed that the optimal "level of negative correlation" of an OCS is between $[0.167, 0.25]$, improving on our bounds of $[0.1099, 1)$. Unlike the matching-based approach in this article, their OCS uses probabilistic automata to obtain stronger negative correlation. Using their 0.167-OCS, they provided a 0.519-competitive algorithm for edge-weighted online bipartite matching. Blanc and Charikar [3] showed that *multiway OCS* (i.e., considering more than two elements in each round) is strictly more powerful than two-way OCS, obtaining a 0.5368 competitive ratio through a simpler reduction from online matching. Among other novelties, they relaxed the distinction between consecutive and inconsecutive steps in our definition of OCS to partially ignore small gaps. Shin and An [47] constructed a three-way OCS composed of two-way OCS subroutines to provide a 0.513-competitive algorithm for edge-weighted online bipartite matching when combined with the 0.167-OCS of Gao et al. [19].

Huang et al. [31] generalized the OCS technique to break the $1/2$ barrier for the AdWords problem, achieving a 0.5016-competitive algorithm for general bids without any stochastic assumptions.

Huang et al. [27] applied an OCS to online stochastic matching and improved the competitive ratios of unweighted and vertex-weighted matching to 0.716. Tang et al. [48] further used OCS to break the $1 - 1/e$ barrier for online stochastic matching with non-IID online vertices.

2 PRELIMINARIES

The *edge-weighted online matching* problem considers a bipartite graph $G = (L, R, E)$, where L and R are the sets of vertices on the left-hand side (LHS) and right-hand side (RHS), respectively, and $E \subseteq L \times R$ is the set of edges. Every edge $(i, j) \in E$ is associated with a nonnegative weight $w_{ij} \geq 0$, and we can assume without loss of generality that this is a complete bipartite graph, that is, $E = L \times R$, by assigning zero weights to the missing edges.

The vertices on the LHS are offline in that they are all known to the algorithm in advance. The vertices on the RHS, however, arrive online one at a time. When an online vertex $j \in R$ arrives, its incident edges and their weights are revealed to the algorithm, which must then irrevocably match j to an offline neighbor. Each offline vertex can be matched any number of times; however, only the weight of its heaviest edge counts towards the objective. This is equivalent to allowing a matched offline vertex i, say, to j, to be rematched to a new online vertex j' with edge weight $w_{ij'} > w_{ij}$, disposing of vertex j and edge (i, j) for free. This assumption is known as the *free disposal model* [14].

The goal is to maximize the total weight of the matching. A randomized algorithm is Γ-*competitive* if its expected objective value is at least Γ times the offline optimal, in hindsight, for any instance of edge-weighted online matching. We refer to $0 \leq \Gamma \leq 1$ as the *competitive ratio* of the algorithm.

2.1 Complementary Cumulative Distribution Function Viewpoint

Next, we describe an alternative formulation of the edge-weighted online matching problem due to Devanur et al. [8]. This approach captures the contribution of each offline vertex $i \in L$ to the objective in terms of the CCDF of the heaviest edge weight matched to i. We refer to this method as the *CCDF viewpoint.*

For any offline vertex $i \in L$ and any weight level $w \geq 0$, let $y_i(w)$ be the CCDF of the weight of the heaviest edge matched to i, that is, the probability that the algorithm has matched at least one online vertex j to i such that $w_{ij} \geq w$. It follows that $y_i(w)$ is a non-increasing function of w that takes values between 0 and 1. Further observe that $y_i(w)$ is a step function with polynomially many pieces, since the number of pieces is at most the number of incident edges. Hence, we will be able to maintain $y_i(w)$ in polynomial time.

The expected weight of the heaviest edge matched to i then equals the area under $y_i(w)$, that is,

$$\int_0^\infty y_i(w)\,\mathrm{d}w. \tag{1}$$

This follows from an alternative formula for the expected value of a nonnegative random variable involving only its cumulative distribution function.

We illustrate this idea with an example in Figure 1. Suppose that an offline vertex i has four online neighbors j_1, j_2, j_3, and j_4 with edge weights $w_1 < w_2 < w_3 < w_4$. Further, suppose that j_1 is matched to i with certainty, whereas j_2, j_3, and j_4 each have some probability of being matched to i. (The latter events may be correlated.) Next, suppose that a new neighbor arrives whose edge weight is also w_3. The values of $y_i(w)$ are then increased for $w_1 < w \leq w_3$ accordingly, and the total area of the shaded regions is the increment in the expected weight of the heaviest edge matched to vertex i.

2.2 Online Primal-Dual Framework

We analyze our algorithm using a linear program (LP) for edge-weighted matching under the online primal-dual framework. Consider the standard matching LP and its dual below. We interpret the primal variables x_{ij} as the probability that (i, j) is the heaviest edge matched to vertex i.

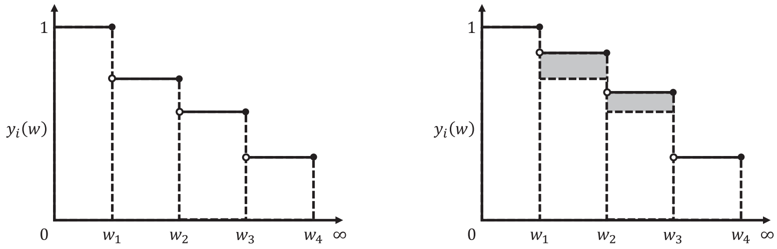


Fig. 1. Complementary cumulative distribution function (CCDF) viewpoint. The first function is the CCDF of vertex i, and the second function demonstrates how the CCDF of vertex i is updated.

$$\begin{array}{llll} \text{maximize} & \sum_{i\in L}\sum_{j\in R} w_{ij}x_{ij} & & \\ \text{subject to} & \sum_{j\in R} x_{ij} \le 1 & \forall\, i \in L \\ & \sum_{i\in L} x_{ij} \le 1 & \forall\, j \in R \\ & x_{ij} \ge 0 & \forall\, i \in L, \forall\, j \in R \end{array} \qquad \begin{array}{llll} \text{minimize} & \sum_{i\in L}\alpha_i + \sum_{j\in R}\beta_j & \\ \text{subject to} & \alpha_i + \beta_j \ge w_{ij} & \forall\, i \in L, \forall\, j \in R \\ & \alpha_i \ge 0 & \forall\, i \in L \\ & \beta_j \ge 0 & \forall\, j \in R \end{array}$$

Let P denote the primal objective. If x_{ij} is the probability that (i, j) is the heaviest edge matched to i, then P also equals the objective of the algorithm. Let D denote the dual objective.

Online algorithms under the online primal-dual framework maintain not only a matching but also a dual assignment (not necessarily feasible) at all times subject to the conditions summarized in the following.

Lemma 2.1. *Suppose that an online algorithm simultaneously maintains primal and dual assignments such that for some constant* $0 \le \Gamma \le 1$, *the following conditions hold at all times:*

(1) **Approximate dual feasibility:** *For any* $i \in L$ *and any* $j \in R$, *we have that* $\alpha_i + \beta_j \ge \Gamma \cdot w_{ij}$.
(2) **Reverse weak duality:** *The objectives of the primal and dual assignments satisfy* $P \ge D$.

Then, the algorithm is Γ*-competitive.*

Proof. By the first condition, the values $\Gamma^{-1}\alpha_i$ and $\Gamma^{-1}\beta_j$ form a feasible dual assignment with objective value Γ^{-1}D. By weak duality of linear programming, the objective of any feasible dual assignment upper bounds the optimal (i.e., D is at least Γ times the optimal). Applying the second condition proves the lemma. □

Online Primal-Dual in the CCDF Viewpoint. In light of the CCDF viewpoint, for any offline vertex $i \in L$ and any weight level $w > 0$, we introduce and maintain new variables $\alpha_i(w)$ that satisfy

$$\alpha_i = \int_0^\infty \alpha_i(w)\,\mathrm{d}w. \tag{2}$$

Accordingly, we rephrase approximate dual feasibility in Lemma 2.1 in the CCDF viewpoint as

$$\int_0^\infty \alpha_i(w)\,\mathrm{d}w + \beta_j \ge \Gamma \cdot w_{ij}. \tag{3}$$

Concretely, at each step of our primal-dual algorithm, $\alpha_i(w)$ is a piecewise constant function with possible discontinuities at the weight levels $w \in \{w_{ij} : \text{online vertex } j \text{ has arrived}\}$. Initially, all of the $\alpha_i(w)$s are the zero function. Then, as each online vertex $j \in R$ arrives, if j is potentially matched to an offline candidate $i \in L$, the function values of $\alpha_i(w)$ are systematically increased according to the dual update rules in Section 4.1. In contrast, each dual variable β_j is a scalar value that is initialized to zero and increased only once during the algorithm, at the time when j arrives.

3 ONLINE CORRELATED SELECTION: AN INTRODUCTION

This section introduces our novel ingredient for online algorithms, which we believe to be widely applicable and of independent interest. To motivate this technique, consider the following thought experiment in the case of *unweighted* online matching, that is, $w_{ij} \in \{0, 1\}$ for any $i \in L$ and any $j \in R$.

Deterministic Greedy. We first recall why all deterministic greedy algorithms that match each online vertex to an unmatched offline neighbor are at most $1/2$-competitive. Consider an instance with a graph that has two offline and two online vertices. The first online vertex is adjacent to both offline vertices; the algorithm deterministically chooses one of them. The second online vertex, however, is adjacent to the previously matched vertex only.

Two-Choice Greedy with Independent Random Bits. We can avoid this problem by matching the first online vertex randomly, which improves the expected matching size from 1 to 1.5. In this spirit, consider the following two-choice greedy algorithm. When an online vertex arrives, identify its neighbors that are least likely to be matched (over the randomness in previous rounds). If there is more than one such neighbor, choose any two, for example, lexicographically, and match to one with a fresh random bit. Otherwise, match to the least-matched neighbor deterministically. We refer to the former as a *randomized round* and the latter as a *deterministic round*. Since each randomized round uses a fresh random bit, this is equivalent to matching to neighbors that have been chosen in the least number of randomized rounds and in no deterministic round. Unfortunately, this algorithm is also $1/2$-competitive due to upper triangular graphs. We defer this standard example to Appendix B.1.

Two-Choice Greedy with Perfect Negative Correlation. The last algorithm in this thought experiment is an imaginary variant of two-choice greedy that perfectly and negatively correlates the randomized rounds so that each offline vertex is matched with certainty after being a candidate in two randomized rounds. It is impossible to achieve such perfect negative correlation in the online setting in general (see Appendix B.3 for an explanation). Nevertheless, if we assume feasibility then this algorithm is $5/9$-competitive [30]. In fact, it is effectively the 2-matching algorithm of Kalyanasundaram and Pruhs [35] by having two copies of each online vertex and allowing offline vertices to be matched twice. This motivates the following question:

Can we use partial negative correlation to retain feasibility and break the $1/2$ barrier?

We answer this affirmatively by introducing an algorithmic ingredient called *online correlated selection* (OCS), which allows us to quantify the amount of negative correlation between the randomized rounds. Appendix A provides an analysis of the two-choice greedy algorithm powered by OCS in the unweighted case. Section 4 generalizes this approach to edge-weighted online matching, giving us the first algorithm with a competitive ratio that is provably greater than $1/2$.

Definition 3.1 (γ-semi-OCS). Consider a set of ground elements. For any $\gamma \in [0, 1]$, a γ-semi-OCS is an online algorithm that takes as input a sequence of pairs of elements, and selects one element per pair in an online fashion such that if an element has appeared in $k \geq 1$ pairs, it is selected at least once with probability at least

$$1 - 2^{-k}(1 - \gamma)^{k-1}.$$

Using independent random bits is a 0-semi-OCS, and the perfect negative correlation in the thought experiment is a 1-semi-OCS, although it is typically infeasible. Our algorithms satisfy a stronger definition, which considers any collection of pairs containing an element i. This stronger definition is useful for generalizing to the edge-weighted bipartite matching problem.

In the following definition, a subsequence (not necessarily contiguous) of pairs containing element i is *consecutive* if it includes all of the pairs that contain element i between the first and last pair in the subsequence. Two subsequences of pairs are *disjoint* if no pair belongs to both of them. For example, consider the sequence $(\{a, i\}, \{b, i\}, \{c, d\}, \{e, i\}, \{i, z\})$. The subsequences $(\{a, i\}, \{b, i\})$ and $(\{i, z\})$ are consecutive and disjoint, but the subsequence $(\{a, i\}, \{b, i\}, \{i, z\})$ is not consecutive because it does not include the pair $\{e, i\}$. As a special case, a pair is consecutive to another pair if they share an element i and no other pairs between them contain element i.

Definition 3.2 (γ-OCS). Consider a set of ground elements. For any $\gamma \in [0, 1]$, a γ-OCS is an online algorithm that takes as input a sequence of pairs of elements and selects one per pair such that for any element i and any disjoint subsequences of $k_1, k_2, \dots, k_m$ consecutive pairs containing i, i is selected in at least one of these pairs with probability at least

$$1 - \prod_{\ell=1}^{m} 2^{-k_\ell} (1 - \gamma)^{k_\ell - 1} .$$

THEOREM 3.3. *There exists a* $\frac{13\sqrt{13}-35}{108} > 0.1099$*-OCS.*

We defer the design and analysis of the 0.1099-OCS to Section 5. Here, we describe a weaker $1/16$-OCS, which suffices for breaking the $1/2$ barrier. We give the formal proof of the $1/16$-OCS construction in Section 5.1.

PROOF SKETCH OF A $1/16$-OCS. Consider two sequences of independent random bits. The first sequence is used to construct a random matching among the pairs, where any two consecutive pairs (with respect to some common element) are matched with probability $1/16$. Each pair is consecutive to at most four pairs, one before it and one after it for each of its two elements. For each pair, choose one of its consecutive pairs, each with probability $1/4$. Two consecutive pairs are matched if they choose each other.

The second random sequence is used to select an element from each pair. For an unmatched pair, choose one of its elements with a fresh random bit. For any two matched pairs, use a fresh random bit to choose an element in the first pair. Then, make the opposite selection in the later pair (i.e., select the common element if it is not selected in the earlier pair and vice versa). Observe that even if two matched pairs are identical, there is no ambiguity in the opposite selection.

Next, fix any element i and any disjoint subsequences of $k_1, k_2, \dots, k_m$ consecutive pairs that contain i. We bound the probability that i is never selected. If any two of these pairs are matched, i is selected once in the two pairs. Otherwise, the selections from the pairs are independent and the probability that i is never selected is $\prod_{\ell=1}^{m} 2^{-k_\ell}$. Applying the law of total probability to the event that i is in a matched pair, it remains to upper bound the probability that no pairs are matched by $\prod_{\ell=1}^{m}(1-1/16)^{k_\ell-1}$. Intuitively, this is because there are $k_\ell - 1$ choices of two consecutive pairs within the ℓ-th subsequence, each of which is matched with probability $1/16$. These events are negatively dependent; thus, the probability that none of them happens is upper bounded by the independent case. □

Readers familiar with the theory of *negative association* [34] can directly argue that the events in this proof sketch are negatively dependent, which then makes this a complete proof. Instead of using negative association, we present a proof of this warm-up case from first principles in Section 5.1. In the process, we will build up tools for constructing and analyzing the OCS in Theorem 3.3.

