2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)

Edge-Weighted Online Bipartite Matching

Matthew Fahrbach
Google Research
fahrbach@google.com

Zhiyi Huang
The University of Hong Kong
zhiyi@cs.hku.hk

Runzhou Tao
Columbia University
runzhou.tao@columbia.edu

Morteza Zadimoghaddam
Google Research
zadim@google.com

***Abstract*—Online bipartite matching is one of the most fundamental problems in the online algorithms literature. Karp, Vazirani, and Vazirani (STOC 1990) introduced an elegant algorithm for the 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 paper, 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) that restricts beating a $1/2$ competitive ratio for the more general problem of monotone submodular welfare maximization, our result can be seen as strong evidence that edge-weighted bipartite matching is strictly easier than submodular welfare maximization in the 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.

***Keywords*-bipartite matching; negative correlation; online algorithm; primal-dual algorithm**

I. Introduction

Matchings are fundamental graph-theoretic objects that play an indispensable role in combinatorial optimization. For decades, there have been tremendous and ongoing efforts to design more efficient algorithms for finding maximum matchings in terms of their cardinality, and more generally, their 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 set (e.g., 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. In particular, 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 are known in advance. When an impression arrives, its edges to the advertisers are revealed and an irrevocable decision has to be made about to which advertiser the impression should be assigned. Karp, Vazirani, and Vazirani [28] gave 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. Later, Aggarwal et al. [1] generalized their algorithm to the vertex-weighted online bipartite matching problem and showed that the $1-1/e$ competitive ratio is still attainable.

The edge-weighted case, however, is much more nebulous. 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, but it has to decide whether or not 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 somehow an option.

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 any given advertiser. However, instead of achieving the weights of all the edges assigned to it, we only acknowledge the maximum weight (i.e., the objective equals the sum of the heaviest edge weight assigned

This document is an extended abstract which has neither full proofs nor our full set of results. Readers are strongly encouraged to read the full version of this paper [7] instead, which can be found at https://arxiv.org/abs/2005.01929.

2575-8454/20/$31.00 ©2020 IEEE
DOI 10.1109/FOCS46700.2020.00046

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 [10, 30], the free-disposal assumption is well received and widely applied because of its natural economic interpretation. Finally, edge-weighted online bipartite matching with free disposal is a special case of the monotone submodular welfare maximization problem, where we can apply known $1/2$-competitive greedy algorithms [12, 31].

A. Our Contributions

Despite thirty years of research in online matching since the seminal work of Karp et al. [28], finding an algorithm for edge-weighted online bipartite matching that achieves a competitive ratio greater than $1/2$ has remained a tantalizing open problem. This paper gives a new online algorithm and answers the question affirmatively, breaking the long-standing $1/2$ barrier (under free disposal).

Theorem 1. *There is a 0.5086-competitive algorithm for edge-weighted online bipartite matching.*

Given the hardness result of Kapralov, Post, and Vondrák [26] that restricts beating a competitive ratio of $1/2$ for monotone submodular welfare maximization, our algorithm shows that edge-weighted bipartite matching is strictly easier than submodular welfare maximization in the online setting.

From now on, 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 ultimately guarantees that a vertex appearing in k pairs is selected at least once with probability strictly greater than $1 - 2^{-k}$. We give a short introduction to OCS in Section III and defer all other details to the full version of this paper [7].

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 of them, or choose one offline neighbor deterministically. More concretely, among the neighbors that have not been matched deterministically, find the least-matched ones (i.e., those that have appeared in the least number of pairs). Pick two if there are at least two of them; otherwise, choose one deterministically.

Although the competitive ratio of the algorithm above is far worse than the optimal $1 - 1/e$ ratio by Karp et al. [28], 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 one neighbor's heaviest edge weight is either 1 or 4 each with probability $1/2$, 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 remedy this, we use the online primal-dual framework for matching problems by Devanur, Jain, and Kleinberg [5], along with an alternative formulation of the edge-weighted online bipartite matching problem by Devanur et al. [4]. In short, we account for the contribution of each offline vertex by weight-levels, and at each weight-level we consider the probability that the heaviest edge matched to the vertex has weight at least this level. This is the complementary cumulative distribution function (CCDF) of the heaviest edge weight, and 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 II. Then we formally present our edge-weighted matching algorithm in Section IV, followed by its analysis. In [7, Appendix B] we include hard instances that show the competitive ratio of our algorithm is nearly tight.

