SIAM J. COMPUT.
Vol. 47, No. 4, pp. 1529–1546

RANKING ON ARBITRARY GRAPHS: REMATCH VIA CONTINUOUS LINEAR PROGRAMMING*

T-H. HUBERT CHAN†, FEI CHEN†, XIAOWEI WU†, AND ZHICHAO ZHAO†

Abstract. Motivated by online advertisement and exchange settings, greedy randomized algorithms for the maximum matching problem have been studied, in which the algorithm makes (random) decisions that are essentially oblivious to the input graph. Any greedy algorithm can achieve a performance ratio of 0.5, which is the expected number of matched nodes to the number of nodes in a maximum matching. Since Aronson, Dyer, Frieze, and Suen [*Random Structures Algorithm*, 6 (1991), pp. 29–46] proved that the modified randomized greedy algorithm achieves a performance ratio of $0.5 + \epsilon$ (where $\epsilon = \frac{1}{400000}$) on arbitrary graphs in the midnineties, no further attempts in the literature have been made to improve this theoretical ratio for arbitrary graphs until two papers were published in *FOCS* 2012 [G. Goel and P. Tripathi, IEEE Computer Society, Los Alamitos, CA, 2012, pp. 718–727; M. Poloczek and M. Szegedy, IEEE Computer Society, Los Alamitos, CA, 2012, pp. 708–717]. In this paper, we revisit the ranking algorithm using the linear programming framework. Special care is given to analyze the structural properties of the ranking algorithm in order to derive the linear programming constraints, of which one known as the *boundary* constraint requires totally new analysis and is crucial to the success of our linear program (LP). We use continuous linear programming relaxation to analyze the limiting behavior as the finite LP grows. Of particular interest are new duality and complementary slackness characterizations that can handle the monotone and the boundary constraints in continuous linear programming. Improving previous work, this paper achieves a theoretical performance ratio of $\frac{2(5-\sqrt{7})}{9} \approx 0.523$ on arbitrary graphs.

Key words. oblivious matching problem, continuous linear programming, ranking algorithm

AMS subject classification. 68R01

DOI. 10.1137/140984051

1. Introduction.

Maximum matching [16] in undirected graphs is a classical problem in computer science. However, as motivated by online advertising [8, 2] and exchange settings [18], information about the graphs can be incomplete or unknown.

As mentioned in [9], an example of this setting is the kidney exchange problem [18], in which an incompatible patient-donor pair is represented by a node. A probe between two nodes corresponds to testing whether swapping between the two pairs can allow both patients to have compatible donors, in which case an edge exists between the two nodes. Due to cost and ethical concerns, it is preferable that an exchange is performed whenever two such compatible patient-donor pairs are discovered. Hence, the procedure is greedy in nature.

Other online or greedy versions of matching problems [5, 17, 9] can also be formulated by the following problem, in which the algorithm is initially oblivious to the input graph.

Oblivious Matching Problem. An *adversary* commits to a graph $G(V, E)$ and reveals the nodes V (where $n = |V|$) to the (possibly randomized) *algorithm*, while keeping the edges E 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

*Received by the editors August 27, 2014; accepted for publication (in revised form) June 11, 2018; published electronically August 7, 2018. The conference version of this paper appeared in SODA, SIAM, Philadelphia, 2014, pp. 1112–1122.
http://www.siam.org/journals/sicomp/47-4/98405.html
Funding: This research was supported in part by a grant from Hong Kong RGC under the contract 17200214.
†Department of Computer Science, the University of Hong Kong, Hong Kong (hubert@cs.hku.hk, fchen@cs.hku.hk, xwwu@cs.hku.hk, zczhao@cs.hku.hk).

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) number of nodes matched by the algorithm to the number of nodes in a maximum matching in G.

Observe that any ordering of the pairs $\binom{V}{2}$ will result in a maximal matching in $G(V, E)$, giving a trivial performance ratio of at least 0.5. However, for any deterministic algorithm, the adversary can choose a graph such that the ratio 0.5 is attained. The interesting question is: how much better can randomized algorithms perform on arbitrary graphs? For bipartite graphs, there are theoretical analyses of randomized algorithms [12, 15] achieving ratios better than 0.5.

The Ranking algorithm (an early version appeared in [13]) is simple to describe: a permutation σ on V is selected uniformly at random and, naturally, induces a lexicographical order on the unordered pairs in $\binom{V}{2}$ used for probing. Although, by experiments, the Ranking algorithm and other randomized algorithms seem to achieve performance ratios much larger than 0.5, until very recently, the best theoretical performance ratio $0.5 + \epsilon$ (where $\epsilon = \frac{1}{400000}$) for the problem on arbitrary graphs was proved in the midnineties by Aronson et al. [5], who analyzed the Modified Randomized Greedy algorithm (MRG). In the MRG algorithm, in addition to choosing the random permutation that decides the ordering of nodes to be probed, an extra independent random permutation is chosen for every node in the graph.

After more than a decade of research, two papers were published in FOCS 2012 that attempted to give theoretical ratios significantly better than the $0.5 + \epsilon$ bound. Poloczek and Szegedy [17] also analyzed the MRG algorithm to give a ratio $0.5 + \frac{1}{256} \approx 0.5039$. Goel and Tripathi [9] analyzed the Ranking algorithm and claimed the ratio 0.56 can be achieved, but they later realized that there was a crucial bug in their proof, and announced the withdrawal of the paper on arXiv [10]. Both papers used a common framework which has been successful for analyzing bipartite graphs: (i) utilize the structural properties of the matching problem to form a minimization linear program that gives a lower bound on the performance ratio; (ii) analyze the linear probram (LP) theoretically and/or experimentally to give a lower bound.