4 EDGE-WEIGHTED ONLINE MATCHING

This section presents an online primal-dual algorithm for the edge-weighted online bipartite matching problem. The algorithm uses a γ-OCS as a black box, and its competitive ratio depends on the value of γ. For $\gamma = 1/16$ (as sketched in Section 3) it is 0.505-competitive and for $\gamma \approx 0.1099$ (as in Theorem 3.3) it is 0.5086-competitive, proving our main result about edge-weighted online matching.

4.1 Online Primal-Dual Algorithm

The algorithm is similar to the two-choice greedy in the previous section. It maintains an OCS with the offline vertices as the ground elements. For each online vertex, the algorithm either (1) matches it deterministically to one offline neighbor, (2) chooses a pair of offline neighbors and matches to the one selected by the OCS, or (3) leaves it unmatched. We refer to the first case as a *deterministic* round, the second as a *randomized* round, and the third as an *unmatched* round.

How does the algorithm decide whether it is a randomized, deterministic, or unmatched round, and how does it choose the candidate offline vertices? We leverage the online primal-dual framework. When an online vertex j arrives, it calculates for every offline vertex i how much the dual variable β_j would gain if j is matched to i in a deterministic round, denoted as $\Delta_i^D \beta_j$, and similarly $\Delta_i^R \beta_j$ for a randomized round. Then, it finds i^* with the maximum $\Delta_i^D \beta_j$ and i_1, i_2 with the maximum $\Delta_i^R \beta_j$. If both $\Delta_{i_1}^R \beta_j + \Delta_{i_2}^R \beta_j$ and $\Delta_{i^*}^D \beta_j$ are negative, it leaves j unmatched. If $\Delta_{i_1}^R \beta_j + \Delta_{i_2}^R \beta_j$ is nonnegative and greater than $\Delta_{i^*}^D \beta_j$, it matches j in a randomized round with i_1 and i_2 as the candidates using its OCS. Finally, if $\Delta_{i^*}^D \beta_j$ is nonnegative and greater than $\Delta_{i_1}^R \beta_j + \Delta_{i_2}^R \beta_j$, it matches j to i^* in a deterministic round. See Algorithm 1 for the formal definition.

It remains to explain how $\Delta_i^D \beta_j$ and $\Delta_i^R \beta_j$ are calculated. For any offline vertex $i \in L$ and any weight level $w > 0$, let $k_i(w)$ be the number of randomized rounds in which i has been chosen (as input to the OCS) and has edge weight at least w. The values of $k_i(w)$ may change over time; thus, we consider these values at the beginning of each online round. The increments to the dual variables $\alpha_i(w)$ and β_j depend on the values of $k_i(w)$ via the following *gain-sharing* parameters, which we determine later using a factor-revealing LP to optimize the competitive ratio. The gain-sharing values are listed at the end of this section in Table 1.

- $a(k)$: Amortized increment in the dual variable $\alpha_i(w)$ if i is chosen as one of the two candidates in a randomized round in which its edge weight is at least w and $k_i(w) = k$.
- $b(k)$: Increment in the dual variable β_j due to an offline vertex i at weight level $w \le w_{ij}$ if j is matched in a randomized round with i as one of the two candidates and $k_i(w) = k$.

Note that these gain-sharing values $a(k)$ and $b(k)$ are instance independent (i.e., they do not depend on the underlying graph) and defined for all $k \in \mathbb{Z}_{\ge 0}$. We interpret these parameters according to a gain-splitting rule. If i is one of the two candidates to be matched to j in a randomized round, the increase in the expected weight of the heaviest edge matched to i equals the integration of $y_i(w)$'s increments, for $0 < w \le w_{ij}$, which can be related to the values of the $k_i(w)$s. We then lower bound the gain due to the increment of $y_i(w)$ using the definition of a γ-OCS and split the gain into two parts, $a(k_i(w))$ and $b(k_i(w))$. The former is assigned to $\alpha_i(w)$ and the latter goes to β_j.

In fact, we prove at the end of this subsection the following invariant about how the dual variables $\alpha_i(w)$ are incremented:

$$\alpha_i(w) \ge \sum_{0 \le \ell < k_i(w)} a(\ell). \tag{4}$$

ALGORITHM 1: Online Primal-Dual Edge-Weighted Bipartite Matching Algorithm

```
State variables:
  • $k_i(w) \ge 0$: Number of randomized rounds in which $i$ is a candidate and its edge weight is at least $w$;
    $k_i(w) = \infty$ if it has been chosen in a deterministic round in which its edge weight is at least $w$.
On the arrival of an online vertex $j \in R$:
(1) For every offline vertex $i \in L$, compute $\Delta_i^R \beta_j$ and $\Delta_i^D \beta_j$ according to Equations (5) and (6).
(2) Find $i_1, i_2$ with the maximum $\Delta_i^R \beta_j$.
(3) Find $i^*$ with the maximum $\Delta_i^D \beta_j$.
(4) If $\Delta_{i_1}^R \beta_j + \Delta_{i_2}^R \beta_j < 0$ and $\Delta_{i^*}^D \beta_j < 0$, leave $j$ unmatched.    (unmatched)
(5) If $\Delta_{i_1}^R \beta_j + \Delta_{i_2}^R \beta_j \ge 0$ and $\Delta_{i_1}^R \beta_j + \Delta_{i_2}^R \beta_j \ge \Delta_{i^*}^D$, let the OCS pick one of $i_1$ and $i_2$.    (randomized)
(6) If $\Delta_{i^*}^D \beta_j \ge 0$ and $\Delta_{i^*}^D \beta_j > \Delta_{i_1}^R \beta_j + \Delta_{i_2}^R \beta_j$, match $j$ to $i^*$.    (deterministic)
(7) Update the $k_i(w)$s accordingly.
```

Next, define $\Delta_i^R \beta_j$ to be

$$\Delta_i^R \beta_j \overset{\text{def}}{=} \int_0^{w_{ij}} b\left(k_i(w)\right) \mathrm{d}w - \frac{1}{2} \int_{w_{ij}}^{\infty} \sum_{0 \le \ell < k_i(w)} a(\ell) \, \mathrm{d}w. \tag{5}$$

We should think of $\Delta_i^R \beta_j$ as the increase in the dual variable β_j due to offline vertex i, if i is chosen as one of the two candidates for j in a randomized round. The first term in Equation (5) follows from the interpretation of $b(k)$ cited earlier (and would be the only term in the unweighted case). The second term is designed to cancel out the extra help we get from the $\alpha_i(w)$s at weight-levels $w > w_{ij}$ in order to satisfy approximate dual feasibility for the edge (i, j). Concretely, if j is matched in a randomized round to two candidates at least as good as i, our choice of $b(k)$'s ensures approximate dual feasibility between i and j (i.e., the following inequality holds):

$$\int_0^{\infty} \alpha_i(w) \, \mathrm{d}w + 2 \cdot \Delta_i^R \beta_j \ge \Gamma \cdot w_{ij}.$$

Finally, for some $1 < \kappa < 2$, define the value of $\Delta_i^D \beta_j$ to be

$$\Delta_i^D \beta_j \overset{\text{def}}{=} \kappa \cdot \Delta_i^R \beta_j = \kappa \int_0^{w_{ij}} b\left(k_i(w)\right) \mathrm{d}w - \frac{\kappa}{2} \int_{w_{ij}}^{\infty} \sum_{0 \le \ell < k_i(w)} a(\ell) \, \mathrm{d}w. \tag{6}$$

For concreteness, readers can assume that $\kappa = 1.5$. The competitive ratio, however, is insensitive to the choice of κ as long as it is neither too close to 1 nor to 2. On one hand, $\kappa > 1$ ensures that if the algorithm chooses a randomized round with offline vertex i_1 and another vertex i_2 as the candidates, the contribution from i_2 to β_j must be at least a $\kappa-1$ fraction of what i_1 offers. Otherwise, the algorithm would have preferred a deterministic round with i_1 alone. On the other hand, we have $\kappa < 2$ because otherwise a randomized round would always be inferior to a deterministic round. We further explain the definitions of $\Delta_i^R \beta_j$ and $\Delta_i^D \beta_j$ in Section 4.3 and demonstrate how their terms interact when proving that the dual assignments always satisfy approximate dual feasibility.

Primal Increments. We have defined the primal algorithm and, implicitly, how the dual algorithm updates the β_js. It remains to define the updates to the $\alpha_i(w)$s. First, we need to characterize the primal increment since the dual updates are driven by it. Recall that by the CCDF viewpoint:

$$\mathrm{P} = \sum_{i \in L} \int_0^{\infty} y_i(w) \, \mathrm{d}w.$$

Since it is difficult to account for the exact CCDF $y_i(w)$ due to complicated correlations in the selections, we instead consider a lower bound for it given by the γ-OCS. A critical observation here is that the decisions made by the primal-dual algorithm are deterministic except for the randomness in the OCS. In particular, its choices of i_1, i_2, i^* and the decisions about whether a round is unmatched, randomized, or deterministic are independent of the selections in the OCS and, therefore, *deterministic quantities governed solely by the input graph and arrival order of the online vertices.* Hence, we may view the sequence of pairs of candidates the OCS considers as fixed.

For any offline vertex i and weight level $w > 0$, consider the randomized rounds in which i is a candidate and has edge weight at least w. Decompose these rounds into disjoint collections of, say, $k_1, k_2, \ldots, k_m$ consecutive rounds. We consider the nontrivial partition of $k_1, k_2, \ldots, k_m$ subsequences instead of one large subsequence because i may have been chosen in intermediate randomized rounds with weight level less than w. This is *precisely why* we need the guarantee of a γ-OCS instead of a γ-semi-OCS. By Definition 3.2, vertex i is selected by the algorithm (either deterministically or in a randomized round by the γ-OCS) with probability at least

$$\overline{y}_i(w) \overset{\text{def}}{=} \begin{cases} 1 & \text{if } i \text{ has been matched in a deterministic round;} \\ 1 - \prod_{\ell=1}^{m} 2^{-k_\ell} \left(1-\gamma\right)^{k_\ell - 1} & \text{otherwise.} \end{cases} \tag{7}$$

Accordingly, we will use the following surrogate primal objective:

$$\overline{\mathrm{P}} = \sum_{i \in L} \int_0^\infty \overline{y}_i(w)\,\mathrm{d}w.$$

Lemma 4.1. *The primal objective is lower bounded by the surrogate, that is, $\overline{P} \le P$.*

It will often be more convenient to consider the following characterization of $\overline{y}_i(w)$:

- Initially, let $\overline{y}_i(w) = 0$.
- If i is matched in a deterministic round in which its edge weight is at least w, set $\overline{y}_i(w) = 1$.
- If i is chosen in a randomized round in which its edge weight is at least w, further consider w', its edge weight in the previous round involving i; let $w' = 0$ if it is the first randomized round involving i. Then, decrease the gap $1 - \overline{y}_i(w)$ by a $1/2(1-\gamma)$ factor if $w' \ge w$, that is, if it is the second or later pair of a collection of consecutive pairs containing i with edge weight at least w; otherwise, just decrease $1 - \overline{y}_i(w)$ by $1/2$. The missing $1-\gamma$ factor in the second case corresponds to the -1 in the exponent of $1-\gamma$ in Equation (7).

Lemma 4.2. *For any offline vertex i and any weight level $w > 0$, we have that*

$$1 - \overline{y}_i(w) \ge 2^{-k_i(w)} \left(1-\gamma\right)^{\max\{k_i(w)-1,0\}}.$$

Proof. Initially, $1 - \overline{y}_i(w)$ equals 1. Then, by Equation (7), it decreases by $1/2$ in the first randomized round involving i with edge weight at least w and by at most $1/2\,(1-\gamma)$ in each of the subsequent $k_i(w) - 1$ rounds. □

Recall that $\overline{y}_i(w)$ increases to 1 in a deterministic round; Lemma 4.2 gives a lower bound for this increment.

Lemma 4.3. *For any offline vertex i and any weight level $w > 0$, if i is matched in a deterministic round in which its edge weight is at least w, the increment in $\overline{y}_i(w)$ is at least*

$$2^{-k_i(w)} \left(1-\gamma\right)^{\max\{k_i(w)-1,0\}}.$$

Lemma 4.4. *For any offline vertex i and any weight level $w > 0$, if i is chosen as a candidate in a randomized round in which its edge weight is at least w, the increment in $\overline{y}_i(w)$ is at least*

$$2^{-k_i(w)-1} \left(1-\gamma\right)^{\max\{k_i(w)-1,0\}}.$$

Suppose further that vertex i's edge weight is also at least w in the last randomized round involving i. Then, it follows that $k_i(w) \geq 1$ and the increment in $\overline{y}_i(w)$ is at least

$$2^{-k_i(w)-1}\left(1-\gamma\right)^{k_i(w)-1}\left(1+\gamma\right).$$

Proof. By definition, $1 - \overline{y}_i(w)$ decreases by a factor of either $^1\!/_2(1-\gamma)$ or $^1\!/_2$ in a randomized round, depending on whether vertex i's edge weight is at least w the last time it is chosen in a randomized round. Therefore, the increment in $\overline{y}_i(w)$ is either a $^1\!/_2(1+\gamma)$ fraction of $1-\overline{y}_i(w)$ or a $^1\!/_2$ fraction. Putting this together with the lower bound for $1-\overline{y}_i(w)$ in Lemma 4.2 proves the lemma. □

Dual Updates to Online Vertices. Consider any online vertex $j \in R$ at the time of its arrival. The dual variable β_j will only increase at the end of this round depending on the type of assignment. If j is left unmatched, then the value of β_j remains zero. If j is matched in a randomized round, set $\beta_j = \Delta^R_{i_1}\beta_j + \Delta^R_{i_2}\beta_j$. Last, if j is matched in a deterministic round, set $\beta_j = \Delta^D_{i^*}\beta_j$.