B. Related Works

The literature of online weighted bipartite matching algorithms is extensive, but most of these works are devoted to achieving 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. Below we list the most relevant works and refer interested readers to the excellent survey of Mehta [34]. We note that there have recently been several significant advances in more general settings, including different arrival models and general (non-bipartite) graphs [18, 13, 14, 19, 21].

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 two decades ago to Kalyanasundaram and Pruhs [25]. Feldman et al. [10] gave a $(1 - 1/e)$-competitive algorithm for Display Ads, which is equivalent to edge-weighted online bipartite matching assuming large capacities. Under similar assumptions, the same competitive ratio was obtained for AdWords [35, 2], in which offline vertices have some 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 primary goals in the online matching literature is to provide algorithms with 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 the online vertices are drawn from some known or unknown distribution [11, 27, 6, 16, 33, 24], or that they arrive in a random order [15, 3, 9, 32, 36, 20]. 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. Korula, Mirrokni, and Zadimoghaddam [29] 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 random order assumption is particularly justified because Kapralov et al. [26] proved that beating $1/2$ for submodular welfare maximization in the oblivious adversary model implies **NP** = **RP**.

II. 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, i.e., $E = L \times R$, 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, who must then irrevocably match j to an offline neighbor. Each offline vertex can be matched any number of times, but 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*.

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.

A. Complementary Cumulative Distribution Function Viewpoint

Next we describe an alternative formulation of the edge-weighted online matching problem due to Devanur et al. [4] that captures the contribution of each offline vertex $i \in L$ to the objective in terms of the complementary cumulative distribution function (CCDF) of the heaviest edge weight matched to i. We refer to this approach 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, i.e., the probability that i is matched to at least one online vertex j such that $w_{ij} \geq w$. Then, $y_i(w)$ is a non-increasing function of w that takes values between 0 and 1. Observe that $y_i(w)$ is a step function with polynomially many pieces, because 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 $y_i(w)$, i.e.:

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

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

We illustrative this idea with an example in Figure 1. Suppose 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, while j_2, j_3, and j_4 each have some probability of being matched to i. (The latter events may be correlated.) Next, suppose 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.

B. Online Primal-Dual Framework

We analyze our algorithms 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 edge (i, j) is the heaviest edge matched to vertex i. The primal and dual LPs are:

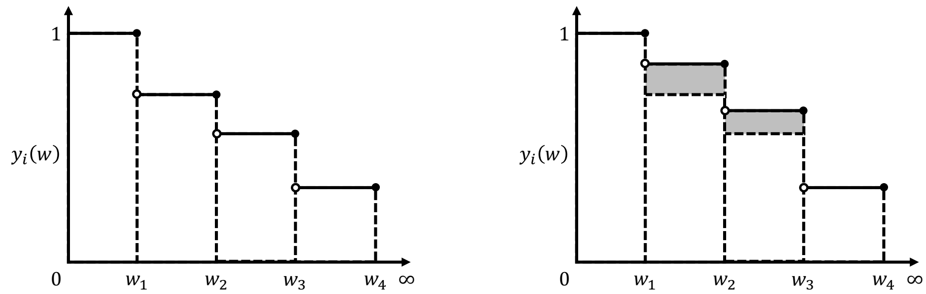

Figure 1: Complementary cumulative distribution function (CCDF) viewpoint. The first function is the CCDF of vertex i, and the second function demonstrates how the CCDF of vertex i is updated.

