Beating Ratio 0.5 for Weighted Oblivious Matching Problems

Melika Abolhassani*[1], T-H. Hubert Chan†[2], Fei Chen‡[3], Hossein Esfandiari*[1], MohammadTaghi Hajiaghayi*[1], Hamid Mahini*[1], and Xiaowei Wu[2]

1 Department of Computer Sicence, Univeristy of Maryland, USA
melika,hossein,hajiagha,hmahini@cs.umd.edu

2 Department of Computer Sicence, The University of Hong Kong, Hong Kong
hubert,xwwu@cs.hku.hk

3 KTH Royal Institute of Technology, Sweden
feichen@kth.se

Abstract

We prove the first non-trivial performance ratios strictly above 0.5 for weighted versions of the oblivious matching problem. Even for the unweighted version, since Aronson, Dyer, Frieze, and Suen first proved a non-trivial ratio above 0.5 in the mid-1990s, during the next twenty years several attempts have been made to improve this ratio, until Chan, Chen, Wu and Zhao successfully achieved a significant ratio of 0.523 very recently (SODA 2014). To the best of our knowledge, our work is the first in the literature that considers the node-weighted and edge-weighted versions of the problem in arbitrary graphs (as opposed to bipartite graphs).

(1) For arbitrary node weights, we prove that a weighted version of the Ranking algorithm has ratio strictly above 0.5. We have discovered a new structural property of the ranking algorithm: if a node has two unmatched neighbors at the end of algorithm, then it will still be matched even when its rank is demoted to the bottom. This property allows us to form LP constraints for both the node-weighted and the unweighted oblivious matching problems. As a result, we prove that the ratio for the node-weighted case is at least 0.501512. Interestingly via the structural property, we can also improve slightly the ratio for the unweighted case to 0.526823 (from the previous best 0.523166 in SODA 2014).

(2) For a bounded number of distinct edge weights, we show that ratio strictly above 0.5 can be achieved by partitioning edges carefully according to the weights, and running the (unweighted) Ranking algorithm on each part. Our analysis is based on a new primal-dual framework known as *matching coverage*, in which dual feasibility is bypassed. Instead, only dual constraints corresponding to edges in an optimal matching are satisfied. Using this framework we also design and analyze an algorithm for the edge-weighted online bipartite matching problem with free disposal. We prove that for the case of bounded online degrees, the ratio is strictly above 0.5.

1998 ACM Subject Classification G.1.2 Approximation, G.1.6 Optimization, G.2.1 Combinatorics, G.2.2 Graph Theory

Keywords and phrases weighted matching, oblivious algorithms, Ranking, linear programming

Digital Object Identifier 10.4230/LIPIcs.ESA.2016.48

* Partially supported by NSF CAREER award CCF-1053605, NSF BIGDATA grant IIS-1546108, NSF AF:Medium grant CCF-1161365, DARPA GRAPHS/AFOSR grant FA9550-12-1-0423, and another DARPA SIMPLEX grant.

† Partially supported by a grant from Hong Kong RGC under the contract 17200214.

‡ Partially supported by Swedish Research Council grant 2015-04659 "Algorithms and Complexity for Dynamic Graph Problems".

1 Introduction

While the classical maximum matching problem [14] is well understood, the oblivious version is motivated by exchange settings [15] and online advertising [9, 1], in which information about the underlying graphs might be unknown. For instance, in the kidney exchange problem [15], donor-recipient pairs are probed and greedily matched when two pairs are compatible. Another example is pay-per-click online advertising, in which the revenue for a click on a particular ad showing on a particular page is known, but it is unknown whether the user will actually click on that ad. In this paper, we analyze two weighted versions of the oblivious matching problem (ObMP). To be more specific, we first state the edge-weighted (Ew) ObMP (and the node-weighted (Nw) version as a special case) formally as follows.

EwObMP . An *adversary* commits to a simple undirected graph $G = (V, E)$, where **every** unordered pair of nodes $e = \{u, v\}$ (even if $e \notin E$) has non-negative weight w_e. The *unweighted* case is the special case in which all pairs have the same weight. The nodes V (where $n = |V|$) and the weights of all pairs are revealed to the (randomized) *algorithm*, while the edges E are kept secret. The algorithm returns a list L that gives a permutation of the set $\binom{V}{2}$ of unordered pairs of nodes. Each pair of nodes in G is probed according to the order specified by L to form a matching greedily. In the round when a pair $e = \{u, v\}$ is probed, if both nodes are currently *unmatched* and the edge e is in E, then the two nodes will be *matched* to each other; otherwise, we skip to the next pair in L until all pairs in L are probed. The goal is to maximize the ***performance ratio*** of the (expected) sum of weights of edges in the matching produced by the algorithm to that of a maximum weight matching in G. The node-weighted version is related to the edge-weighted version as follows.

NwObMP . The node-weighted version is a special case of EwObMP in which each node $u \in V$ has a non-negative weight w_u and the weight of each pair $e = \{u, v\}$ is $w_e = w_u + w_v$.

Greedy Algorithms. Greedy algorithms can achieve ratio 0.5 for both the edge-weighted and node-weighted versions. For the edge-weighted version, the probing order is given by sorting pairs in non-increasing order of weight. For the node-weighted version, the nodes are sorted in non-increasing order of weight to induce a lexicographical order on the pairs. As far as we know, this work is the first in the literature to achieve algorithms for both weighted versions with ratios strictly greater than 0.5.

To achieve non-trivial ratios, different variants of the Ranking algorithm have been investigated for various matching problems [12, 1, 6, 5]. We analyze the following variant that is relevant to NwObMP on arbitrary graphs.

Weighted Ranking Algorithm for NwObMP. Given the node weights w, the algorithm determines a distribution $\mathcal{D}_w$ on permutations of V. It samples a permutation π from $\mathcal{D}_w$, and returns a list L of unordered pairs according to the lexicographical order induced by π, where nodes appearing earlier in the permutation have higher priority. Specifically, for a permutation $\pi : V \to [n]$, given two pairs e_1 and e_2 (where for each j, $e_j = \{u_j, v_j\}$ and $\pi(u_j) < \pi(v_j)$), the pair e_1 has higher priority than e_2 if (i) $\pi(u_1) < \pi(u_2)$, or (ii) $u_1 = u_2$ and $\pi(v_1) < \pi(v_2)$.

Sampling a permutation. Previous works [1, 6] have considered the following way to sample a permutation of nodes. The algorithm uses an *adjustment function* $\varphi(t) := 1 - e^{t-1}$ for $t \in [0, 1]$, and samples a *configuration* $\sigma \in \Omega_\infty := [0, 1]^V$ uniformly at random, i.e., each node u receives independently a random number $\sigma(u)$ in $[0, 1]$ uniformly at random. A permutation is given by sorting the nodes in non-increasing order of the *adjusted weight* $w(\sigma, u) := \varphi(\sigma(u)) \cdot w_u$. Observe that for the unweighted case (i.e., all nodes have the same weight), this is equivalent to sampling a permutation uniformly at random. We consider different adjustment functions φ in this paper.

1.1 Summary of Our Results.

Extending previous linear programming (LP) approaches [1, 13, 11, 5], we prove that a weighted Ranking algorithm has ratio greater than 0.5 for NwObMP with arbitrary node weights in general graphs.

▶ **Theorem 1** (Weighted Ranking for NwObMP). *For $m = 10000$, weighted* Ranking *using the discrete sample space $[0,1]_m^V$ (where $[0,1]_m := \{\frac{i}{m} : i \in [m]\}$ is a discretization of $[0,1]$) and adjustment function $\varphi(t) := 1 - \frac{e^{17t}-1}{e^{17}-1}$ has performance ratio at least 0.501505.*

In the analysis, we have discovered new structural properties of the Ranking algorithm. For instance, if a node has two unmatched neighbors, then it will still be matched even when its rank is demoted to the bottom. These properties enable us to form better LP constraints. We use continuous LP techniques to prove that the above ratio can be improved to 0.501512 if continuous random sample space $[0,1]^V$ is used (due to space constraints, the complete proof is deferred to the full version). Interestingly via these structural properties, we also improve the analysis of (unweighted) Ranking for the unweighted ObMP over the previous best ratio of 0.523166 in the SODA 2014 paper [5].