Dual Updates to Offline Vertices: Proof of Equation (4). Fix any offline vertex $i \in L$. Suppose that i is matched in a *deterministic round* in which its edge weight is w_{ij}. Then, for any weight level $w > w_{ij}$, the value of $k_i(w)$ stays the same. Thus, we leave $\alpha_i(w)$ unchanged. On the other hand, for any weight level $w \leq w_{ij}$, the value of $k_i(w)$ becomes ∞ by definition. Therefore, to maintain the invariant in Equation (4), we increase $\alpha_i(w)$ for each weight level $w \leq w_{ij}$ by

$$\sum_{\ell=k_i(w)}^{\infty} a(\ell). \tag{8}$$

The updates in *randomized rounds* are more subtle. Suppose that i is one of the two candidates in a randomized round in which its edge weight is w_{ij}. Further, consider i's edge weight the last time it was chosen in a randomized round, denoted as w'; let $w' = 0$ if this is the first randomized round involving vertex i. Then, w_{ij} and w' partition the weight levels $w > 0$ into up to three subsets, each of which requires a different update rule for $\alpha_i(w)$. Concretely, the algorithm increases $\alpha_i(w)$ by

$$\begin{cases} a\left(k_i(w)\right) & \text{if } 0 < w \leq w_{ij}\text{, and either } w \leq w' \text{ or } k_i(w) = 0; \\ a\left(k_i(w)\right) - 2^{-k_i(w)-1}\left(1-\gamma\right)^{k_i(w)-1}\gamma & \text{if } w' < w \leq w_{ij} \text{ and } k_i(w) \geq 1; \\ 2^{-k_i(w)-1}\left(1-\gamma\right)^{k_i(w)-1}\gamma & \text{if } w > w_{ij} \text{ and } k_i(w) \geq 1. \end{cases} \tag{9}$$

The first case is straightforward—we simply increase $\alpha_i(w)$ by $a\left(k_i(w)\right)$ to maintain the invariant in Equation (4). Observe that this is the only case in the unweighted problem.

For a weight level w that falls into the second case (if there is any), the increment in $\alpha_i(w)$ is smaller than the first case by $2^{-k_i(w)-1}(1-\gamma)^{k_i(w)-1}\gamma$. This is the difference between the lower bounds for the increments in $\overline{y}_i(w)$ in Lemma 4.4 depending on whether i's edge weight was at least w the last time it was chosen in a randomized round. Since the increase in the surrogate primal objective $\overline{\mathrm{P}}$ due to vertex i and weight level w (when $w' < w$) is less than the first case of Equation (9), we subtract this difference from the increment in $\alpha_i(w)$ so that the update to β_j is unaffected.

How can we still maintain the invariant in Equation (4) given the subtraction in the second case? Observe that if the second case happens, the same weight level must fall into the third case in the previous randomized round involving i. Thus, an equal amount is prepaid to each $\alpha_i(w)$ in the previous round. This give-and-take in the offline dual vertex updates becomes clear when we prove reverse weak duality in the next subsection.

4.2 Online Primal-Dual Analysis: Reverse Weak Duality

This subsection derives a set of sufficient conditions under which the increment in the surrogate primal $\overline{\mathrm{P}}$ is at least that of the dual solution D. Reverse weak duality then follows from $\mathrm{P} \geq \overline{\mathrm{P}} \geq \mathrm{D}$.

Deterministic Rounds. Suppose that j is matched to i in a deterministic round. Using the lower bound for the increase of $\overline{\mathrm{P}}$ in Lemma 4.3, the increase of the $\alpha_i(w)$s in Equation (8) and an upper bound for β_j by dropping the second term in Equation (6), we need

$$\int_0^{w_{ij}} \sum_{\ell=k_i(w)}^{\infty} a(\ell)\,\mathrm{d}w + \kappa \int_0^{w_{ij}} b\left(k_i(w)\right)\mathrm{d}w \leq \int_0^{w_{ij}} 2^{-k_i(w)}\left(1-\gamma\right)^{\max\{k_i(w)-1,0\}}\,\mathrm{d}w.$$

We will ensure the inequality locally at every weight level; thus, it suffices to have

$$\forall k \geq 0 \quad : \quad \sum_{\ell=k}^{\infty} a(\ell) + \kappa \cdot b(k) \leq 2^{-k}\left(1-\gamma\right)^{\max\{k-1,0\}}. \tag{10}$$

Randomized Rounds. Now, suppose that j is matched with candidates i_1, i_2 in a randomized round. We show that the increment in $\overline{\mathrm{P}}$ due to i_1 is at least the increase in the $\alpha_{i_1}(w)$s plus its contribution to β_j (i.e., $\Delta^R_{i_1}\beta_j$). This also holds for i_2 by symmetry; together, they prove reverse weak duality.

Let w_1 be the edge weight of $i \leftarrow i_1$ in this round, and let w'_1 be its edge weight the last time it was chosen in a randomized round. Set $w'_1 = 0$ if this has not happened. Then, w_1 and w'_1 partition the weight levels $w > 0$ into three subsets corresponding to the three cases for incrementing the dual variables $\alpha_i(w)$ in a randomized round, as in Equation (9).

The *first case* is when $0 < w \leq w_{ij}$, and either $w \leq w'$ or $k_i(w) = 0$. By Lemma 4.4, the increase in $\overline{\mathrm{P}}$ due to vertex i at weight level w is at least

$$\begin{cases} \frac{1}{2} & \text{if } k_i(w) = 0; \\ 2^{-k_i(w)-1}\left(1-\gamma\right)^{k_i(w)-1}\left(1+\gamma\right) & \text{if } k_i(w) \geq 1 \text{ and } w \leq \min\{w_1, w'_1\}. \end{cases}$$

By the first case of Equation (9), the increase in $\alpha_i(w)$ is $a(k_i(w))$. Finally, the contribution to the first term of $\beta_j = \Delta^R_i\beta_j + \Delta^R_{i_2}\beta_j$, at weight level w, in Equation (5) is $b(k_i(w))$. Hence, it suffices to ensure that

$$a(0) + b(0) \leq \frac{1}{2} \quad \text{and} \quad \forall k \geq 1 : a(k) + b(k) \leq 2^{-k-1}\left(1-\gamma\right)^{k-1}\left(1+\gamma\right). \tag{11}$$

The *second case* is when $w'_1 < w \leq w_1$ and $k_i(w) \geq 1$. By Lemma 4.4, the increment in $\overline{\mathrm{P}}$ due to i at weight level w is at least $2^{-k_i(w)-1}(1-\gamma)^{k_i(w)-1}$. By the second case of Equation (9), the increase in $\alpha_i(w)$ is $a(k_i(w)) - 2^{-k_i(w)-1}(1-\gamma)^{k_i(w)-1}\gamma$. Finally, the contribution to the first term of β_j, at weight level w, is $b(k_i(w))$. Hence, we need

$$a\left(k_i(w)\right) - 2^{-k_i(w)-1}\left(1-\gamma\right)^{k_i(w)-1}\gamma + b\left(k_i(w)\right) \leq 2^{-k_i(w)-1}\left(1-\gamma\right)^{k_i(w)-1}.$$

Rearranging the second term to the RHS gives us the same conditions as the second part of Equation (11).

The *third case* is when $w > w_1$ and $k_i(w) \geq 1$. The increment in $\overline{\mathrm{P}}$ due to i at weight level w is 0. By the last case of Equation (9), the increase in $\alpha_i(w)$ is $2^{-k_i(w)-1}(1-\gamma)^{k_i(w)-1}\gamma$. The negative contribution from the second term of β_j, at weight level w, is $\frac{1}{2}\sum_{0\le\ell<k_i(w)} a(\ell)$. Hence, we need

$$2^{-k_i(w)-1}\left(1-\gamma\right)^{k_i(w)-1}\gamma - \frac{1}{2}\sum_{0\le\ell<k_i(w)} a(\ell) \le 0.$$

The first term is decreasing in $k_i(w)$ and the second is increasing (in absolute value). Thus, it suffices to consider $k_i(w) = 1$:

$$a(0) \ge \frac{\gamma}{2}. \tag{12}$$

4.3 Online Primal-Dual Analysis: Approximate Dual Feasibility

This subsection derives a set of conditions that are sufficient for approximate dual feasibility, that is, Equation (3). Start by fixing any $i \in L$ and any $j \in R$, as well as the values of the $k_i(w)$s when j arrives.

Boundary Condition at the Limit. First, it may be the case that $k_i(w) = \infty$ for all $0 < w \le w_{ij}$ and j is unmatched. This means that $\beta_j = 0$ in this round. Thus, the contribution from the $\alpha_i(w)$s alone must ensure approximate dual feasibility. To do so, we will ensure that the value of $\alpha_i(w)$ is at least Γ whenever $k_i(w) = \infty$. By the invariant in Equation (4), it suffices to have

$$\sum_{\ell=0}^{\infty} a(\ell) \ge \Gamma. \tag{13}$$

Next, we consider five different cases that depend on whether the round of j is randomized, deterministic, or unmatched, and if i is chosen as a candidate. We first analyze the cases in which j is in a randomized round. Then, we show that the other cases require only weaker conditions.

Case 1: Round of j is randomized, i is not chosen. By definition, $\beta_j = \Delta_{i_1}^R\beta_j + \Delta_{i_2}^R\beta_j$. Since i is not chosen, both terms on the RHS are at least $\Delta_i^R\beta_j$. Using the definition of $\Delta_i^R\beta_j$ in Equation (5) and lower bounding $\alpha_i(w)$ by Equation (4), approximate dual feasibility in Equation (3) reduces to

$$\int_0^{w_{ij}} \sum_{0\le\ell<k_i(w)} a(\ell)\,\mathrm{d}w + 2\int_0^{w_{ij}} b\left(k_i(w)\right)\mathrm{d}w \ge \Gamma\cdot w_{ij}.$$

We will again ensure this inequality at every weight level. Therefore, it suffices to have

$$\forall k \ge 0 \quad : \quad \sum_{0\le\ell<k} a(\ell) + 2\cdot b(k) \ge \Gamma. \tag{14}$$

Case 2: Round of j is randomized, i is chosen. By symmetry, suppose without loss of generality that $i \leftarrow i_1$ and i_2 is the other candidate. By definition, $\beta_j = \Delta_i^R\beta_j + \Delta_{i_2}^R\beta_j$. Next, we derive a lower bound only in terms of $\Delta_i^R\beta_j$. Since the algorithm does not choose a deterministic round with i alone, we have that $\Delta_i^R\beta_j + \Delta_{i_2}^R\beta_j \ge \Delta_i^D\beta_j$. Further, we have that $\Delta_i^D\beta_j = \kappa\cdot\Delta_i^R\beta_j$ by Equation (6). Combining these, we have that $\beta_j \ge \kappa\cdot\Delta_i^R\beta_j$. Finally, by the definition of $\Delta_i^R\beta_j$ in Equation (5), β_j is at least

$$\kappa\cdot\left(\int_0^{w_{ij}} b\Big(k_i(w)\Big)\,\mathrm{d}w - \frac{1}{2}\int_{w_{ij}}^{\infty}\sum_{0\le\ell<k_i(w)} a(\ell)\,\mathrm{d}w\right).$$

Lower bounding the $\alpha_i(w)$s is more subtle. Recall that $k_i(w)$ denotes the value at the beginning of the round when j arrives. Thus, the value of $k_i(w)$ increases by 1 for any weight level $0 < w \le w_{ij}$

and stays the same for any other weight level $w > w_{ij}$. Therefore, the contribution of the $\alpha_i(w)$s to approximate dual feasibility is at least

$$\int_0^{w_{ij}} \sum_{0 \le \ell \le k_i(w)} a(\ell)\,\mathrm{d}w + \int_{w_{ij}}^{\infty} \sum_{0 \le \ell < k_i(w)} a(\ell)\,\mathrm{d}w.$$

Finally, since $\kappa < 2$, the net contribution from weight levels $w > w_{ij}$ is nonnegative; thus, we can drop them. Then, approximate dual feasibility as in Equation (3) becomes

$$\int_0^{w_{ij}} \left(\sum_{0 \le \ell \le k_i(w)} a(\ell) + \kappa \cdot b\left(k_i(w)\right) \right) \mathrm{d}w \ge \Gamma \cdot w_{ij}.$$

Thus, it suffices to ensure the inequality locally at every weight level:

$$\forall k \ge 0 \quad : \qquad \sum_{0 \le \ell \le k} a(\ell) + \kappa \cdot b(k) \ge \Gamma. \tag{15}$$

There are two differences between Equations (14) and (15). First, the summation above includes $\ell = k$. We can do this because i is one of the two candidates and, therefore, $k_i(w)$ increases by 1 in the round of j for any weight level $w \le w_{ij}$. Second, the κ coefficient for the second term is smaller.

Case 3: Round of j is deterministic, i is not chosen. By definition, $\beta_j = \Delta^D_{i^*}\beta_j$. Next, we derive a lower bound in terms of $\Delta^R_i\beta_j$. Since the algorithm does not choose a randomized round with i and i^* as the two candidates, we have that $\Delta^D_{i^*}\beta_j > \Delta^R_{i^*}\beta_j + \Delta^R_i\beta_j$. By Equation (6) and $\kappa < 2$, we have that $\Delta^R_{i^*}\beta_j > \frac{1}{2} \cdot \Delta^D_{i^*}\beta_j$. Here, we use the fact that $\Delta^D_{i^*}\beta_j \ge 0$, because i^* is chosen in a deterministic round. Putting this together gives us $\beta_j = \Delta^D_{i^*}\beta_j > 2 \cdot \Delta^R_i\beta_j$, which is identical to the lower bound in the first case. Therefore, approximate dual feasibility is guaranteed by Equation (14).

Case 4: Round of j is deterministic, i is chosen. For any $0 < w \le w_{ij}$, we have that $k_i(w) = \infty$ after this round. Therefore, approximate dual feasibility follows from the contribution of the $\alpha_i(w)$s alone due to the invariant in Equation (4) and the boundary condition in Equation (13).

Case 5: Round of j is unmatched. By definition, $\beta_j = 0$. Moreover, $\Delta^D_i\beta_j < 0$ since the algorithm chooses to leave j unmatched, which further implies that $\Delta^R_i\beta_j < 0$ by Equation (6). Therefore, we have that $\beta_j \ge 2 \cdot \Delta^R_i\beta_j$, which is identical to the lower bound in the first case. Thus, approximate dual feasibility is guaranteed by Equation (14).