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in L} \sum_{j \in R} w_{ij} x_{ij} \\ \text{subject to} \quad & \sum_{j \in R} x_{ij} \le 1 && \forall\, i \in L \\ & \sum_{i \in L} x_{ij} \le 1 && \forall\, j \in R \\ & x_{ij} \ge 0 && \forall i \in L, \forall\, j \in R \end{aligned}$$

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in L} \alpha_i + \sum_{j \in R} \beta_j \\ \text{subject to} \quad & \alpha_i + \beta_j \ge w_{ij} && \forall\, i \in L, \forall j \in R \\ & \alpha_i \ge 0 && \forall\, i \in L \\ & \beta_j \ge 0 && \forall\, j \in R \end{aligned}$$

Let P denote the primal objective. If x_{ij} is the probability that edge (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 a matching as well as a dual assignment (although not necessarily feasible) at all times subject to the conditions summarized below.

Lemma 2. *Suppose 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 $\alpha_i + \beta_j \ge \Gamma \cdot w_{ij}$.*
2) **Reverse weak duality:** *The objectives of the primal and dual assignments satisfy* P $\ge$ D.

Then, the algorithm is Γ-competitive.

Proof: The values $\Gamma^{-1}\alpha_i$ and $\Gamma^{-1}\beta_j$ form a feasible dual assignment whose objective equals Γ^{-1}D. By weak duality, 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 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 $w \in \{w_{ij} \in E :$ online vertex j 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 IV-A. 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.

III. 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, i.e., $w_{ij} \in \{0, 1\}$ 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, and the algorithm deterministically chooses one of them. The second online vertex, however, is only adjacent to the previously matched vertex.

Two-Choice Greedy with Independent Random Bits: We can avoid the problem above 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, e.g., 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 the full version of the paper [7, Appendix B].

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. This is infeasible in general. Nevertheless, if we assume feasibility then this algorithm is $5/9$-competitive, as shown in an earlier version of this paper [22]. In fact, it is effectively the 2-matching algorithm of Kalyanasundaram and Pruhs [25], by having two copies of each online vertex and allowing offline vertices to be matched twice.

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

We answer this question affirmatively by introducing an algorithmic ingredient called *online correlated selection* (OCS), which allows us to quantify the negative correlation among randomized rounds. We analyze the two-choice greedy algorithm powered by OCS in the unweighted case in [7, Appendix A]. In this extended abstract, we present tools to generalize that approach to edge-weighted online bipartite matching, and achieve the first algorithm with a competitive ratio that is provably greater than $1/2$.

Definition 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 per pair such that if an element appears 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 corresponds to 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 edge-weighted online bipartite matching.

In the following definition, a subsequence (not necessarily contiguous) of pairs containing element i is *consecutive* if it includes all the pairs that contain element i between the first and last pair in the subsequence. Further, 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 subsequence $(\{a,i\},\{b,i\},\{i,z\})$ is not consecutive because it does not include the pair $\{e,i\}$.

Definition 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 $k_1, k_2, \ldots, k_m$ 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. *There exists a* $\frac{13\sqrt{13}-35}{108} > 0.1099$*-OCS.*

We defer the design and analysis of this OCS to the full version of the paper [7, Section 5]. Instead, we will describe a weaker $1/16$-OCS, which is already sufficient for breaking the $1/2$ barrier in edge-weighted online bipartite matching. We also include a formal construction and proof for the $1/16$-OCS in the full version.

Proof Sketch of a $1/16$-OCS: Consider two sequences of independent random bits. The first set is used to construct a random matching among the pairs, where any two consecutive pairs (with respect to some element) are matched with probability $1/16$. Concretely, each pair is consecutive to at most four other 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 any 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 from the first pair, and then make the opposite selection for the later one (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 $k_1, k_2, \ldots, k_m$ consecutive pairs containing i. We bound the probability that i is never selected. If any two of the 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 of having no such matched pairs 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 $1/16$. Further, these events are negatively dependent and therefore, the probability that none of them happens is upper bounded by the independent case. ■

IV. 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 III) it is 0.505-competitive. For $\gamma \approx 0.1099$ (as in Theorem 3) it is 0.5086-competitive, which proves our main result about edge-weighted online matching.