▶ **Theorem 2** (Ranking for Unweighted ObMP). *The* Ranking *algorithm for unweighted* ObMP *has performance ratio at least 0.526823.*

For EwObMP with a bounded number of distinct edge weights, we show that ratio strictly above 0.5 can be achieved by partitioning edges carefully according to the weights, and running the (unweighted) Ranking algorithm on each part.

▶ **Theorem 3** (EwObMP with Bounded Number of Distinct Weights). *Suppose there is an algorithm on unweighted* ObMP *with performance ratio $\frac{1}{2} + \xi_1$. Then, for each positive integer $k > 1$, there exists $\xi_k = \Omega(\xi_1)^{O(k^2)}$ such that the following holds. There exists an algorithm for* EwObMP *such that on instances with k distinct edge weights, the performance ratio is at least $\frac{1}{2} + \xi_k$.*

Our analysis is based on a new primal-dual framework of the standard matching LP known as *matching coverage*, in which dual feasibility is bypassed. Instead, only dual constraints corresponding to edges in an optimal matching are satisfied. Indeed the framework of matching coverage introduced for weighted oblivious matching has applications for other well-known problems. In particular using this framework we also design and analyze an algorithm for the *edge-weighted online bipartite matching problem with free disposal.* We prove that for the case of bounded online degrees, the ratio is strictly above 0.5.

EwOnBiMP with free disposal. An adversary fixes an edge-weighted bipartite graph $G(U \cup V, E)$ between a set U of *online* nodes and a set V of *offline* nodes, and determines the arrival order of the online nodes. When an online node u arrives, all the weights w_{uv}'s of edges between u and the offline nodes v in V are revealed to the (randomized) algorithm. The algorithm matches u to one of the offline nodes v. Even if an offline node v is already matched to a previous online node u', the algorithm is allowed to dispose of the edge $\{u', v\}$ and include the edge $\{u, v\}$ in the matching. The goal is to maximize the performance ratio, which is the (expected) sum of weights of edges in the final matching to that of a maximum weight matching in hindsight.

Feldman et al. [8] proved that a greedy algorithm can achieve ratio 0.5. We proposed a randomized algorithm that achieves ratio strictly greater than 0.5 for the case in which each online node has bounded degree.

▶ **Theorem 4** (EwOnBiMP with Bounded Online Degree). *There exists an algorithm for edge-weighted online bipartite matching with free disposal such that on instances in which every online node has degree at most* Δ*, the performance ratio is* $\frac{1}{2} + \Omega(\frac{1}{\Delta^2})$.

1.2 Related Work.

Unweighted ObMP. For the unweighted version, Dyer and Frieze [7] showed that the performance ratio is $0.5 + o(1)$ when the permutation of unordered pairs is chosen uniformly at random. In the mid-1990s, Aronson et al. [2] showed that the Modified Randomized Greedy (MRG) algorithm has ratio $0.5 + \epsilon$ (where $\epsilon = \frac{1}{400000}$). Goel and Tripathi [10] showed a hardness result of 0.7916 for any algorithm and 0.75 for adaptive vertex-iterative algorithms. In a recent SODA 2014 paper, Chan et al. [5] proved that Ranking algorithm has performance ratio at least 0.523166. We improve their analysis and performance ratio in this paper.

A version of the ranking algorithm was first proposed by Karp et al. [12] to solve the online bipartite matching problem (OnBiMP) with ratio $1 - \frac{1}{e}$. Subsequent works by Goel and Mehta [9], and Birnbaum and Mathieu [3] simplified the proof. Since the arrival order of online nodes is arbitrary, the same analysis carries over to obtain the same ratio for ObMP on bipartite graphs.

Since running Ranking on bipartite graphs for ObMP is equivalent to running the ranking algorithm for OnBiMP with *random arrival order*, the result of Karande et al. [11] implies that the ranking algorithm has a ratio at least 0.653 for the ObMP on bipartite graphs. Mahdian and Yan [13] improved the ratio to 0.696 using the technique of strongly factor-revealing LP. Karande et al. [11] also constructed a hard instance in which Ranking performs no better than 0.727.

Weighted Ranking. Aggarwal et al. [1] showed that the ranking algorithm can be applied to OnBiMP when the offline nodes have general weights. They proved that the performance ratio is $1 - \frac{1}{e}$. Devanur et al. [6] gave an alternative proof using randomized primal-dual analysis. We observe that their analysis can be applied to the NwObMP on bipartite graphs. Since their analysis assumes that the online nodes arrive in arbitrary order, by exchanging the roles of online and offline nodes for both partition of nodes, it can be shown that weighted Ranking achieves the same ratio of $1 - \frac{1}{e}$ on bipartite graphs.

EwOnBiMP with Free Disposal. Feldman et al. [8] proposed the free disposal feature for EwOnBiMP. They considered the setting in which each offline node v has capacity $n(v)$, and an online algorithm benefits from the $n(v)$ highest-weighted edges matched to v. They proposed an online algorithm with ratio $1 - \frac{1}{e_k}$, where $e_k = (1 + \frac{1}{k})^k$, and k is a lower bound on capacities. Thus, the proposed algorithm has performance ratio $\frac{1}{2}$ for the classic weighted version, when all capacities are 1.

1.3 Analyzing NwObMP via Linear Programming

A common technique [1, 11, 13, 10, 5] for analyzing Ranking algorithms is to define variables capturing the behavior of the algorithm in question, and derive structural properties that translate into constraints on the variables. A minimization LP with the performance ratio as the objective expressed in terms of the variables gives a lower bound on the ratio of the algorithm.

Let Ω be the sample space of configurations from which the algorithm derives its randomness. An *instance* $(\sigma, u) \in \Omega \times V$ is *good* if node u is matched when the algorithm is run with σ, and *bad* otherwise. We first describe the challenges encountered when previous techniques are applied to the node-weighted version of the problem on general graphs.

- *Why is the problem difficult on general graphs (as opposed to bipartite graphs)?* Bipartite graphs have the following nice property. Suppose in configuration σ, node u is unmatched, while its partner u^* in the optimal matching is matched to some node v. If the rank of u is promoted to form configuration σ', then u^* will be matched to some node v' such that the adjusted weight $w(\sigma', v') \geq w(\sigma, v)$ does not decrease. This naturally gives a way to relate the bad instance (σ, u) to the good instance (σ', v') [12, 11, 13, 1, 6], but unfortunately this property does not hold in general graphs. In fact, u^* might be unmatched in σ' as a result of u's promotion.
- *Why is the problem difficult when nodes have arbitrary weights (as opposed to uniform weight)?* In previous work [5] on the unweighted case, when u^* is matched in σ' in the above scenario, it is argued that the bad instance (σ, u) can be related to the good instance (σ', v), where v is matched in σ' to u^*. However, there is no guarantee that the adjusted weight $w(\sigma', v)$ of the good instance is at least $w(\sigma, u)$, which is needed as in [1, 6] to analyze the ratio for the weighted version.

To overcome the difficulties mentioned above, we have exploited the following structural properties of the Ranking algorithm. We analyze how the resulting matching would change if the rank of one node is changed (in Lemma 13), and give finer classification of good instances. In particular, the following notions are useful for relating bad instances to good instances in order to form LP constraints.

- **Graceful Instance.** A good instance (σ, u) is *graceful* if u is currently matched to a node v such that its optimal partner v^* does not exist or is also matched in σ.
- **Perpetual Instance.** If in a good instance (σ, u), node u has two unmatched neighbors, then (σ, u) is *perpetually good* in the sense that u will still be matched even when its rank is demoted to the bottom.

Breaking 0.5 Ratio for NwObMP. As in [1], we analyze the discrete sample space $\Omega_m := [m]^V$ (with the adjustment function $\varphi(t) := 1 - \frac{e^{17t}-1}{e^{17}-1}$, $\psi(i) := \varphi(\frac{i}{m})$ and adjusted weight $w(\sigma, u) := \psi(\sigma(u)) \cdot w_u$), and show that the performance ratio of weighted Ranking is at least the optimal value of some finite LP_m^ψ with m variables. Since LP_m^ψ does not depend on the size of G, computing the optimal value of LP_m^ψ for some large enough m is sufficient to prove a lower bound on the ratio of weighted Ranking. We show in our full version that a slightly better ratio can be analyzed using continuous LP for the limiting case as m tends to infinity.