4.4 Optimizing the Gain-Sharing Parameters

To optimize the competitive ratio Γ in the above online primal-dual analysis, it remains to solve for the gain sharing parameters $a(k)$ and $b(k)$ using the following LP:

$$\begin{aligned} &\text{maximize} && \Gamma \\ &\text{subject to} && \text{Equations (10)–(15)} \end{aligned}$$

We obtain a lower bound on the competitive ratio by solving a more restricted LP, which is finite. In particular, we set $a(k) = b(k) = 0$ for all $k > k_{\max}$ for some sufficiently large integer $k_{\max}$ so

Table 1. Approximately Optimal Solutions to the Factor-Revealing LP with $\kappa = 3/2$ and $k_{\max} = 8$

(a) $\gamma = 1/16$, $\Gamma = 0.50503484$

k	$a(k)$	$b(k)$
0	0.24748256	0.25251744
1	0.13684883	0.12877617
2	0.06415997	0.06035174
3	0.03009310	0.02827176
4	0.01413332	0.01322521
5	0.00666576	0.00615855
6	0.00318572	0.00282566
7	0.00158503	0.00123280
8	0.00088057	0.00044028

(b) $\gamma = \frac{13\sqrt{13}-35}{108} \approx 0.109927$, $\Gamma = 0.508672$

k	$a(k)$	$b(k)$
0	0.24566361	0.25433639
1	0.14597716	0.13150459
2	0.06497349	0.05851601
3	0.02892807	0.02602926
4	0.01289279	0.01156523
5	0.00576587	0.00511883
6	0.00260819	0.00223589
7	0.00122399	0.00093180
8	0.00063960	0.00031980

that it becomes

$$
\begin{aligned}
\text{maximize} \quad & \Gamma \\
\text{subject to} \quad & \sum_{k \le \ell \le k_{\max}} a(\ell) + \kappa \cdot b(k) \le 2^{-k} (1-\gamma)^{\max\{k-1,0\}} && \forall\, 0 \le k \le k_{\max} \\
& a(0) + b(0) \le \frac{1}{2} \\
& a(k) + b(k) \le 2^{-k-1} (1-\gamma)^{k-1} (1+\gamma) && \forall\, 1 \le k \le k_{\max} \\
& a(0) \ge \frac{\gamma}{2} \\
& \sum_{0 \le \ell \le k_{\max}} a(\ell) \ge \Gamma \\
& \sum_{0 \le \ell < k} a(\ell) + 2 \cdot b(k) \ge \Gamma && \forall\, 0 \le k \le k_{\max} \\
& \sum_{0 \le \ell \le k} a(\ell) + \kappa \cdot b(k) \ge \Gamma && \forall\, 0 \le k \le k_{\max} \\
& a(k), b(k) \ge 0 && \forall\, 0 \le k \le k_{\max}
\end{aligned}
$$

We present an approximate solution to the finite LP in Table 1(a) with $\gamma = 1/16$, $\kappa = 3/2$, and $k_{\max} = 8$, which gives $\Gamma > 0.505$. We also tried different values of $\kappa = 1 + \ell/16$, for $0 \le \ell \le 16$. If $\kappa = 1$ or $\kappa = 2$, then $\Gamma = 0.5$; if $\kappa = 1 + 15/16$, then $\Gamma \approx 0.5026$; for all other values of κ that are multiples of $1/16$, we have $\Gamma > 0.505$. Hence, the analysis is robust to the choice of κ so long as it is neither too close to 1 nor to 2. Furthermore, even for $k_{\max}$ as small as 3 (and $\gamma = \frac{13\sqrt{13}-35}{108}$), we get a competitive ratio $\Gamma > 0.504$, improving upon greedy. Table 1(b) presents an approximately optimal solution under the same setting except that we use a larger $\gamma = \frac{13\sqrt{13}-35}{108} > 0.1099$ as in Theorem 3.3, which leads to the improved competitive ratio $\Gamma > 0.5086$.[2]

[2] The source code is available at https://github.com/fahrbach/edge-weighted-online-bipartite-matching.

ALGORITHM 2: Online Correlated Selection (OCS)

```
State variables:
• $\tau_i \in \{\textit{selected, not selected, unknown}\}$ for each ground element $i$; initially, let $\tau_i = \textit{unknown}$.
On receiving a pair of elements $i_1$ and $i_2$:
(1) With probability 1/2:                                                    (sender)
  (a) Draw $\ell, m \in \{1, 2\}$ uniformly at random.
  (b) Let $\tau_{i_{-m}} = \textit{unknown}$.
  (c) If $m = \ell$, let $\tau_{i_m} = \textit{selected}$; otherwise, let $\tau_{i_m} = \textit{not selected}$.
(2) Otherwise (i.e., with probability 1/2):                                  (receiver)
  (a) Draw $m \in \{1, 2\}$ uniformly at random.
  (b) If $\tau_{i_m} = \textit{selected}$, let $\ell = -m$;
      if $\tau_{i_m} = \textit{not selected}$, let $\ell = m$;
      if $\tau_{i_m} = \textit{unknown}$, draw $\ell \in \{1, 2\}$ uniformly at random.
  (c) Let $\tau_{i_1} = \tau_{i_2} = \textit{unknown}$.
(3) Select $i_\ell$.
```

5 ONLINE CORRELATED SELECTION: IN DETAIL

This section provides the formal description and analysis of the OCS used in Section 4. Section 5.1 introduces the basics of OCS with the proof of a 1/16-OCS, substantiating the sketch in Section 3. Section 5.2 then shows how to improve the design and analysis of the OCS to prove Theorem 3.3.

5.1 Warmup: Constructing a 1/16-OCS

Algorithm 2 presents the 1/16-OCS. It maintains a state variable τ_i for each element i. If the state τ_i equals *selected* or *not selected*, it reflects the selection in the last pair involving i and indicates that this information can be used in the next pair involving i. If the state τ_i is *unknown*, it means that the past selection result of element i cannot be used to determine the selections in future pairs.

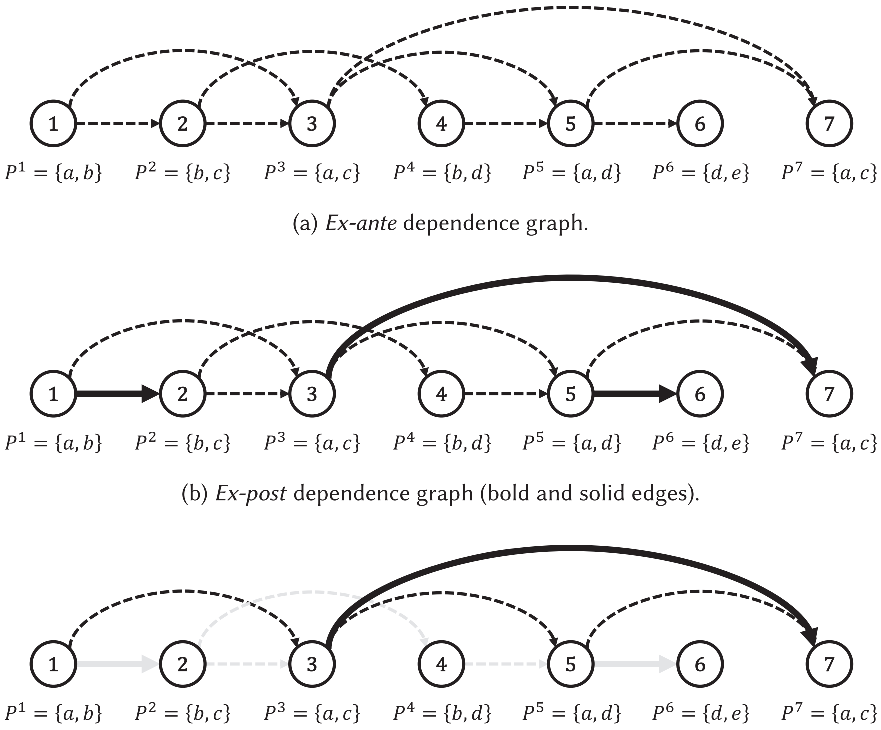
For each pair of elements i_1 and i_2 in the sequence, the OCS first decides whether this pair is a *sender* or a *receiver* uniformly at random. If it is a sender, use a fresh random bit to select i_ℓ, $\ell \in \{1, 2\}$, for this pair. Then, draw $m \in \{1, 2\}$ uniformly at random and set τ_{i_m} to reflect the selection in this round; set $\tau_{i_{-m}}$ to be *unknown*, where $-m$ is an abbreviation for $3 - m$. That is, the OCS forwards the random selection in this round to subsequent rounds for only one of the two elements in the current pair, chosen uniformly at random.

If it is a receiver, on the other hand, the OCS seeks to use the previous selection result of the elements to determine its choice of i_ℓ. First, it draws $m \in \{1, 2\}$ uniformly at random and checks the state variable of i_m. To achieve negative correlation, the OCS makes the opposite selection in this round whenever possible. If the state is *selected*, indicating that i_m is selected in the last pair involving it, the OCS selects i_{-m} this time, and vice versa. If the state variable equals *unknown*, the OCS uses a fresh random bit to select i_ℓ. In either case, reset the states of i_1 and i_2 to be *unknown*.

In fact, we will show a result stronger than the definition of 1/16-OCS.

Lemma 5.1. *For any fixed sequence of pairs of elements, any fixed element i, and any integer $k \geq 0$, Algorithm 2 ensures that after appearing in a collection of k consecutive pairs, i is selected at least once with probability at least $1 - 2^{-k} \cdot f_k$, where f_k is defined recursively as*

$$f_k = \begin{cases} 1 & \text{if } k = 0, 1; \\ f_{k-1} - \frac{1}{16} f_{k-2} & \text{if } k \geq 2. \end{cases} \tag{16}$$



(a) *Ex-ante* dependence graph.

(b) *Ex-post* dependence graph (bold and solid edges).

(c) Dependence subgraph associated with a fixed element a.

Fig. 2. Example of dependence graphs with five ground elements and a sequence of seven pairs.

Lemma 5.1 implies that Algorithm 2 is a 1/16-semi-OCS by considering the subsequence of all pairs involving element i because

$$f_k = f_{k-1} - \frac{1}{16} f_{k-2} \leq \left(1 - \frac{1}{16}\right) f_{k-1} \leq \left(1 - \frac{1}{16}\right)^{k-1} f_1 = \left(1 - \frac{1}{16}\right)^{k-1}.$$

Let $P^1 = \{i_1^1, i_2^1\}, P^2 = \{i_1^2, i_2^2\}, \ldots, P^n = \{i_1^n, i_2^n\}$ be the sequence of pairs of ground elements. We start with a graph-theoretic interpretation of the OCS algorithm.

Ex-ante Dependence Graph. Consider a graph $G^{\textit{ex-ante}} = (V, E^{\textit{ex-ante}})$ as follows, which we shall refer to as the *ex-ante dependence graph*. To make a distinction with the vertices and edges in the online matching problem, we shall refer to the vertices and edges in the dependence graph as *nodes* and *arcs*, respectively. Let there be a node for each pair of elements in the collection. We will refer to them as $1 \leq j \leq n$, that is,

$$V = \left\{ j \in \mathbb{Z} : 1 \leq j \leq n \right\}.$$

Further, for any fixed element i in the ground set, let there be a directed arc from j_1 to j_2 for any two consecutive pairs $j_1 < j_2$ involving i, that is,

$$E^{\textit{ex-ante}} = \left\{ (j_1, j_2)_i : j_1 < j_2 \text{ s.t. } i \in P^{j_1}, i \in P^{j_2}, \text{ and } \forall j_1 < t < j_2, i \notin P^t \right\}.$$

The subscript i helps to distinguish parallel arcs when the pairs j_1 and j_2 have the same two elements. See Figure 2(a) for an illustrative example of the ex-ante dependence graph.

Each arc in the *ex-ante* dependence graph represents two pairs in the sequence in which the OCS could use the same random bit to select oppositely. By construction, there are at most two outgoing arcs and at most two incoming arcs for each node.

In particular, consider any arc $(j_1, j_2)_i$ in the *ex-ante* dependence graph, with i being the common element. If the randomness used by the OCS satisfies (1) pair j_1 is a sender, (2) $i_m = i$ in pair j_1, (3) pair j_2 is a receiver, and (4) $i_m = i$ in pair j_2, the selections in the two pairs would be perfectly negatively correlated in the sense that i is selected in exactly one of the two pairs. Each of these four events happens independently with probability $1/2$. Hence, we achieve this perfect negative correlation with probability $1/16$.

Ex-post Dependence Graph. The *ex-post* dependence graph $G^{ex\text{-}post} = (V, E^{ex\text{-}post})$ is a subgraph of the *ex-ante* dependence graph that keeps the arcs corresponding to pairs that are perfectly negatively correlated given the realization of whether each step is a sender or a receiver and the value of m therein. Equivalently, the *ex-post* dependence graph is realized as follows. Over the randomness with which the OCS decides whether each step is a sender or a receiver, and the values of m, each node in the *ex-ante* dependence graph effectively picks at most one of its incident arcs, each with probability $1/4$. An arc is realized in the *ex-post* graph if both incident nodes choose it. With this interpretation, we get that the *ex-post* graph is a matching. The OCS may be viewed as a randomized online algorithm that picks a matching in the *ex-ante* graph such that each arc in the *ex-ante* graph is chosen with probability lower bounded by a constant. See Figure 2(b) for an example.

Proof of Lemma 5.1. Let $j_1 < j_2 < \dots$ be the pairs involving a ground element i. We will use the element a and $k = 4$ in Figure 2(c) as a running example, where $j_1 = 1, j_2 = 3, j_3 = 5, j_4 = 7$ and the relevant arcs in the dependence graphs are $(1, 3)_a$, $(3, 5)_a$, $(3, 7)_c$, and $(5, 7)_a$.

If at least one of the arcs among $j_1 < j_2 < \cdots < j_k$ is realized in the *ex-post* dependence graph, element i must be selected at least once. This is because the randomness (related to the choice of ℓ in the OCS) is perfectly negatively correlated in the two incident nodes of the arc and thus, i is selected exactly once in these two steps. Importantly, this is true even if the arc is not due to element i. For example, given that the arc $(3, 7)_c$ is realized in Figure 2(c), element a must be selected at least once after step 7.

On the other hand, if none of these arcs is realized, then the random bits used in the k steps $j_1 < j_2 < \cdots < j_k$ are independent. For example, consider element a and $k = 3$ in Figure 2(c). Element a is selected independently with probability $1/2$ in steps $j_1 = 1$, $j_2 = 3$, and $j_3 = 5$ given that neither $(1, 3)_a$ nor $(3, 5)_a$ is realized.

Importantly, even if some of these pairs are receivers in that the selections therein are based on the random bits realized earlier by some senders, from i's point of view, they are still independent of the randomness in the other rounds that i is involved in. For example, from c's point of view in Figure 2(c), even though the selection in step 2 is determined by the selection in step 1, it is independent of the selections in steps 3 and 7, which involve c.

Putting this together, the probability that i is *never selected* after steps $j_1 < j_2 < \cdots < j_k$ is equal to the probability that (1) none of the arcs among these steps is realized, times the probability that (2) none of the k independent random selections picks i. This follows from the law of total probability. The latter quantity equals 2^{-k}; thus, it remains to analyze the former. We shall upper bound it by the probability that none of the arcs $(j_1, j_2)_i, (j_2, j_3)_i, \dots, (j_{k-1}, j_k)_i$ is realized. We shall omit the subscript i in the rest of the proof for brevity. Denote this event as F_k and its probability as f_k.