A. *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 of the algorithm.

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 and has edge weight at least w. The values of $k_i(w)$ may change over time, so 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 presented at the end of this section in Table I.

- $a(k)$: 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:

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

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

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

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 an randomized round. The first term in Eqn. 5 follows from the interpretation of $b(k)$ above (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 $w > w_{ij}$

Algorithm 1 Online primal-dual edge-weighted bipartite matching algorithm.

State variables:

- $k_i(w) \ge 0$: The 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 Eqn. (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 $0 > \Delta_{i_1}^R \beta_j + \Delta_{i_2}^R \beta_j$ and $\Delta_{i^*}^D \beta_j$, leave j unmatched. **(unmatched)**
5) If $\Delta_{i_1}^R \beta_j + \Delta_{i_2}^R \beta_j \ge \Delta_{i^*}^D \beta_j$ and 0, let the OCS pick one of i_1 and i_2. **(randomized)**
6) If $\Delta_{i^*}^D \beta_j > \Delta_{i_1}^R \beta_j$ and 0, match j to i^*. **(deterministic)**
7) Update the $k_i(w)$'s accordingly.

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 $\Delta_i^D \beta_j \stackrel{\text{def}}{=} \kappa \cdot \Delta_i^R \beta_j$ to be:

$$\kappa \int_0^{w_{ij}} b\left(k_i(w)\right)\mathrm{d}w - \frac{\kappa}{2} \int_{w_{ij}}^\infty \sum_{0 \le \ell < k_i(w)} a(\ell)\,\mathrm{d}w \,. \quad (6)$$

For concreteness, readers can assume $\kappa = 1.5$. 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 the 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 Subsection IV-C, and we 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_j's. It remains to define the updates to $\alpha_i(w)$'s. Before that, we first 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 as fixed.

For any offline vertex i and any 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, $k_1, k_2, \dots, k_m$ consecutive rounds. By Definition 2, vertex i is selected by the γ-OCS in at least one of these rounds with probability at least:

$$\overline{y}_i(w) \stackrel{\text{def}}{=} 1 - \prod_{\ell=1}^m 2^{-k_\ell} \left(1 - \gamma\right)^{k_\ell - 1} \,. \quad (7)$$

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. *The primal objective is lower bounded by the surrogate, i.e.,* $\overline{\mathrm{P}} \le \mathrm{P}$.

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, let $\overline{y}_i(w) = 1$.
- 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 $(1-\gamma)/2$ factor if $w' \geq w$, i.e., if it is the second or later pair of a collection of consecutive pairs containing i with edge weight at least w; otherwise, decrease $1 - \overline{y}_i(w)$ by $1/2$ to account for the -1 in the exponent of $1-\gamma$ in Eqn 7.

Lemma 5. *For any offline vertex i and any weight-level $w > 0$, we have:*

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

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

This is equivalent to a lower bound of the increment in $y_i(w)$ in a deterministic round.

Lemma 6. *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:*

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

Lemma 7. *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:*

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

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:

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

Proof: By definition, $1 - \overline{y}_i(w)$ decreases by a factor of either $(1-\gamma)/2$ 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 $(1+\gamma)/2$ 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 5 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 $\beta_j = \Delta^R_{i_1}\beta_j + \Delta^R_{i_2}\beta_j$. Lastly, if j is matched in a deterministic round, set $\beta_j = \Delta^D_{i^*}\beta_j$.

Dual Updates to Offline Vertices—Proof of Eqn. (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, so 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 Eqn. (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 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 three subsets A, B, C, each of which requires a different update rule for $\alpha_i(w)$. Concretely, the algorithm increases $\alpha_i(w)$ by:

$$\begin{cases} a\left(k_i(w)\right) & \text{if } w \in A, \\ a\left(k_i(w)\right) - 2^{-k_i(w)-1}(1-\gamma)^{k_i(w)-1}\gamma & \text{if } w \in B, \\ 2^{-k_i(w)-1}(1-\gamma)^{k_i(w)-1}\gamma & \text{if } w \in C, \end{cases} \tag{9}$$

where

$$\begin{aligned} A &= \{w > 0 : 0 < w \leq w_{ij}, w' \text{ or } k_i(w) = 0\}, \\ B &= \{w > 0 : w' < w \leq w_{ij} \text{ and } k_i(w) \geq 1\}, \\ C &= \{w > 0 : w > w_{ij} \text{ and } k_i(w) \geq 1\}. \end{aligned}$$

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

For a weight-level w that falls into the second case B (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 7, 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 Eqn. (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 Eqn. (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 C in the previous randomized round in which i is involved. 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.