We are aware of other adjustment functions that can achieve even slightly better ratios for the weighted Ranking, but we just present here a simple form that crosses the 0.5 barrier. Our result for the node-weighted case achieves the first non-trivial performance ratio that is strictly larger than 0.5.

Improved Ratio for Unweighted ObMP. We also apply our new combinatorial analysis to derive a new finite LP_n^U, that gives a lower bound on the performance ratio of unweighted Ranking running on graphs of size n. For the unweighted version of the problem, the limiting behavior of LP_n^U is analyzed when n tends to infinity and an improved lower bound on the performance ratio of unweighted Ranking is proved using a new class of continuous LP with jump discontinuity. The ideas for formulating the constraints are similar to the node-weighted case and we defer the proof to the full version.

1.4 Analyzing EwObMP and EwOnBiMP via Matching Coverage

Researchers have successfully applied the primal-dual LP framework to design approximation algorithms for matching problems [4, 6]. Consider the following standard maximum weight matching LP relaxation for an undirected graph $G = (V, E)$ with non-negative edge weights. Its dual is known as *vertex cover*.

$$\begin{aligned} \max \quad & w(x) := \sum_{\{u,v\} \in E} w_{uv} x_{uv} && (1) \\ \text{s.t} \quad & \sum_{u:\{u,v\} \in E} x_{uv} \leq 1, && \forall v \in V \\ & x_{uv} \geq 0, && \forall \{u,v\} \in E \end{aligned}$$

$$\begin{aligned} \min \quad & C(\alpha) := \sum_{u \in V} \alpha_u && (2) \\ \text{s.t} \quad & \alpha_v + \alpha_u \geq w_{uv}, && \forall \{u,v\} \in E \\ & \alpha_v \geq 0, && \forall v \in V \end{aligned}$$

An integral feasible primal solution x indicates whether an edge is selected and corresponds to some matching M, whose weight is denoted by $w(M) := w(x)$. When G is a bipartite graph between U and V, we use α_u for the variables for nodes in U and β_v for those corresponding to V.

Standard Primal-Dual Analysis. Typically, during the execution of an algorithm, both a primal and a dual solution are constructed. To analyze the approximation ratio, the value of the primal solution returned by the algorithm is compared with that of the dual solution. Since the primal is a maximization problem, any feasible dual provides an upper bound on the optimal primal value and can guarantee some approximation ratio. Hence, it is crucial in such a framework to establish the feasibility of the dual solution, for instance by either ensuring feasibility during construction, or scale the dual solution at the end by some appropriate factor. Dual feasibility requires that, for every edge in the graph, the sum of the dual values of its incident nodes is large enough.

New Framework. We observe that this strict requirement of dual feasibility is an artifact of the approximation analysis, and instead explore a new analysis method in which dual feasibility can be bypassed. Specifically, we use this new approach for different variations of edge-weighted maximum matching, and call it *matching coverage*. To emphasize that we do not achieve dual feasibility of any kind, we use a *vector* to mean an assignment of a non-negative value to each node.

▶ **Definition 5** (Matching Coverage)**.** Let M be a matching in graph G. A vector $\alpha \in \mathbb{R}^V$ is a *matching coverage* for matching M if α is non-negative, and the dual constraints of LP (2) corresponding to the edges of M are satisfied. In other words, for each $\{u, v\} \in M$, $\alpha_u + \alpha_v \geq w_{uv}$.

▶ Remark. Since any two distinct edges in a matching do not share any node, it follows that if a vector α is a matching coverage for a matching M, then $C(\alpha) \geq w(M)$.

General Framework of Matching Coverage. In our new analysis framework, the algorithm does not construct any dual solution (not even an infeasible one). This is a major departure from the conventional primal-dual framework in which some dual solution is usually constructed by an algorithm, whereas in our approach, the vector is used only for analysis. In the analysis, we imagine that as an algorithm ALG is executed, a vector α is constructed alongside with the knowledge of an optimal matching M^*. The idea is that the values in α are increased just enough to make sure that α is a matching coverage for M^*.

Why does this help the analysis? Since the vector α is a matching coverage for M^*, by Remark 1.4, we have $w(M^*) \leq C(\alpha)$. As α does not have to be feasible for all edge constraints, it is possible that the resulting value $C(\alpha)$ could be smaller than that of a feasible dual. Therefore, we can hope to get a smaller value of F when we compare $C(\alpha) \leq F \cdot w(M_{\mathsf{ALG}})$ with the weight of the matching M_{ALG} returned by ALG, thereby getting a larger performance ratio $w(M_{\mathsf{ALG}}) \geq \frac{1}{F} \cdot C(\alpha) \geq \frac{1}{F} \cdot w(M^*)$.

We use the framework of matching coverage to design and analyze algorithms for the following problems.

- EwObMP. In Section 4, we present an algorithm that achieves ratio strictly greater than 0.5 when the number of distinct edge weights is bounded. The full analysis is included in our full version.
- EwOnBiMP with Free Disposal. We present and analyze (in our full version) an algorithm that achieves ratio strictly greater than 0.5 when the online nodes have bounded degree. We show that without the free disposal assumption, no randomized algorithm can achieve any non-trivial constant guarantee on the ratio.

2 Defining Variables for Weighted Ranking on NwObMP

An adversary commits to a graph $G = (V, E)$ with $n = |V|$ nodes, where each node u has a non-negative weight w_u. We fix some maximum weight matching OPT in G. When the context is clear, we also use OPT to denote the set of nodes covered by the matching. Observe that in general OPT might be a proper subset of V. Let $w(\mathsf{OPT}) = \sum_{u \in \mathsf{OPT}} w_u$ be the total weight of OPT. For any $u \in V$, if u is matched in OPT, then we denote by u^* the partner of u in OPT, and we call u^* the *optimal partner* of u. If $u \notin \mathsf{OPT}$, then we say that u^* does not exist.

Weighted Ranking. As described in the introduction, the algorithm derives its randomness by sampling from $\Omega_m := [m]^V$ uniformly at random, where m is a sufficiently large integer and $[m] = \{1, 2, \ldots, m\}$. (We omit the subscript for Ω when the context is clear.) This is equivalent to picking $\sigma(u) \in [m]$ uniformly at random and independently for each $u \in V$. As in [1, 6], the algorithm fixes an *adjustment function* $\varphi : [0, 1] \to [0, 1]$ that is non-increasing. The function $\varphi(t) := 1 - e^{t-1}$ is used in [1, 6]. We shall consider other adjustment functions such that $\varphi(1) = 0$ also holds.

We denote $\psi(i) := \varphi(\frac{i}{m})$. Then, a permutation on V is induced by σ by sorting the nodes in non-increasing order of *adjusted weight* $w(\sigma, u) := \psi(\sigma(u)) \cdot w_u$, where ties are resolved deterministically (for instance by the identities of the nodes). This permutation on V induces a lexicographical order on the node pairs that is used for probing. We denote $(\sigma, u) > (\sigma, v)$ when node u comes before v in the permutation induced by σ, in which case u has higher priority than v.

We denote $\mathcal{U} := \Omega \times V$ as the set of *instances*. Let $M(\sigma)$ be the matching obtained when Ranking is run with configuration σ. If u is matched to some v after running Ranking with configuration σ, then we say that u is matched in σ and v is the (current) partner of u in σ. An instance (σ, u) is *good* if u is matched in σ, and otherwise *bad*. An event is a subset of instances.

Given $\sigma \in \Omega_m$, let σ_u^j be obtained by setting $\sigma_u^j(u) = j$ and $\sigma_u^j(v) = \sigma(v)$ for all $v \neq u$.