Trivially, we have that $f_0 = f_1 = 1$. To prove that the recurrence in Equation (16) governs f_k, further divide event F_k into two subevents. Let A_k be the event that none of the arcs

$(j_1, j_2), (j_2, j_3), \dots, (j_{k-1}, j_k)$ is realized, and node j_k picks arc (j_k, j_{k+1}) in realizing the *ex-post* dependence graph. Let B_k be the event that none of the arcs is realized and node j_k does not pick arc (j_k, j_{k+1}). Let a_k and b_k be the probability of A_k and B_k, respectively. We have that A_k and B_k form a partition of F_k and. thus,

$$f_k = a_k + b_k.$$

If node j_k picks arc (j_k, j_{k+1}), which happens with probability $1/4$, arc (j_{k-1}, j_k) is not realized by definition regardless of the remaining randomness. Therefore, conditioned on the choice of j_k, subevent A_k happens if and only if the choices made by steps $j_1, j_2, \dots, j_{k-1}$ is such that none of $(j_1, j_2), \dots, (j_{k-2}, j_{k-1})$ is realized, ithat is, when F_{k-1} happens. That is,

$$a_k = \frac{1}{4} f_{k-1}.$$

On the other hand, if pair j_k does not pick (j_k, j_{k+1}), there are two possibilities. The first case is when j_k picks (j_{k-1}, j_k), which happens with probability $1/4$. In this case, the choices made by $j_1, \dots, j_{k-1}$ must be such that none of $(j_1, j_2), \dots, (j_{k-2}, j_{k-1})$ is realized, and j_{k-1} does *not* pick (j_{k-1}, j_k), that is, B_{k-1} happens. The second case is when j_k picks neither (j_{k-1}, j_k) nor (j_k, j_{k+1}), which happens with probability $1/2$. In this case, the choices made by $j_1, \dots, j_{k-1}$ must be such that none of $(j_1, j_2), \dots, (j_{k-2}, j_{k-1})$ is realized, that is, F_{k-1} happens. Putting this together, we have that

$$b_k = \frac{1}{4} b_{k-1} + \frac{1}{2} f_{k-1}.$$

Eliminating a_ks and b_ks with the above three equations, we get the recurrence in Equation (16). □

The same analysis generalizes to prove a stronger result, which implies a $1/16$-OCS.

Lemma 5.2. *For any fixed sequence of pairs of elements, any fixed element i, and any disjoint collections of $k_1, k_2, \dots, k_m$ consecutive pairs involving i, Algorithm 2 ensures that i is selected in at least one of these pairs with probability at least*

$$1 - \prod_{\ell=1}^{m} 2^{-k_\ell} \cdot f_{k_\ell}.$$

Proof. Let $j_1^\ell < j_2^\ell < \cdots < j_{k_\ell}^\ell$ be the ℓ-th subsequence of consecutive pairs involving element i for any $1 \le \ell \le m$. The probability that i is never selected is equal to (1) the probability that none of the arcs among the steps in these collections is realized, times (2) the probability that all $\sum_{\ell=1}^{m} k_\ell$ random bits are against i. The latter is $\prod_{\ell=1}^{m} 2^{-k_\ell}$. We upper bound the former with the probability that for any $1 \le \ell \le m$, none of the arcs $(j_1^\ell, j_2^\ell), \dots, (j_{k_\ell-1}^\ell j_{k_\ell}^\ell)$ is realized. Finally, observe that the events are independent for different collections ℓ because the event concerning each collection only relies on the randomness of the nodes in the collection. Hence, it is at most $\prod_{\ell=1}^{m} f_{k_\ell}$. □

5.2 Optimizing the OCS: Proof of Theorem 3.3

Similar to the warmup algorithm, we will realize the *ex-post* dependence graph by letting each node be either a *sender* or a *receiver* independently and randomly. The probability of letting a node be a sender, denoted as p, will be optimized later.

A sender uses a fresh random bit to select an element from the corresponding pair. Further, it randomly picks an out-arc in the *ex-post* graph and sends its selection along the out-arc. Although the out-neighbors and out-arcs have yet to arrive, we can refer to them as the one due to the first and second element in the current pair, respectively. This is identical to the warmup case.

A receiver, on the other hand, adapts to the information it receives and makes the opposite selection. The improved OCS proactively checks both in-arcs of a receiver; in contrast, the warmup

ALGORITHM 3: Improved Online Correlated Selection

```
Parameter:
• $p$: probability that a node is a sender.
State variables:
• $G^{ex\text{-}ante} = (V, E^{ex\text{-}ante})$: ex-ante dependence graph; initially, $V = E^{ex\text{-}ante} = \emptyset$.
• $G^{ex\text{-}post} = (V, E^{ex\text{-}post})$: ex-post dependence graph; initially, $V = E^{ex\text{-}post} = \emptyset$.
• $\tau_j \in \{sender, receiver\}$ for any $j \in V$.
On receiving a pair $j$ of elements $i_1$ and $i_2$:
(1) Add $j$ to $V$.
(2) For $k \in \{1, 2\}$, let $j_k$ be the last pair which involves $i_k$; add an arc $(j_k, j)_{i_k}$ to $E^{ex\text{-}ante}$.
(3) With probability $p$:                                                    (sender)
  (a) Let $i^* = i_1$ or $i_2$, each with probability 1/2.
  (b) Pick an out-arc randomly.
(4) Otherwise, that is, with probability $1 - p$:                            (receiver)
  (a) Pick a $j_m$, $m \in \{1, 2\}$, that is a sender who picks arc $(j_m, j)_{i_m}$ (break ties randomly):
     (i) Add arc $(j_m, j)_{i_m}$ to $E^{ex\text{-}post}$.
    (ii) Let $i^* = i_m$ if $i_m$ is not selected in round $j_m$; let $i^* = i_{-m}$ otherwise.
  (b) Otherwise, let $i^* = i_1$ or $i_2$, each with probability 1/2.
(5) Select $i^*$.
```

algorithm checks only one randomly chosen in-arc. Concretely, check both in-arcs in the *ex-ante* graph to see whether any in-neighbor is a sender who picks the arc between them. If both in-neighbors are senders and both pick the corresponding arcs, choose one randomly. Add the arc to the *ex-post* dependence graph. Supposethat a receiver j receives the selection by a sender j' sent along arc $(j', j)_i$. Then, select i in round j if it is not selected in round j', and vice versa. See Algorithm 3 for a formal definition of the improved OCS.

Lemma 5.3. *For any fixed sequence of pairs of elements, any fixed element i, and any disjoint subsequences of $k_1, k_2, \dots, k_m$ consecutive pairs involving i, Algorithm 3 ensures that i is selected in at least one of these pairs with probability at least*

$$1 - \prod_{\ell=1}^{m} 2^{-k_\ell} \cdot g_{k_\ell},$$

where g_k is defined recursively as follows:

$$g_k = \begin{cases} 1 & \textit{if } k = 0, 1; \\ g_{k-1} - \frac{1}{8} p\,(1-p)\,(4-p) \cdot g_{k-2} & \textit{if } k \geq 2. \end{cases} \tag{17}$$

Corollary 5.4. *Algorithm 3 is a $\frac{1}{8}p(1-p)(4-p)$-OCS.*

To prove Theorem 3.3, let $p = \frac{5-\sqrt{13}}{3}$ to maximize $\frac{1}{8}p\Big(1-p\Big)\Big(4-p\Big) = \frac{13\sqrt{13}-35}{108} > 0.1099$.

Proof of Lemma 5.3. Let $j_1^\ell < j_2^\ell < \cdots < j_{k_\ell}^\ell$, $1 \leq \ell \leq m$, be the subsequences of consecutive pairs that involve element i. The algorithm uses two kinds of independent random bits. The first kind is used to realize the *ex-post* dependence graph, that is, the random type of each pair, the random out-arc chosen by each sender, and the random in-neighbor of each receiver in the tie-breaking case. The second kind is the random selections by senders, and by receivers that fail to receive the selection of any sender. Importantly, the two kinds of randomness are independent.

Similar to the warmup case, we are interested in the event that there is no arc among these pairs in the *ex-post* dependence graph:

$$F = \{\text{nodes } j_s^\ell, \text{ for } 1 \le \ell \le m \text{ and } 1 \le s \le k_\ell, \text{ are disjoint in } G^{\textit{ex-post}}\}.$$

If there is an arc between two pairs in the collections in the *ex-post* dependence graph, i is selected in exactly one of the two pairs. Otherwise, the selections in these pairs are independent. Hence, the probability that i is never selected is equal to the product of (1) the probability that the $\sum_{\ell=1}^m k_\ell$ nodes in the collections are disjoint in the *ex-post* dependence graph, and (2) none of the $\sum_{\ell=1}^m k_\ell$ independent random selections picks i. This follows from the law of total probability. The former quantity is $\Pr(F)$, and the latter is equal to $2^{-\sum_{\ell=1}^m k_\ell}$. Putting this together, it equals

$$2^{-\sum_{1\le\ell\le m} k_\ell} \cdot \Pr(F).$$

Therefore, it remains to show that

$$\Pr(F) \le \prod_{\ell=1}^m g_{k_\ell}. \tag{18}$$

Which arcs are we concerned about in the event F? Since these are subsequences of consecutive pairs involving element i, the arcs of the form $(j_s^\ell, j_{s+1}^\ell)_i$ always exist in the *ex-ante* dependence graph. To characterize whether some of these arcs are realized in the *ex-post* graphs, we need to further consider another set of arcs as follows.

For any $1 \le \ell \le m$, consider the in-arcs of nodes $j_1^\ell < \cdots < j_{k_\ell}^\ell$ in the *ex-ante* dependence graph due to the element other than i. Let them be $(\hat{j}_s^\ell, j_s^\ell)$ for $1 \le s \le k_\ell$. We omit the subscript that denotes the common element in the two nodes, with the understanding that they are due to the element other than i in the round of j_s^ℓ. Then, an arc $(j_s^\ell, j_{s+1}^\ell)_i$ is realized in the *ex-post* graph if:

(1) Node j_s^ℓ is a sender that picks arc $(j_s^\ell, j_{s+1}^\ell)_i$;
(2) Node j_{s+1}^ℓ is a receiver;
(3) Either node $\hat{j}_{s+1}^\ell$ is a receiver, or it is a sender but does not choose arc $(\hat{j}_{s+1}^\ell, j_{s+1}^\ell)$, or the tie-breaking by node j_{s+1}^ℓ is in favor of j_s^ℓ.

Binding Case. First, suppose that all $\hat{j}_s^\ell$'s exist, and the j_s^ℓ's and $\hat{j}_s^\ell$'s are all distinct. It is relatively easy to analyze because, in this case, it suffices to consider arcs of the form $(j_s^\ell, j_{s+1}^\ell)_i$ and different subsequences of consecutive pairs depend on disjoint sets of random bits and, therefore, may be analyzed separately. This turns out to be the binding case of the analysis. We will analyze the binding case in Lemma 5.5 and show in Lemma 5.6 that this is the worst-case scenario that maximizes $\Pr(F)$.

Lemma 5.5. *In the binding case, the probability of event F is:*

$$\Pr(F) = \prod_{\ell=1}^m g_{k_\ell}.$$

Proof. We start by formalizing the aforementioned implications of the assumption that all j_s^ℓs and $\hat{j}_s^\ell$s are distinct. First, two pairs in the collections are connected if and only if they are consecutive pairs in the same collection, for example, j_{s-1}^ℓ and j_s^ℓ, and arc $(j_{s-1}^\ell, j_s^\ell)_i$ is realized. A pair j_s^ℓ with $s > 1$ cannot be the receiver of a sender other than j_{s-1}^ℓ in the collections because $\hat{j}_s^\ell$s are not in the collections by the assumption. Second, the realization of these arcs in different collections are independent. The realization of arcs of the form $(j_{s-1}^\ell, j_s^\ell)_i$, for any fixed collection ℓ, depends only on the realization of first kind of randomness related to nodes with superscript ℓ, that is, j_s^ℓs and $\hat{j}_s^\ell$s.

Next, we focus on a fixed subsequence ℓ and analyze the probability that no arc of the form $(j_s^\ell, j_{s+1}^\ell)_i$, for $1 \leq s < k_\ell$, is realized. To simplify notation, we omit the superscripts and subscripts ℓ and write $j_1 < j_2 < \cdots < j_k$ and $\hat{j}_2, \hat{j}_3, \ldots, \hat{j}_k$. Let G_k denote this event and g_k be its probability. Trivially, we have that $g_0 = g_1 = 1$. It remains to show that g_k follows the recurrence in Equation (17).

We will do so by further considering an auxiliary subevent A_k, which requires not only G_k to happen but also j_k to be a sender who picks the out-arc due to i. Let a_k denote its probability.

Auxiliary Event. If j_k is a sender who picks the out-arc due to i, which happens with probability $p/2$, arc $(j_{k-1}, j_k)_i$ would not be realized regardless of the randomness of the other nodes in the collection. Therefore, under this condition, event A_k reduces to event G_{k-1}.

$$a_k = \frac{p}{2} \cdot g_{k-1}.$$

Main Event. If j_k is a sender, which happens with probability p, arc $(j_{k-1}, j_k)_i$ would not be realized regardless of the randomness of the other nodes in the collection. Therefore, under this condition, event G_k reduces to event G_{k-1}. The contribution of this part to the probability of G_k is

$$p \cdot g_{k-1}.$$

If j_k is a receiver (probability $1 - p$) but $\hat{j}_k$ is a sender who picks arc $(\hat{j}_k, j_k)$ (probability $p/2$), and the tie-breaking at j_k is in favor of $\hat{j}_k$ (probability $1/2$), we still have that arc $(j_{k-1}, j_k)_i$ cannot be realized regardless of the randomness of the other nodes. The contribution of this part to the probability of G_k is

$$\frac{p(1-p)}{4} g_{k-1}.$$

Otherwise, j_{k-1} must not be a sender who picks arc $(j_{k-1}, j_k)_i$ or else arc $(j_{k-1}, j_k)_i$ would be realized. Therefore, conditioned on being in this case, events G_k reduces to event $G_{k-1} \setminus A_{k-1}$. The contribution of this part to the probability of G_k is

$$(1-p)\left(1 - \frac{p}{4}\right)(g_{k-1} - a_{k-1}).$$

Putting everything together, we have that

$$g_k = g_{k-1} - (1-p)\left(1 - \frac{p}{4}\right) a_{k-1}.$$

Eliminating a_ks by combining the two equations, we get the recurrence in Equation (17). □

Lemma 5.6. *The probability of event F is maximized in the binding case.*

Proof. Here are the possible violations of the conditions of the regular case:

(1) Some arc $(\hat{j}_s^\ell, j_s^\ell)$ may not exist, that is, the element other than i in pair j_s^ℓ has its first appearance in pair j_s^ℓ.
(2) There may be ℓ, ℓ', s, s' such that $\hat{j}_s^\ell = j_{s'}^{\ell'}$, that is, the element other than i in pair j_s^ℓ is also an element in pair $j_{s'}^{\ell'}$, and in no other pairs in between.
(3) There may be ℓ, ℓ', s, s' such that $\hat{j}_s^\ell = \hat{j}_{s'}^{\ell'}$.

We use a coupling argument to compare the probability of event F in a general case, potentially with some of these violations, with the probability in the binding case.