B. 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 j is matched to i in a deterministic round. Using the lower bound for the increase of $\overline{\mathrm{P}}$ in Lemma 6, the increase of the $\alpha_i(w)$'s in Eqn. (8), and a lower bound for β_j by dropping the second term in Eqn. (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 this inequality locally at every weight-level, so it suffices to satisfy the following for all $k \geq 0$:

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

Randomized Rounds: Now suppose 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., $\Delta_{i_1}^R \beta_j$). This also holds for i_2 by symmetry, and 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 Eqn. (9)

The *first case* is when $w \in A$ (i.e., $w \leq w_1, w_1'$ or $k_i(w) = 0$). By Lemma 7, the increase in $\overline{\mathrm{P}}$ due to vertex i at weight-level w if $k_i(w) = 0$ is at least $1/2$. If $k_i(w) \geq 1$ and $w \leq w_1, w_1'$ then the increase is at least:

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

By the first case of Eqn. (9), the increase in $\alpha_i(w)$ is $a(k_i(w))$. Finally, the contribution to the first term of $\beta_j = \Delta_i^R \beta_j + \Delta_{i_2}^R \beta_j$, at weight-level w, in Eqn. (5) is $b(k_i(w))$. Hence, it suffices to ensure $a(0) + a(0) \leq 1/2$, and for all $k \geq 1$:

$$a(k) + b(k) \leq 2^{-k-1}\left(1-\gamma\right)^{k-1}\left(1+\gamma\right)\,. \tag{11}$$

The *second case* is when $w \in B$ (i.e., $w_1' < w \leq w_1$ and $k_i(w) \geq 1$). By Lemma 7, 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 Eqn. (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 must have:

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

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

The *third case* is when $w \in C$ (i.e., $w > w_1$ 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 Eqn. (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 $\frac{1}{2}\sum_{0\leq\ell<k_i(w)} a(\ell)$. Hence, we need:

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

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

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

C. Online Primal-Dual Analysis: Approximate Dual Feasibility

This subsection derives a set of conditions that are sufficient for approximate dual feasibility, i.e., Eqn. (3). Start by fixing any $i \in L$ and any $j \in R$, and also 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 $0 < w \leq w_{ij}$ and j is unmatched. This means $\beta_j = 0$ in this round and 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 Eqn. (4), it suffices to have:

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

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 when j is in a randomized round, and then we will show that the other cases only require weaker conditions.

Case 1—Round of j is a 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 Eqn. (5) and lower bounding $\alpha_i(w)$ by Eqn. (4), approximate dual feasibility in Eqn. (3) reduces to:

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

We will again ensure this inequality at every weight-level. Thus, it suffices to have the following for all $k \geq 0$:

$$\sum_{0 \leq \ell < k} a(\ell) + 2 \cdot b(k) \geq \Gamma \,. \tag{14}$$

Case 2—Round of j is randomized, i is chosen: By symmetry, suppose WLOG that $i \leftarrow i_2$ and i_1 is the other candidate. By definition, $\beta_j = \Delta_i^R \beta_j + \Delta_{i_1}^R \beta_j$. 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 $\Delta_i^R \beta_j + \Delta_{i_1}^R \beta_j \geq \Delta_i^D \beta_j$. Further, we have $\Delta_i^D \beta_j = \kappa \cdot \Delta_i^R \beta_j$ by Eqn. (6). Combining these, we have $\beta_j \geq \kappa \cdot \Delta_i^R \beta_j$. Finally, by the definition of $\Delta_i^R \beta_j$ in Eqn. (5), β_j is at least:

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

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 $0 < w \leq w_{ij}$ 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:

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

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

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

Thus, it suffices to ensure the inequality locally at every weight-level, i.e., for all $k \geq 0$ we need:

$$\sum_{0 \leq \ell \leq k} a(\ell) + \kappa \cdot b(k) \geq \Gamma \,. \tag{15}$$

There are two differences between Eqn. (14) and Eqn. (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 $w \leq w_{ij}$. Second, the κ coefficient for the second term is smaller.