▶ **Definition 6** (Events)**.** For each $i \in [m]$, define the following:

- Rank-i good event: $Q_i := \{(\sigma, u) | \sigma(u) = i \textit{ and } u \textit{ is matched in } \sigma\}$
- Rank-i bad event: $R_i := \{(\sigma, u) | \sigma(u) = i, u \textit{ is not matched in } \sigma \text{ and } u \in \mathsf{OPT}\}$

Let $Q := \cup_{i \in [m]} Q_i$ and $R := \cup_{i \in [m]} R_i$.

Notice that Q_i and R_i are disjoint. While Q_i could involve nodes that are not in OPT, R_i only involves nodes in OPT; this idea also appears in [1] for dealing with the case when OPT is a proper subset of V. Define $x_i := \frac{\sum_{(\sigma,u)\in Q_i} w_u}{w(\mathsf{OPT})\cdot m^{n-1}}$, which can be interpreted as the conditional expected contribution of the nodes given that they are at rank i. We next derive some properties of the x_i's.

- **Monotonicity.** For $i \geq 2$, we have $x_{i-1} \geq x_i \geq 0$, since if $(\sigma, u) \in Q_i$, then $(\sigma_u^{i-1}, u) \in Q_{i-1}$. However, $1 \geq x_1$ does not necessarily hold since there may exist $u \notin \mathsf{OPT}$ and $(\sigma, u) \in Q_1$.
- **Loss due to unmatched nodes.** Similar to x_i associated with Q_i, we consider an analogous quantity associated with R_i:

$$\begin{aligned}\overline{x}_i :=& \frac{\sum_{(\sigma,u)\in R_i} w_u}{w(\mathsf{OPT})\cdot m^{n-1}} = \frac{\sum_{(\sigma,u)\in Q_i\cup R_i} w_u - \sum_{(\sigma,u)\in Q_i} w_u}{w(\mathsf{OPT})\cdot m^{n-1}} \\ \geq& \frac{w(\mathsf{OPT})\cdot m^{n-1} - \sum_{(\sigma,u)\in Q_i} w_u}{w(\mathsf{OPT})\cdot m^{n-1}} = 1 - x_i, \end{aligned} \tag{3}$$

  where the inequality $\sum_{(\sigma,u)\in Q_i\cup R_i} w_u \geq w(\mathsf{OPT})\cdot m^{n-1}$ could be strict because Q_i might involve nodes not in OPT.
- **Performance Ratio.** The performance ratio is the expected sum of weights of matched nodes divided by $w(\mathsf{OPT})$, which is given by $\frac{\sum_{(\sigma,u)\in Q} w_u}{w(\mathsf{OPT})\cdot m^n} = \frac{1}{m}\sum_{i=1}^m x_i$.

▶ **Definition 7** (Marginally Bad Event). For $i \in [m]$, we define rank-i marginally bad event as follows. Let $S_1 := R_1$; for $i \geq 2$, let $S_i := \{(\sigma, u) \in R_i | (\sigma_u^{i-1}, u) \in Q_{i-1}\}$.

Let $S := \cup_{i\in[m]} S_i$ and $\alpha_i := \frac{\sum_{(\sigma,u)\in S_i} w_u}{w(\mathsf{OPT})\cdot m^{n-1}}$ for all $i \in [m]$.

Observe that for an instance (σ, u) such that (σ_u^m, u) is bad, there exists a unique $j \in [m]$ such that $(\sigma_u^j, u) \in S_j$, and we say that j is the *marginal position* of (σ, u).

Relating x_i's and α_i's. From a marginally bad instance $(\sigma, u) \in S_i$, node u will be matched when its rank is promoted to $i - 1$. Hence, for $i \geq 2$, we immediately have

$$\alpha_i \leq \frac{\sum_{(\sigma,u)\in Q_{i-1}} w_u - \sum_{(\sigma,u)\in Q_i} w_u}{w(\mathsf{OPT})\cdot m^{n-1}} = x_{i-1} - x_i. \tag{4}$$

Moreover, for $i \in [m]$, any bad instance $(\sigma, u) \in R_i$ has a unique marginal position $j \in [i]$ such that $(\sigma_u^j, u) \in S_j$; for each $(\sigma, u) \in S_j$ such that $j \leq i$, we also have $(\sigma_u^i, u) \in R_i$. Hence, there is a one-one correspondence between R_i and $\cup_{j=1}^i S_j$, and so we have:

$$\sum_{j=1}^{i} \alpha_j = \frac{\sum_{j=1}^i \sum_{(\sigma,u)\in S_j} w_u}{w(\mathsf{OPT})\cdot m^{n-1}} = \frac{\sum_{(\sigma,u)\in R_i} w_u}{w(\mathsf{OPT})\cdot m^{n-1}} = \overline{x}_i \geq 1 - x_i. \tag{5}$$

▶ Remark. Observe that when all nodes in V are covered by OPT, equality holds for both (4) and (5). In fact, Lemma 8 allow us to remove the α_i's from the LP constraints.

▶ Fact 1 (Ranking is Greedy). Suppose Ranking is run with configuration σ. If (σ, u) is bad, then each neighbor of u (in G) is matched in σ to some node v such that $(\sigma, v) > (\sigma, u)$.

3 Analyzing NwObMP Using Graceful and Perpetual Instances

In this section we define some relations from (marginally) bad events to good events to formulate our LP constraints. We describe a general principle which is a weighted version of the argument used in [5].

As mentioned above, the following lemma is used to remove the α_i's from the LP constraints.

▶ **Lemma 8.** *Suppose that* $\{b_i\}_{i=1}^{m+1}$ *is non-negative and non-increasing such that* $b_{m+1} = 0$*, and* $\{c_i\}_{i=1}^{m+1}$ *is non-decreasing such that* $c_1 = 0$*. Then, we have*
(a) $\sum_{i=1}^m b_i\alpha_i \geq b_1 - \sum_{i=1}^m (b_i - b_{i+1})x_i$.
(b) $\sum_{i=1}^m b_i c_i \alpha_i \geq -\sum_{i=1}^m (b_i c_i - b_{i+1}c_{i+1})x_i$.

Proof. Statement (a) follows because
$\sum_{i=1}^m b_i\alpha_i = \sum_{i=1}^m (b_i - b_{i+1}) \sum_{j=1}^i \alpha_j \geq \sum_{i=1}^m (b_i - b_{i+1})(1 - x_i) = b_1 - \sum_{i=1}^m (b_i - b_{i+1})x_i$,
where the inequality comes from (5).

For statement (b), observing that $c_1 = 0$, we can assume that $\alpha_1 = x_0 - x_1$, where $x_0 = 1$. Let $C = \max_i c_i$, and define $d_i := C - c_i \geq 0$. Then, we have
$\sum_{i=1}^m b_i c_i \alpha_i = \sum_{i=1}^m C b_i \alpha_i - \sum_{i=1}^m b_i d_i \alpha_i \geq C b_1 - C \sum_{i=1}^m (b_i - b_{i+1})x_i - \sum_{i=1}^m b_i d_i (x_{i-1} - x_i) = -\sum_{i=1}^m (b_i c_i - b_{i+1}c_{i+1})x_i$,
where in the inequality we apply statement (a) to the first term (which is still valid because $\alpha_1 \geq 1 - x_1$ holds), and apply $\alpha_1 = x_0 - x_1$ and (4) to the second term. ◀

Weighting Principle. Suppose f is a relation from subset A to subset B of instances, where $f(a)$ is the set of elements in B that are related to $a \in A$, and $f^{-1}(b)$ is the set of elements in A that are related to $b \in B$. Recall that each instance $a = (\sigma, u)$ has adjusted weight $w(a) = w(\sigma, u)$. Suppose further that for all $a \in A$, for all $b \in f(a)$, $w(a) \leq w(b)$. Then, by considering the bipartite graph H induced by f on $A \cup B$, and comparing the weights of end-points for each edge in H, it follows that $\sum_{a\in A} |f(a)| \cdot w(a) \leq \sum_{b \in B} |f^{-1}(b)| \cdot w(b)$.