Type 1 Violation. Consider an instance almost identical to the one at hand except that we introduce a new node $\hat{j}^\ell_s$ for such a violation. For example, let pair $\hat{j}^\ell_s$ be at the beginning of the sequence, and let it contain the element other than i in pair j^ℓ_s and a new dummy element that does not appear elsewhere. Further, couple the two instances by letting the common nodes realize identical random bits and by letting the new node draw fresh random bits. We claim that whenever event F happens in the original instance, it also happens in the new instance. If arc $(\hat{j}^\ell_s, j^\ell_s)$ is not realized, the rest of the arcs are realized identically in the two cases. Otherwise, having arc $(\hat{j}^\ell_s, j^\ell_s)$ may preclude arc $(j^\ell_{s-1}, j^\ell_s)_i$ from being realized, making event F more likely to happen in the new instance.

Type 2 Violation. Consider an instance almost identical to the one at hand except that we introduce a new node $\hat{j}^\ell_s \neq j^{\ell'}_{s'}$ for such a violation. For example, let $\hat{j}^\ell_s$ be a pair arriving after $j^{\ell'}_{s'}$ and before j^ℓ_s that involves the element other than i in these two pairs and a new dummy element that does not appear elsewhere. Further, couple the two instances by letting the common nodes realize identical random bits and by letting the new nodes draw fresh random bits. We claim that whenever event F happens in the original instance, it also happens in the new instance. Since F happens in the original instance, arc $(j^{\ell'}_{s'}, j^\ell_s)$ is *not* realized. If further arc $(\hat{j}^\ell_s, j^\ell_s)$ is not realized, the rest of the arcs are realized identically in the two cases. Otherwise, having arc $(\hat{j}^\ell_s, j^\ell_s)$ may preclude arc $(j^\ell_{s-1}, j^\ell_s)_i$ from being realized, making event F more likely to happen in the new instance.

Type 3 Violation. Consider an instance almost identical to the one at hand except that we introduce a new node $\hat{j}^\ell_s \neq \hat{j}^{\ell'}_{s'}$ for such a violation. For example, let $\hat{j}^\ell_s$ be a pair arriving right before j^ℓ_s that involves the element other than i in pair j^ℓ_s and a new dummy element that does not appear elsewhere. To avoid confusion in the following discussion, let $\hat{j}$ be the node in the type 3 violation in the original instance, and let $\hat{j}^\ell_s \neq \hat{j}^{\ell'}_{s'}$ be the nodes in the new instance. Further, couple the two instances by letting nodes other than $\hat{j}$, $\hat{j}^\ell_s$, and $\hat{j}^{\ell'}_{s'}$ realize identical random bits. To define the coupling for these three nodes, we need some notations. We say that node j^ℓ_s *needs help* if node j^ℓ_{s-1} is a sender who picks arc $(j^\ell_{s-1}, j^\ell_s)_i$ and if node j^ℓ_s is a receiver who breaks the tie against j^ℓ_{s-1}. For F to happen in this case, $\hat{j}^\ell_s$ must be a sender who picks arc $(\hat{j}^\ell_s, j^\ell_s)$. Define similarly for node $j^{\ell'}_{s'}$. If j^ℓ_s needs help but $j^{\ell'}_{s'}$ does not, let j^* and j^ℓ_s realize identical random bits and let $\hat{j}^{\ell'}_{s'}$ draw fresh random bits, and vice versa. Otherwise, that is, if none or both need help, let them have independent random bits. Then, when at most one needs help, event F happens in the original instance if and only if it happens in the new instance since the realization of the relevant arcs is identical. If both need help, on the other hand, F cannot happen in the original instance because at least one of $(j^\ell_{s-1}, j^\ell_s)_i$ or $(j^{\ell'}_{s'-1}, j^{\ell'}_{s'})_i$ would be realized. □

6 CONCLUSION

This article presents an online primal-dual algorithm for the edge-weighted bipartite matching problem that is 0.5086-competitive, resolving a long-standing open problem in the study of online algorithms. In particular, this work merges and refines the results of Fahrbach and Zadimoghaddam [12] and Huang and Tao [23, 30] to give a simpler algorithm under the online primal-dual framework. Our work initiates the study of *online correlated selection* (OCS), a key algorithmic ingredient for quantifying the level of negative correlation in online assignment problems. This new technique has already found applications in related online matching problems. Subsequent to our article, Huang et al. [31] generalized our OCS to obtain the first online algorithm that breaks the 1/2 barrier in the general AdWords problem, and Huang et al. [27] and Tang et al. [48] applied OCS to online stochastic matching.

Using independent random bits to make each selection is a 0-OCS (no negative correlation), and using an imaginary 1-OCS with perfect negative correlation results in infeasible assignments. Therefore, we aim to design an online matching algorithm using *partial negative correlation.* We first construct a $^1/_{16}$-OCS, and then we optimize this subroutine to obtain a 0.1099-OCS. Designing a γ-OCS with the largest possible γ is an interesting open problem, as it directly improves the competitive ratio of the edge-weighted online bipartite matching algorithm in this article. Subsequent to our work, Gao et al. [19] improved the lower and upper bounds of γ to 0.167 and 0.25, respectively.

Even if a 1-OCS did exist, the best competitive ratio that can be achieved using this approach (without any deviations) is at most $^5/_9$, as shown by Huang and Tao [30]. Hence, we need fundamentally new ideas to come closer to the optimal $1 - {^1/_e}$ ratio in the unweighted and vertex-weighted cases. One idea is to consider an OCS that allows for more than two candidates in each round, which we call *multiway OCS.* Subsequent to our work, Gao et al. [19] gave a multiway semi-OCS, and Blanc and Charikar [3] and Shin and An [47] made progress on multiway OCS. Blanc and Charikar [3] used their 6-way OCS to get the current best 0.536-competitive ratio for edge-weighted online bipartite matching. Designing optimal multiway OCS is another interesting open problem for future works.

APPENDICES

A UNWEIGHTED ONLINE MATCHING

This section considers the unweighted case and shows that the two-choice greedy algorithm is strictly better than the $^1/_2$-competitive algorithm when it uses OCS for the randomized rounds. We write g_k for $(1-\gamma)^{\max\{k-1,0\}}$ in the following discussion.

Theorem A.1. *The two-choice greedy algorithm with the randomized rounds that use a* 0.1099-*OCS is at least* 0.508-*competitive for unweighted online bipartite matching.*

Proof. In the unweighted case, it suffices to consider a single weight-level $w = 1$. Thus, for each offline vertex i, we write $k_i = k_i(1)$ for brevity. We will maintain $\overline{x}_i = 1 - 2^{-k_i} \cdot g_{k_i}$ for each offline vertex i, which, according to Lemma 5.3, lower bounds the probability that i is matched. Correspondingly, we maintain the following lower bound on the primal objective:

$$\overline{\mathrm{P}} = \sum_{i \in L} \overline{x}_i .$$

To prove the stated competitive ratio, it suffices to explain how to maintain a dual assignment such that (1) the dual objective equals the lower bound of the primal objective, that is, $\mathrm{D} = \overline{\mathrm{P}}$, and (2) it is approximately feasible up to a Γ factor, that is, $\alpha_i + \beta_j \geq \Gamma$ for every edge $(i, j) \in E$.

Dual Updates. The dual updates are based on a solution to a finite version of the following LP. All of the solution values are presented in Table 2 at the end of this section. The constraints that follow are simpler than in the more general edge-weighted case, but the competitive ratio we achieve is almost the same.

Lemma A.2. *The optimal value of the LP that follows is at least* 0.508*:*

$$\begin{aligned} \text{maximize} \quad & \Gamma \\ \text{subject to} \quad & a(k) + b(k) \leq 2^{-k} \cdot g_k - 2^{-(k+1)} \cdot g_{k+1} && \forall\, k \geq 0 \qquad (19) \\ & \sum_{\ell=0}^{k-1} a(\ell) + 2 \cdot b(k) \geq \Gamma && \forall\, k \geq 0 \qquad (20) \end{aligned}$$

$$\begin{aligned} &b(k) \geq b(k+1) && \forall\, k \geq 0 \qquad (21)\\ &a(k), b(k) \geq 0 && \forall\, k \geq 0 \end{aligned}$$

Consider an online vertex $j \in R$, and let $k_{\min} = \min_{i \in N(j)} k_i$ denote the minimum value of k_i among offline neighbors i of vertex j. First, suppose that it is a randomized round. Recall that i_1 and i_2 denote the two candidate offline vertices shortlisted in round j. Then, we have that $k_i = k_{\min}$ for both $i \in \{i_1, i_2\}$. In the primal, $\overline{x}_i$ increases by $2^{-k_{\min}} \cdot g_{k_{\min}} - 2^{-(k_{\min}+1)} \cdot g_{k_{\min}+1}$ for both $i \in \{i_1, i_2\}$. In the dual, increase α_i by $a(k_{\min})$ for both $i \in \{i_1, i_2\}$ and let $\beta_j = 2 \cdot b(k_{\min})$, where each $i \in \{i_1, i_2\}$ contributes $b(k_{\min})$.

Next, suppose that it is a deterministic round. Recall that i^* denotes the offline vertex to which vertex j is matched deterministically. Then, $\overline{x}_{i^*}$ increases by $2^{-k_{\min}} \cdot g_{k_{\min}}$ in the primal. In the dual, increase α_{i^*} by $\sum_{\ell \geq k_{\min}} a(\ell)$ and let $\beta_j = 2 \cdot b(k_{\min} + 1)$. No update is needed in an unmatched round, as $\overline{\mathrm{P}}$ remains the same.

Objective Comparisons. Next, we show that the increment in the dual objective D is at most that in the lower bound of the primal objective, that is, $\overline{\mathrm{P}}$. In a randomized round, it is followed by Equation (19). In a deterministic round, it is followed by the sequence of inequalities here:

$$\begin{aligned} \sum_{\ell \geq k} a(\ell) + 2 \cdot b(k+1) &\leq \sum_{\ell \geq k} a(\ell) + b(k) + b(k+1) && \text{(by Equation (21))}\\ &\leq \sum_{\ell \geq k} \left(a(\ell) + b(\ell)\right)\\ &\leq \sum_{\ell \geq k} \left(2^{-\ell} \cdot g_\ell - 2^{-(\ell+1)} \cdot g_{\ell+1}\right) && \text{(by Equation (19))}\\ &= 2^{-k} \cdot g_k. \end{aligned}$$

Approximate Dual Feasibility. We first summarize the following invariants, which follow from the definitions of the dual updates.

- For any offline vertex $i \in L$, $\alpha_i = \sum_{\ell=0}^{k_i - 1} a(\ell)$.
- For any online vertex j, $\beta_j = 2 \cdot b(k)$ if it is matched in a randomized round to neighbors with $k_i = k$ or in a deterministic round to a neighbor with $k_i = k - 1$.

For any edge $(i, j) \in E$, consider the value of k_i at the time when j arrives. If $k_i = \infty$, the value of α_i alone ensures approximate dual feasibility because

$$\begin{aligned} \alpha_i &= \sum_{\ell \geq 0} a(\ell)\\ &= \lim_{k \to \infty} \sum_{\ell=0}^{k-1} a(\ell)\\ &\geq \Gamma - 2 \lim_{k \to \infty} b(k) && \text{(by Equation (20))}\\ &= \Gamma\,. && \text{(by Equation (19), whose RHS tends to 0)} \end{aligned}$$

Otherwise, by the definition of the two-choice greedy algorithm, j is either matched in a randomized round to two vertices with $k_{i'} \leq k_i$ or in a deterministic round to a vertex with $k_{i'} < k_i$. In both cases, we have that $\beta_j \geq 2 \cdot b(k_i)$. Approximate dual feasibility now follows by $\alpha_i = \sum_{\ell=0}^{k_i - 1} a(\ell)$ and Equation (20). □

Table 2. An Approximately Optimal Solution to the FactorRevealing Linear Program with $k_{\max} = 8$ in the Unweighted Case

k	g_k	$a(k)$	$b(k)$
0	1.00000000	0.24550678	0.25449322
1	1.00000000	0.14574204	0.13173982
2	0.89007253	0.06613120	0.05886880
3	0.78014506	0.02907108	0.02580320
4	0.68230164	0.01273424	0.01126766
5	0.59654227	0.00559236	0.00490054
6	0.52153858	0.00248228	0.00210436
7	0.45596220	0.00114193	0.00086312
8	0.39863078	0.00058431	0.00029216

The competitive ratio obtained is $\Gamma \approx 0.508986$. Recall that $g_k = (1-\gamma)^{\max\{k-1,0\}}$.

Proof of Lemma A.2. Consider a restricted version of the LP that is finite. For some positive $k_{\max}$, let $a(k) = b(k) = 0$ for all $k > k_{\max}$. Then, the linear program becomes

$$\begin{aligned}
&\text{maximize} && \Gamma && \\
&\text{subject to} && a(k) + b(k) \le 2^{-k} \cdot g_k - 2^{-(k+1)} \cdot g_{k+1} && 0 \le k \le k_{\max} \qquad \text{(revised 20)}\\
& && \sum_{\ell=0}^{k-1} a(\ell) + 2 \cdot b(k) \ge \Gamma && 0 \le k \le k_{\max} \qquad \text{(revised 20)}\\
& && \sum_{\ell=0}^{k_{\max}} a(\ell) \ge \Gamma && \text{(revised 20, boundary case)}\\
& && b(k) \ge b(k+1) && 0 \le k < k_{\max} \qquad \text{(revised 21)}\\
& && a(k), b(k) \ge 0 && 0 \le k \le k_{\max}
\end{aligned}$$

See Table 2 for an approximately optimal solution for the restricted LP with $k_{\max} = 8$, which gives a competitive ratio of $\Gamma \approx 0.508986$. □

B HARD INSTANCES

This section presents two families of *unweighted* graphs that demonstrate some hardness results for the online matching algorithms considered in this article. We also give an example showing that the two-choice greedy with perfect negative correlation is infeasible in the online setting.

B.1 Upper Triangular Graphs

Consider a bipartite graph with n vertices on each side. Let each online vertex $1 \le j \le n$ be incident to the offline vertices $j \le i \le n$. Thus, the adjacency matrix (with online vertices as rows and offline vertices as columns) is an upper triangular matrix. This is a standard instance for showing hardness that dates back to Karp et al. [38].

Theorem B.1. *The two-choice greedy algorithm using independent random bits in different randomized rounds is only* $(1/2 + o(1))$*-competitive.*

Proof. For ease of presentation, suppose that the algorithm chooses candidates in *reverse* lexicographical order. Consider an upper triangular graph with $n = 3^k$ for some large positive integer k. First, observe that there is a perfect matching where the i-th online vertex is matched to the i-th offline vertex. Hence, the optimal value is n.