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

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

Case 5—Round of j is unmatched: By definition, $\beta_j = 0$. Moreover, $\Delta_i^D \beta_j < 0$ because the algorithm chooses to leave j unmatched, which further implies $\Delta_i^R \beta_j < 0$ by Eqn. (6). Therefore, we have $\beta_j \geq 2 \cdot \Delta_i^R \beta_j$, identical to the lower bound in the first case. Thus, approximate dual feasibility is guaranteed by Eqn. (14).

D. 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 $b(k)$ via the following LP:

$$\begin{aligned} &\text{maximize} && \Gamma \\ &\text{subject to} && \text{Eqn. (10), (11), (12), (13), (14), (15)} \end{aligned}$$

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

We give an approximately optimal solution to the finite LP in Table Ia with $\gamma = 1/16$, $\kappa = 3/2$, and $k_{\max} = 8$, which shows $\Gamma > 0.505$. We also tried different values of $\kappa = 1 + \ell/16$, for $0 \leq \ell \leq 16$. 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, $\Gamma > 0.505$. Hence, the analysis is robust to the choice of κ, so long as it is neither too close to 1 nor to 2. In Table Ib we give an approximately optimal solution under the same setting, except we use a larger $\gamma = \frac{13\sqrt{13}-35}{108} > 0.1099$ as in Theorem 3, which leads to an improved competitive ratio $\Gamma > 0.5086$.[1]

V. Conclusion

This paper 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 [8] and Huang and Tao [22, 17] to give a simpler algorithm under the online primal-dual framework. Our work initiates the study of *online correlated selection*, a key algorithmic ingredient that quantifies the level of negative correlation in online assignment problems, and we believe this technique will find further applications in other online problems. Indeed,

[1] Our source code is available at https://github.com/fahrbach/focs-2020-edge-weighted-online-bipartite-matching.

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

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

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

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

Table I: Approximately optimal solutions to the factor-revealing LP with $\kappa = 3/2$ and $k_{\max} = 8$.

Huang, Zhang, and Zhang [23] recently generalized the OCS to obtain the first online algorithm that breaks the $1/2$ barrier in the general case of AdWords.

Acknowledgement

Matthew Fahrbach did part of this work while supported by an NSF Graduate Research Fellowship under grant DGE-1650044 at the Georgia Institute of Technology. Zhiyi Huang was supported in part by an RGC grant 17203717E.

References

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

[2] Niv Buchbinder, Kamal Jain, and Joseph Seffi Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *European Symposium on Algorithms*, pages 253–264. Springer, 2007.

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

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

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

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

[7] Matthew Fahrbach, Zhiyi Huang, Runzhou Tao, and Morteza Zadimoghaddam. Edge-weighted online bipartite matching. *arXiv preprint arXiv:2005.01929*, 2020.

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

[9] Jon Feldman, Monika Henzinger, Nitish Korula, Vahab S. Mirrokni, and Cliff Stein. Online stochastic packing applied to display ad allocation. In *European Symposium on Algorithms*, pages 182–194. Springer, 2010.

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

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

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

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

[14] Buddhima Gamlath, Michael Kapralov, Andreas Maggiori, Ola Svensson, and David Wajc. Online matching with general arrivals. In *Proceedings of*

the 60th Annual IEEE Symposium on Foundations of Computer Science, pages 26–37. IEEE, 2019.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[36] Aranyak Mehta, Bo Waggoner, and Morteza Zadimoghaddam. Online stochastic matching with unequal probabilities. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1388–1404. SIAM, 2015.