We shall formulate constraints by considering relations between subsets of instances. The injectivity of a relation f is the minimum integer q such that for all $b \in B$, $|f^{-1}(b)| \leq q$. In this case, we have

$$\textstyle\sum_{a\in A} |f(a)| \cdot w(a) \leq q \sum_{b\in B} w(b). \tag{6}$$

3.1 Demoting Marginally Bad Instances

▶ **Lemma 9.** *We have:* $\frac{1}{m}\sum_{i=1}^m [2\psi(i) + (m-i)(\psi(i) - \psi(i+1))]x_i \geq \psi(1)$.

Proof. We define a relation f from the set S of marginally bad instances to the set Q of good instances. Observe that for a (marginally) bad instance (σ, u), u is unmatched in σ and its optimal partner u^* exists. If we further demote u by setting its rank to $j \geq \sigma(u)$, the resulting matching is unchanged. Therefore, by Fact 1, for each $j \geq \sigma(u)$, u^* is matched to the same v such that $w(\sigma, u) \leq w(\sigma, v) = w(\sigma_u^j, v)$. Hence, we can define $f(\sigma, u) := \{(\sigma_u^j, v) | u^* \textit{ is matched to } v \textit{ in } \sigma_u^j, j \geq \sigma(u)\} \subseteq Q$, where $|f(\sigma, u)| = m - \sigma(u) + 1$, and $w(\sigma, u) \leq w(\sigma', v)$ for all $(\sigma', v) \in f(\sigma, u)$.

We next check the injectivity of f. Suppose $(\rho, v) \in f(\sigma, u)$. Then, u^* is the current partner of v in ρ, and this uniquely determines u, which is unmatched in ρ. Hence, $\sigma = \rho_u^j$, where j is uniquely determined as the marginal position of (ρ, u). Therefore, the injectivity is 1.

Hence, our weighting principle (6) gives the following:

$$\sum_{i=1}^m \sum_{(\sigma,u)\in S_i} (m - i + 1)\psi(i) w_u = \sum_{a\in S} |f(a)| \cdot w(a) \leq \sum_{b\in Q} w(b) = \sum_{i=1}^m \sum_{(\rho,v)\in Q_i} \psi(i) w_v.$$

Dividing both sides by $w(\mathsf{OPT}) \cdot m^n$ gives $\frac{1}{m}\sum_{i=1}^m (m-i+1)\psi(i)\alpha_i \leq \frac{1}{m}\sum_{i=1}^m \psi(i) x_i$.

Since we do not wish α_i's to appear in our constraints, we derive a lower bound for the LHS in terms of x_i's by applying Lemma 8 with $b_i := (m - i + 1)\psi(i)$, where $\psi(m + 1)$ can be chosen to be any value. Rearranging gives the required inequality. ◀

3.2 Promoting Marginally Bad Instances

▶ **Lemma 10.** *We have:* $\frac{2}{m}\sum_{i=1}^{m}\psi(i) \cdot x_m + \frac{1}{m}\sum_{i=1}^{m}[5\psi(i) - i(\psi(i+1) - \psi(i))] \cdot x_i \geq \frac{3}{m}\sum_{i=1}^{m}\psi(i)$.

To use the weighting principle, we shall define relations from marginally bad instances S to the following subsets of special good instances.

▶ **Definition 11** (*If v is matched, would v^* still be matched?*)**.** For $i \in [m]$, let the graceful instances be $Y_i := \{(\sigma, u) \in Q_i | u$ is matched in σ to some v s.t. v^* does not exist or is also matched in $\sigma\}$. Let $y_i := \frac{\sum_{(\sigma,u)\in Y_i} w_u}{w(\mathsf{OPT})\cdot m^{n-1}}$ and $Y := \cup_{i\in[m]} Y_i$.

▶ **Definition 12** (*You will be matched even at the bottom*)**.** For $i \in [m]$, let the perpetual instances be $Z_i = \{(\sigma, u) \in Q_i | (\sigma_u^m, u) \in Q_m\}$. Let $z_i = \frac{\sum_{(\sigma,u)\in Z_i} w_u}{w(OPT)\cdot m^{n-1}}$ and $Z := \cup_{i\in[m]} Z_i$.

By definition, we know that $Y_i \subseteq Q_i$ and hence $x_i \geq y_i \geq 0$. Moreover, observing that there exists a bijection between Z_i and Q_m that maps each $(\sigma, u) \in Z_i$ to $(\sigma_u^m, u) \in Q_m$, we have $z_i = x_m$.

Suppose (σ, u) is a good instance that has marginal position j. We wish to compare the matchings produced by σ and σ_u^j. Sometimes it is more convenient to consider an unmatched node as being ignored. Specifically, given a configuration σ and a node u, running Ranking with σ_u means that we still use σ to generate the probing order, but any edge involving u is ignored. Observe that if (σ, u) has a marginal position j, then σ_u and σ_u^j will produce the same matching.

▶ **Lemma 13** (Ignoring One Node)**.** *Suppose u is covered by the matching $M(\sigma)$ produced by σ, and $M(\sigma_u)$ is the matching produced by using the same probing list, but any edge involving u is ignored. The symmetric difference $M(\sigma) \oplus M(\sigma_u)$ is an alternating path $P = (u = u_1, u_2, \ldots, u_p)$ such that for all $i \in [p-2]$, $(\sigma, u_i) > (\sigma, u_{i+2})$.*

Proof. We can view probing G with σ_u as using the same list L of unordered node pairs to probe another graph G_u, which is the same as G except that the node u is labelled *unavailable* and will not be matched in any case. After each round of probing, we compare what happens to the partially constructed matchings $M(\sigma)$ in G and $M(\sigma_u)$ in G_u. For the sake of this proof, "unavailable" and "matched" are the same *availability status*, while "unmatched" is a different availability status.

We apply induction on the number of rounds of probing. Observe that the following invariants hold initially. (i) There is exactly one node known as the *crucial* node (which is initially u) that has different availability in G and G_u. (ii) The symmetric difference $M(\sigma) \oplus M(\sigma_u)$ is an alternating path P connecting u to the current crucial node; initially, both $M(\sigma)$ and $M(\sigma_u)$ are empty, and path P is degenerate and contains only u. (iii) If the path $P = (u = u_1, u_2, \ldots, u_l)$ contains $l \geq 3$ nodes, then for all $i \in [l-2]$, then $(\sigma, u_i) > (\sigma, u_{i+2})$.

Consider the inductive step. Suppose currently the alternating path $M(\sigma) \oplus M(\sigma_u)$ contains l nodes, where u_l is crucial. Observe that the crucial node and $M(\sigma) \oplus M(\sigma_u)$ do not change in a round except for the case when the pair being probed is an edge in G (and G_u), involving the crucial node u_l with another currently unmatched node u_{l+1} in G, which is also unmatched in G_u (because the induction hypothesis states that all nodes but u_l have the same availability status in G and G_u).

Since u_l has different availability in G and G_u, but u_{l+1} is unmatched in both G and G_u, it follows that the edge $e := \{u_l, u_{l+1}\}$ is added to exactly one of $M(\sigma)$ and $M(\sigma_u)$. Hence, the edge e is added to extend the alternating path $M(\sigma) \oplus M(\sigma_u)$, and the node u_{l+1} becomes crucial.

Next, it remains to show that if $l \geq 2$, then $(\sigma, u_{l-1}) > (\sigma, u_{l+1})$. Suppose we go back in time, and consider at the beginning of the round when the edge $\{u_{l-1}, u_l\}$ is about to be probed, and u_{l-1} is crucial. By the induction hypothesis, both u_l and u_{l+1} are unmatched in both G and G_u. It follows that $(\sigma, u_{l-1}) > (\sigma, u_{l+1})$, because otherwise the edge $\{u_{l-1}, u_l\}$ would have lower probing priority than $\{u_{l+1}, u_l\}$. This completes the inductive step. ◀

▶ **Lemma 14** (Two Unmatched Neighbors Implies Perpetual)**.** *Suppose in configuration σ, node u is matched and has two unmatched neighbors in G. Then, $(\sigma, u) \in Z$ is perpetual.*

Proof. If we assume the opposite, then u will be unmatched in σ_u^m. Suppose x and y are two neighbors of u that are unmatched in σ. Then, by Lemma 13, the symmetric difference $M(\sigma) \oplus M(\sigma_u^m)$ is an alternating path starting from u, and hence at most one of x and y will remain unmatched in σ_u^m.