Next, consider the performance of the online algorithm. The first $n/3 = 3^{k-1}$ vertices are matched to the last $2/3$ fraction of the offline vertices in randomized rounds. That is, their correct neighbors in the perfect matching are left unmatched, while the other offline vertices are only half matched. Then, the first one-third of the remaining online vertices (i.e., $1/3 \cdot 2n/3 = 2 \cdot 3^{k-2}$ in total) are matched to the last $(2/3)^2$ fraction of the offline vertices in randomized rounds. That is, their correct neighbors in the perfect matching are left matched by only half, while the correct neighbors of subsequent online vertices are now matched by three-quarters. The argument goes on recursively.

Therefore, omitting a lower-order term due to the last $2^k = n^{\log_3 2}$ vertices on both sides, the expected size of the matching is

$$\begin{aligned}\left(1 \cdot \frac{1}{3} + \frac{1}{2} \cdot \frac{2}{9} + \cdots + \left(\frac{1}{2}\right)^k \cdot \frac{2^k}{3^{k+1}} + \cdots\right) n &= \left(\frac{1}{3} + \frac{1}{9} + \cdots + \frac{1}{3^{k+1}} + \cdots\right) n \\ &= \frac{n}{2} \,.\end{aligned}$$

Hence, the two-choice greedy algorithm is at best $(1/2 + o(1))$-competitive. □

Theorem B.2. *The imaginary two-choice greedy algorithm with perfect negative correlation across different randomized rounds is only 5/9-competitive.*

Proof. For ease of presentation, suppose that the algorithm chooses candidates in *reverse* lexicographical order. Consider an upper triangular graph with $n = 9$. There are nine vertices on each side, denoted as $i_1, i_2, \dots, i_9$ and $j_1, j_2, \dots, j_9$, and a perfect matching with i_k matched to j_k for $k = 1, 2, \dots, 9$. The first three online vertices, j_1, j_2, and j_3, are connected to all offline vertices. After their arrivals, i_1, i_2, and i_3 are unmatched while the remaining six offline vertices are matched by half. Then, the next two online vertices, j_4 and j_5, are connected to the last six offline vertices, that is, i_4 to i_9. After their arrival, i_4 and i_5 remain matched by half, while i_6 to i_9 are fully matched. Therefore, the algorithm finds a matching of size $1/2 \cdot 2 + 1 \cdot 4 = 5$ in expectation, but the optimal matching has size 9. The competitive ratio is $5/9$, which matches the lower bound that we want to show. □

While these hardness results crucially rely on the fact that the two-choice greedy algorithm breaks ties in a deterministic way, we note that the analysis can be extended to random tie-breaking algorithms by permuting the offline vertices.

B.2 Erdös–Rényi Upper Triangular Graphs

Consider the following random bipartite graph that has n vertices on each side. Each online vertex $1 \le j \le n$ is incident to the offline vertex $i = j$ with certainty. Each offline vertex $j < i \le n$ is adjacent to j independently with probability p, where $0 < p < 1$ is a parameter to be determined.

By considering the Erdös–Rényi variant of upper triangular graphs instead of the original ones, we ensure that with high probability any fixed online vertex is paired with different offline vertices in its randomized round. This is effectively the worst-case scenario in the analysis of the OCS algorithm in Section 5. Letting $n = 2^{13}$ and $p = 2^{-6}$, an empirical evaluation shows that our analysis for the combination of a two-choice greedy algorithm and with an OCS is nearly optimal.

Observation 1. *The competitive ratio of the two-choice greedy algorithm with the OCS in Algorithm 2 is at most* 0.5057*-competitive.*

Observation 2. *The competitive ratio of the two-choice greedy algorithm with the OCS in Algorithm 3 is at most* 0.51*-competitive.*

B.3 Infeasibility

Now, we show that two-choice greedy with perfect negative correlation is infeasible in the online setting. Here, perfect negative correlation means that if an offline vertex is a candidate in two randomized rounds (i.e., it is fed as input to the OCS), then it must be selected by the OCS at least once and, hence, is matched. Consider a graph with 4 offline vertices, denoted as 1 to 4. The first online vertex, denoted as 5, is connected to 1 and 2. The second online vertex, denoted as 6, is connected to 3 and 4. The third online vertex, denoted as 7, has two possibilities: it is connected with either 1 and 3 or 1 and 4. In the former case, the following pairs of edges have perfect negative correlations: $(1,5)$ and $(2,5)$, $(1,7)$ and $(3,7)$, $(1,5)$ and $(1,7)$, and $(3,6)$ and $(3,7)$. The first two pairs are due to having the same online vertex; the last two pairs are due to having the same offline vertex. Hence, we can deduce that $(2,5)$ and $(3,6)$ have perfect positive correlation. In the latter case, however, a similar argument gives that $(2,5)$ and $(3,6)$ have perfect negative correlation. An online algorithm cannot handle both cases simultaneously since the correlation between $(2,5)$ and $(3,6)$ are determined before the arrival of vertex 7 in the online setting.

C CONNECTIONS TO THE ORIGINAL ALGORITHM

This section explains the connections between the online primal-dual algorithm in this article and the original algorithm by Fahrbach and Zadimoghaddam [12]. We start by briefly describing the algorithm in Algorithm 4, with minor modifications to make it consistent with the notations in this article. Then, we simplify the algorithm by considering the special case of unweighted online matching and present the result as Algorithm 5. Finally, we explain how Algorithm 5 in the unweighted case is effectively a two-choice greedy algorithm that implicitly uses the warmup $1/16$-OCS from Section 5.

C.1 Original Algorithm

The algorithm uses two parameters $\varepsilon, \delta > 0$ that are later optimized in the analysis in [12]. These parameters are not necessary for explaining the connections between the two algorithms; thus, we omit their values. For each offline vertex $i \in L$, the algorithm maintains the following state variables that express the behavior of the last randomized round involving i. First, it maintains a Boolean variable $\mathsf{active}(i)$ that indicates whether the realization of the last randomized selection involving vertex i can be adaptively used in the next randomized round in which i is involved. The goal here is to introduce negative correlation in the same way that the $1/16$-OCS does. Each offline vertex also maintains two state variables about the last randomized selection involving i: the corresponding online vertex $\mathsf{index}(i)$ and the other offline vertex $\mathsf{partner}(i)$ in the last randomized round. Finally, the realization of the last randomized selection is stored as $\mathsf{priority}(i)$. Informally in the notation of this article, $\mathsf{priority}(i) = 0$ corresponds to the case in which the last randomized round involving i is a receiver; otherwise (i.e., the sender case) $\mathsf{priority}(i)$ is 1 if i is selected the last time and is 2 if i is not selected.

For each online vertex $j \in R$, the matching decision is made using two different quality measures for the offline vertices. For each offline vertex i, gain_{ij} denotes how much i's heaviest edge weight would increase should j be matched to i. The first measure is the expectation of gain_{ij}, which equals $\int_0^{w_{ij}} (1 - y_i(w))\,\mathrm{d}w$ in the CCDF viewpoint of this article. It also defines $\mathsf{adaptive\_gain}_{ij}$,

ALGORITHM 4: Original 0.501-Competitive Algorithm for Edge-Weighted Bipartite Matching in [12].

```
parameters: $\varepsilon, \delta$
state variables: $\forall i \in L$, active$(i) \leftarrow$ false, index$(i) \leftarrow 0$, partner$(i) \leftarrow 0$, priority$(i) \leftarrow 0$, $S(i) \leftarrow 0$
for all online vertex $j$ do
  $M_j \leftarrow \max_{i \in L} \mathbb{E}[\mathsf{gain}_{ij}] = \int_0^{w_{ij}} \left(1 - y_i(w)\right) \mathrm{d}w$
  for all offline vertices $j$ do
    if active$(i)$ = true and $w_{ij} \geq w_{i,\mathsf{index}(i)} - \delta M_j$ then
      $\mathsf{adaptive\_gain}_{ij} \leftarrow (\mathbb{E}[\mathsf{gain}_{i,\mathsf{index}(i)}]/3 - (w_{i,\mathsf{index}(i)} - w_{ij})^+/3 - S(i))^+/12$
    else
      $\mathsf{adaptive\_gain}_{ij} \leftarrow 0$
    end if
  end for
  $B \leftarrow \{i \in L : w_{ij} \geq w_{i,\mathsf{index}(i)} - \delta M_j$ and $\mathbb{E}[\mathsf{gain}_{ij}] + 2/3 \cdot \mathsf{adaptive\_gain}_{ij} \geq (1-\varepsilon)M_j\}$
  if $|B| \geq 2$ then                (case 1: there are enough candidates to exploit adaptivity)
    pick $i_1, i_2 \in B$ with the largest $\mathbb{E}[\mathsf{gain}_{ij}] + 2/3 \cdot \mathsf{adaptive\_gain}_{ij}$
    set active(partner$(i)$) $\leftarrow$ false for $i = i_1, i_2$
    set active$(i) \leftarrow$ true, $S(i) \leftarrow 0$, index$(i) \leftarrow j$ for $i = i_1, i_2$
    set partner$(i_1) \leftarrow i_2$ and partner$(i_2) \leftarrow i_1$
    pick $\ell \in \{1, 2\}$ with the larger $\mathsf{adaptive\_gain}_{i_\ell j}$ and let $-\ell$ denote $3 - \ell$
    draw $R \in [0, 1)$ uniformly at random
    if $R \in [0, 1/3)$ or $\mathsf{adaptive\_gain}_{i_\ell j} = 0$ then
      if $\mathsf{priority}_{i_\ell} = 2$ and $\mathsf{adaptive\_gain}_{i_\ell j} > 0$ then match $j$ to $i_\ell$
      if $\mathsf{priority}_{i_\ell} = 1$ and $\mathsf{adaptive\_gain}_{i_\ell j} > 0$ then match $j$ to $i_{-\ell}$
      if $\mathsf{priority}_{i_\ell} = 0$ or $\mathsf{adaptive\_gain}_{i_\ell j} = 0$ then match $j$ to $i_1$ or $i_2$ with equal probability
      set $\mathsf{priority}_{i_1} \leftarrow 0$ and $\mathsf{priority}_{i_2} \leftarrow 0$
    else
      if $R \in [1/3, 2/3)$ then assign $i$ to $i_1$ and set $\mathsf{priority}_{i_1} \leftarrow 1$, $\mathsf{priority}_{i_2} \leftarrow 2$
      if $R \in [2/3, 1)$ then assign $i$ to $i_2$ and set $\mathsf{priority}_{i_1} \leftarrow 2$, $\mathsf{priority}_{i_2} \leftarrow 1$
    end if
  else                                              (case 2: there is no adaptivity)
    $B' \leftarrow \{i \in L : (w_{ij} \geq w_{i,\mathsf{index}(i)} - \delta M_j)$ and $(\mathbb{E}[\mathsf{gain}_{ij}] \geq (1-\varepsilon)M_j)\}$
    $C \leftarrow \{i \in L : (w_{ij} < w_{i,\mathsf{index}(i)} - \delta M_j)$ and $(\mathbb{E}[\mathsf{gain}_{ij}] \geq (1-\varepsilon)M_j)\}$
    if $|B' \cup C| = 1$ then
      match $j$ to $i_1 \leftarrow \arg\max_{i \in L} \mathbb{E}[\mathsf{gain}_{ij}]$
      set $S(i_1) \leftarrow S(i_1) + M_j$
    else
      if $B' \neq \emptyset$ then
        $i_1 \leftarrow$ the only advertiser in $B'$
      else
        $i_1 \leftarrow \arg\max_{i \in C} \mathbb{E}[\mathsf{gain}_{ij}]$
      end if
      $i_2 \leftarrow \arg\max_{i \in C \setminus \{i_1\}} \mathbb{E}[\mathsf{gain}_{ij}]$
      match $j$ to $i_1$ or $i_2$ with equal probability
      set $S(i_1) \leftarrow S(i_1) + M_j/2$ and $S(i_2) \leftarrow S(i_2) + M_j/2$
    end if
  end if
end for
```

which captures the extra value of matching j to i due to the ability to make adaptive decisions based on the realization of the last randomized round involving i. Informally, this corresponds to the benefit of using the OCS for negative correlation in our primal-dual algorithm. The formula of $\mathsf{adaptive\_gain}_{ij}$ is derived from the analysis in [12] and its interpretation is not necessary for understanding the connections between the two algorithms. We refer the reader to [12] for a more detailed explanation of Algorithm 4 in the general edge-weighted online matching problem.

C.2 Simplified and Symmetrized Algorithm for Unweighted Online Matching

We now focus on a simplified algorithm in the special case of unweighted online matching in order to better explain the connections to our primal-dual algorithm in this article. In this setting, $w_{ij} \in \{0, 1\}$ for any $i \in L$ and any $j \in R$.

Simplifying Case 2. The case in which $|B| \leq 1$ in Algorithm 4 is significantly simpler in the unweighted case. Observe that for any offline vertex $i \in L$, either $w_{ij} = 0$, in which case we have $\mathbb{E}[\mathsf{gain}_{ij}] = 0$, or $w_{ij} = 1$, in which case we have $w_{ij} > w_{i,\mathsf{index}(i)} - \delta M_j$. In other words, C is always an empty set. On the other hand, $|B'|$ is nonempty because the offline neighbor with the maximum value of $\mathbb{E}[\mathsf{gain}_{ij}]$ is always in the set. Further observe that B' must be a singleton because B' is a subset of B, which has at most one element since we are in the second case of the algorithm. Putting this all together, the algorithm always matches i to the unique element in B'. This corresponds to a deterministic round in the primal-dual algorithm in this article.

Simplifying Gains and Adaptive Gains. Recall that x_i denotes the probability that an offline vertex i is matched. Thus, the expected gain $\mathbb{E}[\mathsf{gain}_{ij}]$ in the unweighted case equals $1 - x_i$. Observe that the expected gain is 0 if an offline vertex i is involved in case 2 (i.e., a deterministic round).

Next, we consider the adaptive gain values. In the unweighted case, the second term in the formula for computing adaptive gains is always 0 for any offline neighbor i of the online vertex j, because both $w_{i,\mathsf{index}(i)}$ and w_{ij} are 1. The third term, on the other hand, equals 0 if i has never been in case 2, and otherwise equals $\mathbb{E}[\mathsf{gain}_{i,\mathsf{index}(i)}]$ due to the earlier discussion on the simplification of case 2 in the unweighted case. In other words, the adaptive gain can be simplified as $\mathbb{E}[\mathsf{gain}_{i,\mathsf{index}(i)}]/36$ for any i that has not yet been matched deterministically and is 0 otherwise.

Simplifying the Candidate Set. Since both the gain and the adaptive gain values are 0 for any offline vertex that has been deterministically matched, they cannot appear in the candidate set B. Therefore, it suffices to consider j's offline neighbors that have not yet been matched deterministically. For such vertices, the first condition of the candidate set B holds trivially because $w_{ij} = 1$ and $w_{i,\mathsf{index}(i)} - \delta M_j = 1 - \delta M_j < 1$. In conclusion, it suffices to keep only the second condition.