This implies that in σ_u^m, the unmatched node u will have at least one unmatched neighbor; this contradicts the fact that that Ranking will always produce a maximal matching. ◀

Next we derive inequalities involving the graceful instances. Combining the inequalities, we can obtain the crucial constraint involving only x_i's for achieving a ratio that is strictly larger than 0.5.

▶ **Lemma 15** (*You are unmatched because someone is not graceful.*)**.** *We have the following inequality:* $\frac{1}{m}\sum_{i=1}^{m} \psi(i) y_i \leq \frac{1}{m}\sum_{i=1}^{m} \psi(i)(2x_i - 1)$.

Proof. We define a relation from the set R of bad instances to the set $Q \setminus Y$ of good instances that are not graceful.

Given any bad instance $(\sigma, u) \in R$, we know that u^* exists and is matched to some node v such that $w(\sigma, v) \geq w(\sigma, u)$, by Fact 1. Moreover, since v is matched to u^* such that u is unmatched, we know that $(\sigma, v) \in Q \setminus Y$ is good but not graceful. Hence, we define $f(\sigma, u) := \{(\sigma, v)\}$, where v is the current partner of u^*. Observe that each $(\sigma, v) \in Q \setminus Y$ can be related to a unique $(\sigma, u) \in R$, where u is the optimal partner of v's current partner in σ. Hence, the injectivity of f is 1.

Hence, the weighting principle (6) gives: $\sum_{(\sigma,u)\in R} w(\sigma, u) \leq \sum_{(\sigma,v)\in Q\setminus Y} w(\sigma, v)$. Dividing both sides by $w(\mathsf{OPT}) \cdot m^n$ gives: $\frac{1}{m}\sum_{i=1}^{m} \psi(i)\overline{x}_i \leq \frac{1}{m}\sum_{i=1}^{m} \psi(i)(x_i - y_i)$.

Finally, using $\overline{x}_i \geq 1 - x_i$ from (3) and rearranging gives the required inequality. ◀

▶ **Lemma 16** (*If you are marginal, someone else is either graceful or perpetual*)**.** *We have the inequality:* $\frac{1}{m}\sum_{i=1}^{m}(i-1)\psi(i)\alpha_i \leq \frac{1}{m}\sum_{i=1}^{m} \psi(i)(3y_i + 2z_i)$.

Proof. As mentioned earlier, we shall define two relations f and g from marginally bad S to graceful Y and perpetual Z, respectively, such that the following properties hold.

1. For each $a \in S$, for each $b \in f(a) \cup g(a)$, $w(a) \leq w(b)$.
2. For each $a \in S$, $|f(a)| + |g(a)| = \sigma(u) - 1$.
3. The injectivity of f is at most 3 and the injectivity of g is at most 2.

Suppose we have f and g with these properties. Then, our weighting principle (6) gives:

$$\sum_{(\sigma,u)\in S}(\sigma(u)-1)w(\sigma,u) \le \sum_{(\rho,v)\in Y} 3w(\rho,v) + \sum_{(\rho,v)\in Z} 2w(\rho,v),$$

which by definition is equivalent to

$$\sum_{i=1}^{m}(i-1)\psi(i)\sum_{(\sigma,u)\in S_i} w_u \le \sum_{i=1}^{m}\psi(i)(3\sum_{(\rho,v)\in Y_i} w_u + 2\sum_{(\rho,v)\in Z_i} w_u).$$

Dividing both sides by $w(\mathsf{OPT})\cdot m^n$ gives the required inequality.

Next we show how f and g are constructed such that all required properties hold.

Given marginally bad $(\sigma,u)\in S$, we consider good instance $(\sigma',u)\in Q$, where $\sigma' = \sigma_u^j, j<\sigma(u)$ is obtained by "promoting" u's rank in σ. Note that by Fact 1, u^* must be matched in σ to some node v_0 such that $(\sigma,v_0) > (\sigma,u)$. Let the partner of u in σ' be p. The next claim is crucial for the construction of f and g.

▶ Claim 3.1. If $w(\sigma',p) < w(\sigma,u)$, then u^* is matched in σ' to some node v such that $w(\sigma',v)\ge w(\sigma,v_0)\ge w(\sigma,u)$.

Proof. By Lemma 13, we know that the symmetric difference $M(\sigma')\oplus M(\sigma)$ is an alternating path $(u=u_1, p=u_2, u_3, u_4\ldots)$ that starts with u. Moreover, we have $w(\sigma',u)\ge w(\sigma',u_3)\ge w(\sigma',u_5)\ge\ldots$ and $w(\sigma',p)\ge w(\sigma',u_4)\ge w(\sigma',u_6)\ge\ldots$. If u^* is not contained in the alternating path, then directly we have $v=v_0$ and hence the claim holds.

Assume that u^* is contained in the alternating path. Then, v_0 must also appear in the alternating path. Let $v_0=u_i$. Since $w(\sigma',v_0)=w(\sigma,v_0)\ge w(\sigma,u) > w(\sigma',p)$, we conclude that i must be odd. By Lemma 13, we know that u^* must be u_{i-1} since u_i is matched to u_{i-1} in σ. Moreover, we know that $u^*=u_{i-1}$ is matched to u_{i-2} in σ' such that $w(\sigma',u_{i-2})\ge w(\sigma',u_i)=w(\sigma,v_0)$. ◀

Next we include instances in Y into $f(\sigma,u)$ and instances in Z into $g(\sigma,u)$ on a case by case basis. Recall that for each $1\le j<\sigma(u)$, we consider $\sigma'=\sigma_u^j$; moreover, after promoting u to rank j, u is matched in σ' to p.

Case-1(a). u^* is matched in σ' and $w(\sigma',p)=w(\sigma,p)\ge w(\sigma,u)$. In this case, (σ',p) is graceful, because p is matched in σ' to u, whose optimal partner u^* is also matched. Hence, we include $(\sigma',p)\in Y$ in $f(\sigma,u)$.

Case-1(b). u^* is matched in σ' and $w(\sigma',p)=w(\sigma,p)<w(\sigma,u)$. By Claim 3.1, u^* is matched in σ' to some node v such that $w(\sigma',v)\ge w(\sigma,u)$. Observe that (σ',v) is graceful, and we include $(\sigma',v)\in Y$ in $f(\sigma,u)$.

Case-2(a). u^* is unmatched in σ', and p^* (if it exists) is also matched in σ'. Note that after promoting u, we have $w(\sigma',u)\ge w(\sigma,u)$. Moreover, (σ',u) is graceful, because the optimal partner p^* either does not exist or is matched in σ'. We include $(\sigma',u)\in Y$ in $f(\sigma,u)$.

Case-2(b). u^* is unmatched in σ', p^* exists and is the only unmatched neighbor of p in σ'. By Claim 3.1, since u^* is unmatched in σ', we have $w(\sigma,p)=w(\sigma',p)\ge w(\sigma,u)$; also, since p is matched in σ', $p\ne u^*$. Moreover, by Lemma 13, the symmetric difference $M(\sigma)\oplus M(\sigma')$ is an alternating path, and only two nodes (u and u^*) can have different matching status in σ and σ'.

Hence, in σ, p must remain matched and p^* must remain unmatched; this means that p has exactly two unmatched neighbors, namely u and p^*, in σ. By Lemma 14, we conclude that (σ,p) is perpetual, and include $(\sigma,p)\in Z$ in $g(\sigma,u)$.

Case-2(c). u^* is unmatched in σ', p^* exists and is not the only unmatched neighbor of p in σ'. Similar to Case-2(b), in this case, $w(\sigma', p) = w(\sigma, p) \geq w(\sigma, u)$ and p has two different unmatched neighbors in σ', so (σ', p) is perpetual by Lemma 14. We include $(\sigma', p) \in Z$ in $g(\sigma, u)$.

By construction, property 1 holds. Moreover, for each $1 \leq j < \sigma(u)$ and $\sigma' = \sigma_u^j$, exactly one of the above 5 cases happens. Hence, we also have property 2: $|f(\sigma, u)| + |g(\sigma, u)| = \sigma(u) - 1$. Next, we prove the injectivity.

Injectivity Analysis. Observe that in our construction, if $(\rho, v) \in f(\sigma, u) \cup g(\sigma, u)$, then $\sigma = \rho_u^t$, where t is the marginal position of (ρ, u). Hence, in the injectivity analysis, once (ρ, v) and u are identified, σ can be uniquely determined.

For relation f, suppose $(\rho, v) \in Y$ is included in some $f(\sigma, u)$ in the following cases.

- **Case-1(a).** Node u is uniquely identified as the current partner of v in ρ.
- **Case-1(b).** Node u is uniquely identified as the optimal partner of v's current partner.
- **Case-2(a).** Node u is the same as v.

Hence, each $(\rho, v) \in Y$ is related to at most 3 instances in S, which means that f has injectivity at most 3.

For relation g, suppose $(\rho, v) \in Z$ is included in some $g(\sigma, u)$ in the following cases.

- **Case-2(b).** By construction $\rho = \sigma$, and v has exactly two neighbors that are unmatched in ρ, one of which is v^*. Node u is uniquely identified as the other unmatched neighbor.
- **Case-2(c).** Node u is uniquely identified as the current partner of v in ρ.

Hence, each $(\rho, v) \in Z$ is related to at most 2 instances in S, which means that g has injectivity at most 2. This completes the proof of Lemma 16. ◀

We can now derive the main constraint of this subsection.

Proof of Lemma 10: We start from the inequality in Lemma 15. Observing that $z_i = x_m$, and using the upper bound for $\frac{1}{m}\sum_{i=1}^m \psi(i) y_i$ in Lemma 16, we have $\frac{1}{m}\sum_{i=1}^m (i-1)\psi(i)\alpha_i \leq \frac{1}{m}\sum_{i=1}^m \psi(i)(6x_i + 2x_m - 3)$.

We next use Lemma 8 by setting $b_i := \psi(i)$ and $c_i := i-1$; observe that $c_1 = 0$, and we set $\psi(m+1) := 0$, which is consistent with $\psi(m) \geq 0 = \psi(m+1)$. Hence, we have the following lower bound for the LHS: $\frac{1}{m}\sum_{i=1}^m (i-1)\psi(i)\alpha_i \geq \frac{1}{m}\sum_{i=1}^m (\psi(i) + i(\psi(i+1) - \psi(i))) \cdot x_i$.

Rearranging gives the required inequality. ∎

3.3 Using LP to Bound Performance Ratio

Putting all achieved constraints on x_i's together, we obtain the following linear program LP_m^ψ, which is a lower bound on the performance ratio when weighted Ranking is run with weight adjustment function ψ and sample space $\Omega_m = [m]^V$.

$$
\begin{aligned}
\mathsf{LP}_m^\psi \qquad & \min \quad \frac{1}{m}\sum_{i=1}^m x_i \\
& \text{s.t.} \quad x_i - x_{i+1} \geq 0, \qquad i \in [m-1] \\
\frac{2}{m}\sum_{i=1}^m \psi(i)\cdot x_m + \frac{1}{m}\sum_{i=1}^m [5\psi(i) - i(\psi(i+1) - \psi(i))]\cdot x_i &\geq \frac{3}{m}\sum_{i=1}^m \psi(i) \qquad (7) \\
\frac{1}{m}\sum_{i=1}^m [2\psi(i) + (m-i)(\psi(i) - \psi(i+1))] x_i &\geq \psi(1) \qquad (8) \\
x_i &\geq 0, \qquad i \in [m].
\end{aligned}
$$

Achieving ratio strictly larger than 0.5. Observe that LP_m^ψ is independent of the size of G. Hence, to obtain a lower bound on the ratio, we can use an LP solver to solve LP_m^ψ for some large enough m and some appropriate non-negative non-increasing sequence $\{\psi(i)\}_{i=1}^m$. In particular, there exists a weighted Ranking algorithm with ratio strictly above 0.5.

▶ **Theorem 17.** *Using $m = 10000$ and $\psi(i) := 1 - \frac{e^{\frac{17i}{m}}-1}{e^{17}-1}$, the weighted* Ranking *algorithm has performance ratio at least the value given by* LP_m^ψ*:* 0.501505*.*

Although the function $\varphi(t) := 1 - e^{t-1}$ (that is used in [1, 6]) cannot give a ratio better 0.5 from our LP, it is still possible that the function could have good performance ratio. More experimental results and our source code can be downloaded at:

`http://i.cs.hku.hk/~algth/project/online_matching/weighted.html`.

Limiting case when m tends to infinity. Experiments show that LP_m^ψ is increasing in m. This suggests that a (slightly) better analysis may be achieved if Ranking samples σ from the continuous space $\Omega_\infty = [0,1]^V$, and uses adjusted weight $w(\sigma, u) := \varphi(\sigma(u)) \cdot w_u$ for each node u.

The variables x_i's are replaced by the function $z(t) := \frac{\sum_{u \in V} \Pr_\sigma[(\sigma,u) \text{ is good} | \sigma(u) = t] \cdot w_u}{w(\mathsf{OPT})}$. Our combinatorial counting argument can be replaced by measure analysis. For instance, $\Omega_\infty = [0,1]^V$ is equipped with the uniform n-dimensional measure, while $z(t)$ has measure of dimension $n-1$. Since we assume that $\psi(m+1) = 0$ in the finite analysis, this corresponds to $\varphi(1) = 0$ in continuous case.

Observe that it is possible to describe a continuous version of the weighting principle using measure theory to derive all the corresponding constraints involving z. However, the formal rigorous proof is out of the scope of this paper, and one can intuitively see that each constraint involving the x_i's translates naturally to a constraint involving z in the limiting case. Hence, the following continuous $\mathsf{LP}_\infty^\varphi$ gives a lower bound on the ratio when Ranking samples continuously, and we analyze it in our full version as a case study.

$$\begin{array}{llrl}
\mathsf{LP}_\infty^\varphi & \min & \int_0^1 z(t)dt & \\
 & \text{s.t.} & z'(t) \le 0 & \forall t \in [0,1] \\
\multicolumn{3}{r}{2\Phi \cdot z(1) + \int_0^1 [5\varphi(t) - t\varphi'(t)]\, z(t)dt \ge 3\Phi} & \\
\multicolumn{3}{r}{\int_0^1 [2\varphi(t) - (1-t)\varphi'(t)]\, z(t)dt \ge \varphi(0)} & \\
 & & z(t) \ge 0 & \forall t \in [0,1] \\
 & & \Phi = \int_0^1 \varphi(t)dt. &
\end{array}$$

▶ **Theorem 18** (Weighted Ranking with Continuous Sampling)**.** *Using continuous sample space Ω_∞ (with adjustment function $\varphi(t) := 1 - \frac{e^{17t}-1}{e^{17}-1}$), weighted* Ranking *has performance ratio at least* 0.501512*.*

4 Beating Ratio 0.5 for EwObMP

We consider EwObMP where the number of distinct weights is k. We give an algorithm whose performance ratio is $\frac{1}{2} + \xi_k$, where ξ_k only depends on k. As a subroutine, we use an algorithm $\mathcal{A}^{un}$ for the unweighted version of the problem with performance ratio $\frac{1}{2} + \xi_1$, where $\xi_1 > 0$. For instance, Theorem 2 implies that $\xi_1 \ge 0.0268$. When we run $\mathcal{A}^{un}$ on a subset $H \subseteq \binom{V}{2}$, $\mathcal{A}^{un}$ is first run to produce a random order L of node pairs. Only pairs in H are kept in L, while pairs not in H are removed. Then, the list L is used for probing as before. We partition the pairs in $\binom{V}{2}$ into batches $\{H_i\}_{i \ge 1}$, where the weights of pairs in each batch are similar. Then, starting from the batch with largest weights, we run $\mathcal{A}^{un}$ on each batch H_i to produce a list L_i, and return the concatenated list used for probing.