Symmetrizing the Choice of ℓ. We observe that choosing ℓ to maximize the adaptive gain has no significance in the analysis by the authors of [12]. The analysis therein distributes the benefit of making adaptive decisions with regard to to i_ℓ equally between i_1 and i_2, and for $i_{-\ell}$ it merely needs its share to be at least half the benefit of making adaptive decisions with regard to $i_{-\ell}$. To this end, we symmetrize the choice of $\ell \in \{1, 2\}$ to be uniformly at random. Doing so makes the connection to the algorithm in this article more apparent.

Optimizing the Efficiency of Adaptivity. Finally, we remove the condition on the value of adaptive gain being 0 in the if statement in the first case of the algorithm. This is driven by the observation that the online vertex j is matched randomly to i_1 and i_2 with equal probability whenever it holds, which is identical to the else case of the if statement. Moreover, the latter case allows us to store the realization of the random selection to be exploited adaptively later, whereas the former does not.

ALGORITHM 5: Simplified and Symmetrized Version of the Original Algorithm in the Unweighted Case

```
parameters: $\varepsilon, \delta$
state variables: $\forall i \in L$, active$(i) \leftarrow$ false, index$(i) \leftarrow 0$, partner$(i) \leftarrow 0$, priority$(i) \leftarrow 0$, $S(i) \leftarrow 0$
for all online vertices $j$ do
    let $U_j$ be the set of neighbors of $j$ that have not been matched deterministically
    $M_j \leftarrow \max_{i \in U_j} \mathbb{E}[\mathrm{gain}_{ij}] = 1 - x_i$
    for all $i \in U_j$ do
        if active$(i)$ = true then
            $\mathrm{adaptive\_gain}_{ij} \leftarrow \mathbb{E}[\mathrm{gain}_{i,\mathrm{index}(i)}]/36$
        else
            $\mathrm{adaptive\_gain}_{ij} \leftarrow 0$
        end if
    end for
    $B \leftarrow \{i \in U_j : \mathbb{E}[\mathrm{gain}_{ij}] + 2/3 \cdot \mathrm{adaptive\_gain}_{ij} \geq (1 - \varepsilon)M_j\}$
    if $|B| \geq 2$ then                                                  (randomized round)
        pick $i_1, i_2 \in B$ with the largest $\mathbb{E}[\mathrm{gain}_{ij}] + 2/3 \cdot \mathrm{adaptive\_gain}_{ij}$
        set active(partner$(i)$) $\leftarrow$ false for $i = i_1, i_2$
        set active$(i) \leftarrow$ true, $S(i) \leftarrow 0$, index$(i) \leftarrow j$ for $i = i_1, i_2$
        set partner$(i_1) \leftarrow i_2$ and partner$(i_2) \leftarrow i_1$
        draw $\ell \in \{1, 2\}$ uniformly at random and let $-\ell$ denote $3 - \ell$
        draw $R \in [0, 1)$ uniformly at random
        if $R \in [0, 1/3)$ then
            if $\mathrm{priority}_{i_\ell} = 2$ and $\mathrm{adaptive\_gain}_{i_\ell j} > 0$ then match $j$ to $i_\ell$
            if $\mathrm{priority}_{i_\ell} = 1$ and $\mathrm{adaptive\_gain}_{i_\ell j} > 0$ then match $j$ to $i_{-\ell}$
            if $\mathrm{priority}_{i_\ell} = 0$ or $\mathrm{adaptive\_gain}_{i_\ell j} = 0$ then match $j$ to $i_1$ or $i_2$ with equal probability
            set $\mathrm{priority}_{i_1} \leftarrow 0$ and $\mathrm{priority}_{i_2} \leftarrow 0$
        else
            if $R \in [1/3, 2/3)$ then assign $i$ to $i_1$ and set $\mathrm{priority}_{i_1} \leftarrow 1$, $\mathrm{priority}_{i_2} \leftarrow 2$
            if $R \in [2/3, 1)$ then assign $i$ to $i_2$ and set $\mathrm{priority}_{i_1} \leftarrow 2$, $\mathrm{priority}_{i_2} \leftarrow 1$
        end if
    else                                                                  (deterministic round)
        match $j$ to the $i$ with the largest $\mathbb{E}[\mathrm{gain}_{ij}]$
    end if
end for
```

Hence, other than making the algorithm closer to the OCS introduced in this article, this technical change also improves the efficiency of adaptivity in the original algorithm.

This simplified and symmetrized version of the original algorithm in the unweighted case is summarized as Algorithm 5.

C.3 Connections Between the Unweighted Algorithms

We focus on the randomized rounds to explain the connections to the warmup $1/16$-OCS in Section 5. If $R \in [1/3, 2/3)$, it corresponds to a sender round in the OCS where i_1 is selected. If $R \in [2/3, 1)$, it corresponds to a sender round where i_2 is selected. If $R \in [0, 1/3)$, it corresponds to a receiver round.

The choice of ℓ corresponds to the choice of a random in-arc by a receiver in the OCS. The state variable active(i) ensures that each sender's selection is adaptively used by at most one receiver. In the OCS, the sender randomly picks an out-arc as the potential receiver. In contrast, Algorithm 5 deterministically picks the out-neighbor that arrives earlier. The authors of [12] effectively use an amortization in the analysis to distribute the benefit between the two out-arcs.

In conclusion, aside from the different choices of constants and the use of an amortization in the analysis instead of a symmetrized algorithm, the 0.501-competitive algorithm in [12] implicitly contains the ideas behind the warmup $1/16$-OCS presented as Algorithm 2 in this article.

REFERENCES

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

[2] Itai Ashlagi, Maximilien Burq, Chinmoy Dutta, Patrick Jaillet, Amin Saberi, and Chris Sholley. 2019. Edge weighted online windowed matching. In *Proceedings of the 20th ACM Conference on Economics and Computation*. ACM, 729–742.

[3] Guy Blanc and Moses Charikar. 2021. Multiway online correlated selection. In *Proceedings of the 62nd Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 1277–1284.

[4] Brian Brubach, Karthik A. Sankararaman, Aravind Srinivasan, and Pan Xu. 2020. Attenuate locally, win globally: Attenuation-based frameworks for online stochastic matching with timeouts. *Algorithmica* 82, 1 (2020), 64–87.

[5] Niv Buchbinder, Moran Feldman, Yuval Filmus, and Mohit Garg. 2020. Online submodular maximization: Beating 1/2 made simple. *Mathematical Programming* 183, 1 (2020), 149–169.

[6] Niv Buchbinder, Kamal Jain, and Joseph Seffi Naor. 2007. Online primal-dual algorithms for maximizing ad-auctions revenue. In *Proceedings of the 15th Annual European Conference on Algorithms*. Springer, 253–264.

[7] Nikhil R. Devanur and Thomas P. Hayes. 2009. The adwords problem: Online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM Conference on Electronic Commerce*. ACM, 71–78.

[8] Nikhil R. Devanur, Zhiyi Huang, Nitish Korula, Vahab S. Mirrokni, and Qiqi Yan. 2016. Whole-page optimization and submodular welfare maximization with online bidders. *ACM Transactions on Economics and Computation* 4, 3 (2016), 1–20.

[9] Nikhil R. Devanur, Kamal Jain, and Robert D. Kleinberg. 2013. Randomized primal-dual analysis of ranking for online bipartite matching. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 101–107.

[10] Nikhil R. Devanur, Kamal Jain, Balasubramanian Sivan, and Christopher A. Wilkens. 2011. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In *Proceedings of the 12th ACM Conference on Electronic Commerce*. ACM, 29–38.

[11] Hossein Esfandiari, Nitish Korula, and Vahab Mirrokni. 2015. Online allocation with traffic spikes: Mixing adversarial and stochastic models. In *Proceedings of the 16th ACM Conference on Economics and Computation*. 169–186.

[12] Matthew Fahrbach and Morteza Zadimoghaddam. 2019. Online weighted matching: Breaking the $\frac{1}{2}$ barrier. *arXiv preprint arXiv:1704.05384v2*.

[13] Jon Feldman, Monika Henzinger, Nitish Korula, Vahab S. Mirrokni, and Cliff Stein. 2010. Online stochastic packing applied to display ad allocation. In *Proceedings of the 18th Annual European Conference on Algorithms*. Springer, 182–194.

[14] Jon Feldman, Nitish Korula, Vahab S. Mirrokni, Shanmugavelayutham Muthukrishnan, and Martin Pál. 2009. Online ad assignment with free disposal. In *International Workshop on Internet and Network Economics*. Springer, 374–385.

[15] Jon Feldman, Aranyak Mehta, Vahab S. Mirrokni, and Shan Muthukrishnan. 2009. Online stochastic matching: Beating $1 - 1/e$. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 117–126.

[16] Marshall L. Fisher, George L. Nemhauser, and Laurence A. Wolsey. 1978. An analysis of approximations for maximizing submodular set functions-II. In *Polyhedral Combinatorics*. Springer, 73–87.

[17] Buddhima Gamlath, Sagar Kale, and Ola Svensso. 2019. Beating greedy for stochastic bipartite matching. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2841–2854.

[18] Buddhima Gamlath, Michael Kapralov, Andreas Maggiori, Ola Svensson, and David Wajc. 2019. Online matching with general arrivals. In *Proceedings of the 60th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 26–37.

[19] Ruiquan Gao, Zhongtian He, Zhiyi Huang, Zipei Nie, Bijun Yuan, and Yan Zhong. 2021. Improved online correlated selection. In *Proceedings of the 62nd Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 1265–1276.

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

[21] Guru Prashanth Guruganesh and Sahil Singla. 2017. Online matroid intersection: Beating half for random arrival. In *International Conference on Integer Programming and Combinatorial Optimization*. Springer, 241–253.

[22] Bernhard Haeupler, Vahab S. Mirrokni, and Morteza Zadimoghaddam. 2011. Online stochastic weighted matching: Improved approximation algorithms. In *International Workshop on Internet and Network Economics*. Springer, 170–181.

[23] Zhiyi Huang. 2019. Understanding Zadimoghaddam's edge-weighted online matching algorithm: Weighted case. *arXiv preprint arXiv:1910.03287* (2019).

[24] Zhiyi Huang, Ning Kang, Zhihao Gavin Tang, Xiaowei Wu, Yuhao Zhang, and Xue Zhu. 2018. How to match when all vertices arrive online. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 17–29.

[25] Zhiyi Huang, Binghui Peng, Zhihao Gavin Tang, Runzhou Tao, Xiaowei Wu, and Yuhao Zhang. 2019. Tight competitive ratios of classic matching algorithms in the fully online model. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2875–2886.

[26] Zhiyi Huang and Xinkai Shu. 2021. Online stochastic matching, Poisson arrivals, and the natural linear program. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 682–693.

[27] Zhiyi Huang, Xinkai Shu, and Shuyi Yan. 2022. The power of multiple choices in online stochastic matching. In *Proceedings of the 53rd Annual ACM Symposium on Theory of Computing*. ACM, 91–103.

[28] Zhiyi Huang, Zhihao Gavin Tang, Xiaowei Wu, and Yuhao Zhang. 2019. Online vertex-weighted bipartite matching: Beating $1 - 1/e$ with random arrivals. *ACM Transactions on Algorithms* 15, 3 (2019), 38.

[29] Zhiyi Huang, Zhihao Gavin Tang, Xiaowei Wu, and Yuhao Zhang. 2020. Fully online matching II: Beating ranking and water-filling. In *Proceedings of the 61st Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 1380–1391.

[30] Zhiyi Huang and Runzhou Tao. 2019. Understanding Zadimoghaddam's edge-weighted online matching algorithm: Unweighted case. *arXiv preprint arXiv:1910.02569*

[31] Zhiyi Huang, Qiankun Zhang, and Yuhao Zhang. 2020. AdWords in a panorama. In *Proceedings of the 61st Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 1416–1426.

[32] Patrick Jaillet and Xin Lu. 2013. Online stochastic matching: New algorithms with better bounds. *Mathematics of Operations Research* 39, 3 (2013), 624–646.

[33] Billy Jin and David P. Williamson. 2021. Improved analysis of RANKING for online vertex-weighted bipartite matching in the random order model. In *International Conference on Web and Internet Economics*. Springer, 207–225.

[34] Kumar Joag-Dev and Frank Proschan. 1983. Negative association of random variables with applications. *The Annals of Statistics* (1983), 286–295.

[35] Bala Kalyanasundaram and Kirk R. Pruhs. 2000. An optimal deterministic algorithm for online b-matching. *Theoretical Computer Science* 233, 1-2 (2000), 319–325.

[36] Michael Kapralov, Ian Post, and Jan Vondrák. 2013. Online submodular welfare maximization: Greedy is optimal. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 1216–1225.

[37] Chinmay Karande, Aranyak Mehta, and Pushkar Tripathi. 2011. Online bipartite matching with unknown distributions. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*. ACM, 587–596.

[38] Richard Karp, Umesh Vazirani, and Vijay Vazirani. 1990. An optimal algorithm for on-line bipartite matching. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*. ACM, 352–358.

[39] Nitish Korula, Vahab S. Mirrokni, and Morteza Zadimoghaddam. 2013. Bicriteria online matching: Maximizing weight and cardinality. In *International Conference on Web and Internet Economics*. Springer, 305–318.

[40] Nitish Korula, Vahab S. Mirrokni, and Morteza Zadimoghaddam. 2018. Online submodular welfare maximization: Greedy beats 1/2 in random order. *SIAM Journal on Computing* 47, 3 (2018), 1056–1086.

[41] Benny Lehmann, Daniel Lehmann, and Noam Nisan. 2006. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior* 55, 2 (2006), 270–296.

[42] Mohammad Mahdian and Qiqi Yan. 2011. Online bipartite matching with random arrivals: An approach based on strongly factor-revealing LPs. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*. ACM, 597–606.

[43] Vahideh H. Manshadi, Shayan Oveis Gharan, and Amin Saberi. 2012. Online stochastic matching: Online actions based on offline statistics. *Mathematics of Operations Research* 37, 4 (2012), 559–573.

[44] Aranyak Mehta. 2013. Online matching and ad allocation. *Foundations and Trends in Theoretical Computer Science* 8, 4 (2013), 265–368.

[45] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. 2005. AdWords and generalized on-line matching. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 264–273.

[46] Vahab S. Mirrokni, Shayan Oveis Gharan, and Morteza Zadimoghaddam. 2012. Simultaneous approximations for adversarial and stochastic online budgeted allocation. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 1690–1701.

[47] Yongho Shin and Hyung-Chan An. 2021. Making three out of two: three-way online correlated selection. In *Proceedings of the 32nd International Symposium on Algorithms and Computation*. Schloss Dagstuhl, 49:1–49:17.

[48] Zhihao Gavin Tang, Jinzhao Wu, and Hongxun Wu. 2022. (Fractional) online stochastic matching via fine-grained offline statistics. In *Proceedings of the 53rd Annual ACM Symposium on Theory of Computing*. ACM, 77–90.

Received 11 February 2021; revised 10 March 2022; accepted 19 July 2022