The following lemma, whose proof can be found in the full version, describes the properties of the intervals picked by the algorithm. Recall that $\mathcal{A}^{un}$ has performance ratio $\frac{1}{2} + \xi_1$ on unweighted ObMP. Given two real numbers $a \le b$, we denote $\mathsf{dist}(a, b) := 1 - \frac{a}{b}$.

▶ **Lemma 19** (Partitioning Weights into Batches). *Given a set W of k distinct weights, there exists an integer $r = O(k^2)$ and $\epsilon = \frac{\xi_1}{2}$ such that the algorithm can return disjoint intervals $\{I_i := [a_i, b_i]\}_{i \geq 1}$, whose union contains W, and for each $i \geq 1$, $b_{i+1} < a_i$, $\mathsf{dist}(a_i, b_i) \leq \epsilon^r$ and $\mathsf{dist}(b_{i+1}, b_i) \geq \epsilon^{r-1}$.*

Algorithm 1 Algorithm for Edge-Weighted ObMP

1: $W \leftarrow \{w_e : e \in \binom{V}{2}\}$ ▷ Set of weights of pairs in $\binom{V}{2}$.
2: $\{I_i := [a_i, b_i]\}_{i=1}^{K} \leftarrow$ Disjoint intervals as given in Lemma 19 to partition W, where I_1 is the interval with the largest weights.
3: **for** i from 1 to K **do**
4: $\quad H_i \leftarrow$ Pairs in $\binom{V}{2}$ with weights in I_i
5: $\quad L_i \leftarrow$ List produced by running unweighted $\mathcal{A}^{un}$ on H_i using independent randomness
6: **return** concatenated list $L := L_1 \oplus L_2 \oplus \cdots \oplus L_K$

Assuming the knowledge of an optimal matching OPT, we construct a matching coverage $\alpha \in \mathbb{R}^V$ for OPT during an execution of the algorithm. For a matching M, we use $|M|$ to denote its cardinality and $w(M)$ to denote the sum of weights of its edges. We say an edge e in OPT is *destroyed* by a matching M if edge e is not in M but at least one end-point of e is matched in M. Moreover, two edges *intersect* if they share at least one end-point. We define the following edge sets for $i \geq 1$.

- ALG_i is the set of edges the algorithm includes in the matching when list L_i is probed.
- OPT_i is the set of edges in OPT that intersect with edges in ALG_i, but do not intersect with edges in ALG_j, for all $j < i$.
- $\mathsf{OPT}_i^H := \mathsf{OPT}_i \cap H_i$, each of which has weight in $[a_i, b_i]$.

The matching resulting from the probing list L returned by the algorithm is $\mathsf{ALG} := \cup_i \mathsf{ALG}_i$. Since ALG is a maximal matching in G, it follows that every edge in OPT appears in exactly one OPT_i.

Suppose V_i is the set of nodes matched in ALG_i. Let $C(\alpha_{V_i}) := \sum_{v \in V_i} \alpha_v$, where α_{V_i} is the vector α restricted to coordinates corresponding to V_i.

We defer the proof of the following lemma to our full version.

▶ **Lemma 20** (Local Performance Ratio). *Suppose the weights of H_i are in $[a_i, b_i]$, where $\eta := \mathsf{dist}(a_i, b_i)$; moreover, let $\lambda := \mathsf{dist}(b_{i+1}, b_i)$. Then, $\mathbf{E}[w(\mathsf{ALG}_i)] \geq (1 - \eta) \cdot (\frac{1}{2} + \frac{\xi_1 \lambda}{1+2\xi_1}) \cdot \mathbf{E}[C(\alpha_{V_i})]$.*

Finally, we are ready to prove the performance ratio of the algorithm.

Proof of Theorem 3: From Lemma 19, it follows that the parameters in Lemma 20 satisfy $\lambda \geq \epsilon^{r-1}$ and $\eta \leq \epsilon^r$, where $r = O(k^2)$. Observing that $\epsilon = \frac{\xi_1}{2} \leq \frac{1}{4}$, it follows that the local performance ratio is

$$\begin{aligned}\frac{\mathbf{E}[w(\mathsf{ALG}_i)]}{\mathbf{E}[C(\alpha_{V_i})]} \geq& (1-\eta)\left(\frac{1}{2} + \frac{\xi_1 \lambda}{1+2\xi_1}\right) \geq (1-\epsilon^r)\left(\frac{1}{2} + \frac{\xi_1 \epsilon^{r-1}}{1+2\xi_1}\right) = (1-\epsilon^r)\left(\frac{1}{2} + \frac{2\epsilon^r}{1+2\xi_1}\right) \\ =& \frac{1}{2} + \frac{\epsilon^r}{2+4\xi_1} \cdot (3 - 4(\epsilon + \epsilon^r)) \geq \frac{1}{2} + \frac{1}{2+4\xi_1}(\frac{\xi_1}{2})^r,\end{aligned}$$

where the last inequality follows because $r \geq 1$ and $\epsilon \leq \frac{1}{4}$.

Hence, we have $\mathbf{E}[\mathsf{ALG}] = \sum_i \mathbf{E}[\mathsf{ALG}_i] \geq (\frac{1}{2} + \frac{1}{2+4\xi_1}(\frac{\xi_1}{2})^r) \sum_i \mathbf{E}[C(\alpha_{V_i})]$. Finally, observe that $\sum_i \mathbf{E}[C(\alpha_{V_i})] = E[C(\alpha)] \geq w(\mathsf{OPT})$, as α is a matching coverage for OPT. Therefore, we conclude that the performance ratio for the whole algorithm is at least $\frac{1}{2} + \xi_k$, where $\xi_k = \frac{1}{2+4\xi_1}(\frac{\xi_1}{2})^r = \Omega(\xi_1)^{O(k^2)}$, as required. ∎

References

1 Gagan Aggarwal, Gagan Goel, Chinmay Karande, and Aranyak Mehta. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *SODA*, pages 1253–1264, 2011.

2 Jonathan Aronson, Martin Dyer, Alan Frieze, and Stephen Suen. Randomized greedy matching. ii. *Random Struct. Algorithms*, 6(1):55–73, January 1995.

3 Benjamin Birnbaum and Claire Mathieu. On-line bipartite matching made simple. *SIGACT News*, 39(1):80–87, March 2008.

4 Niv Buchbinder, Kamal Jain, and Joseph Seffi Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *ESA*, pages 253–264, 2007.

5 T.-H. Hubert Chan, Fei Chen, Xiaowei Wu, and Zhichao Zhao. Ranking on arbitrary graphs: Rematch via continuous lp with monotone and boundary condition constraints. In *SODA*, pages 1112–1122, 2014.

6 Nikhil R. Devanur, Kamal Jain, and Robert D. Kleinberg. Randomized primal-dual analysis of ranking for online bipartite matching. In *SODA*, pages 101–107, 2013.

7 Martin E. Dyer and Alan M. Frieze. Randomized greedy matching. *Random Struct. Algorithms*, 2(1):29–46, 1991.

8 Jon Feldman, Nitish Korula, Vahab S. Mirrokni, S. Muthukrishnan, and Martin Pál. Online ad assignment with free disposal. In *WINE*, pages 374–385, 2009.

9 Gagan Goel and Aranyak Mehta. Online budgeted matching in random input models with applications to adwords. In *SODA*, pages 982–991, 2008.

10 Gagan Goel and Pushkar Tripathi. Matching with our eyes closed. In *FOCS*, pages 718–727, 2012.

11 Chinmay Karande, Aranyak Mehta, and Pushkar Tripathi. Online bipartite matching with unknown distributions. In *STOC*, pages 587–596, 2011.

12 Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for on-line bipartite matching. In *STOC*, pages 352–358, 1990.

13 Mohammad Mahdian and Qiqi Yan. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing LPs. In *STOC*, pages 597–606, 2011.

14 Silvio Micali and Vijay V. Vazirani. An $O(\sqrt{V}E)$ algorithm for finding maximum matching in general graphs. In *FOCS'80*, pages 17–27. IEEE Computer Society, 1980.

15 Alvin E Roth, Tayfun Sönmez, and M Utku Ünver. Pairwise kidney exchange. *Journal of Economic theory*, 125(2):151–188, 2005.