In this paper, we revisit the Ranking algorithm using the same framework: (i) we use novel techniques to carefully analyze the structural properties of Ranking for producing new linear programming constraints; (ii) moreover, we develop new primal-dual techniques for continuous LPs to analyze the limiting behavior as the finite LP grows. Of particular interest are new duality and complementary slackness results that can handle monotone constraints and boundary conditions in continuous linear programming. Compared to previous work, our paper achieves the best theoretical performance ratio of $\frac{2(5-\sqrt{7})}{9} \approx 0.523$ on arbitrary graphs. As a side note, our experiments suggest that Ranking cannot perform better than 0.724 in general.

1.1. Our contribution and techniques. We show the following main result in this paper.

THEOREM 1. *For the Oblivious Matching Problem on arbitrary graphs, the Ranking algorithm achieves a performance ratio of at least* $\frac{2(5-\sqrt{7})}{9} \approx 0.523$.

Following previous work on the analysis of Ranking [13], we consider a set $\mathcal{U}$ of *instances,* each of which has the form (σ, u), where σ is a permutation on V and u is a node in V. An instance (σ, u) is *good* if the node u is matched when Ranking is run with σ, and *bad* otherwise; an event is a subset of instances. As argued in [17, 9], one can assume that G contains a perfect matching when analyzing the ratio of Ranking. Hence, the performance ratio of Ranking is the fraction of good instances.

(1) Relating bad and good events to form linear programming constraints. A simple combinatorial argument [13] is often used to relate bad and good instances. For example, if each bad instance relates to at least two good instances, and each good instance is related to at most one bad instance, then the fraction of good instances would be at least $\frac{2}{3}$. By considering the structural properties of Ranking, one can define various relations between different bad and good events, and hence can generate various constraints in an LP, whose optimal value gives a lower bound on the performance ratio. Despite the simplicity of this combinatorial argument, the analysis of these relations can be elusive for arbitrary graphs.

We define and analyze our relations carefully to derive three type of constraints: *monotone* constraints, *evolving* constraints, and a *boundary* constraint, the last of which involves a novel construction of a sophisticated relation, and is crucial to the success of our LP_n.

(2) Developing new primal-dual techniques for continuous linear programming. As in previous works, the optimal value of LP_n decreases as n increases. Hence, to obtain a theoretical proof, one needs to analyze the asymptotic behavior of LP_n. It could be tedious to find the optimal solution of LP_n and investigate its limiting behavior. One could also use experiments (for example, using strongly factor-revealing LPs [15]) to give a proof. We instead observe that LP_n has a continuous LP_∞ relaxation (in which normal variables become a function variable). However, the monotone constraints in LP_n require that the function in LP_∞ be monotonically decreasing. Moreover, the boundary constraint has its counterpart in LP_∞. Contrary to previous work [15], the method of continuous relaxation avoids the computation of an additional strongly factor-revealing LP to bound the factor-revealing LP. To the best of our knowledge, such continuous LPs have not been analyzed in the literature.

We describe our formal notation in section 2. In section 3, we relate bad and good events in order to form LP_n. In section 4, we prove a lower bound on the performance ratio; in particular, we develop new primal-dual and complementary slackness characterizations for a general class of continuous linear programming, and solve the continuous LP_∞ relaxation (and its dual). In section 5, we describe a hard instance and our experiments suggest that Ranking performs no better than 0.724 in general.

1.2. Related work. We describe and compare the most relevant related work. Please refer to the references in [17, 9] for a more comprehensive background on the problem. We describe Oblivious Matching Problem general enough so that we can compare different works that are studied under different names and settings. Dyer and Frieze [6] showed that picking a permutation of unordered pairs uniformly at random cannot produce a constant ratio that is strictly greater than 0.5. On the other hand, this framework also includes the MRG algorithm, which was analyzed by Aronson et al. [5] to prove the first nontrivial constant performance ratio crossing the 0.5 barrier. One can also consider *adaptive* algorithms in which the algorithm is allowed to change the order in the remaining list after seeing the probing results; although hardness results have been proved for adaptive algorithms [9], no algorithm in the literature seems to utilize this feature yet.

On bipartite graphs. Running Ranking on bipartite graphs for the Oblivious Matching Problem is equivalent to running ranking [13] for the Online Bipartite Matching problem with *random arrival order* [12]. From Karande, Mehta, and Tripathi [12], one can conclude that Ranking achieves a ratio of 0.653 on bipartite graphs. Moreover, they constructed a hard instance in which Ranking performs no better than 0.727; we modify their hard instance and our experiments suggests that Ranking performs no better than 0.724.

On a high level, most works on analyzing Ranking or similar randomized algorithms on matching are based on variations of the framework by Karp, Vazirani, and Vazirani [13]. The basic idea is to relate different bad and good events to form constraints in an LP, whose asymptotic behavior is analyzed when n is large. For Online Bipartite Matching, Karp, Vazirani, and Vazirani [13] showed that ranking achieves a performance ratio of $1 - \frac{1}{e}$; similarly, Aggarwal et al. [2] also showed that a modified version of Ranking achieves the same ratio for the node-weighted version of the problem.

Sometimes very sophisticated mappings are used to relate different events, and produce linear programs whose asymptotic behavior is difficult to analyze. Mahdian and Yan [15] developed the technique of strongly factor-revealing LP. The idea is to consider another family of linear programs whose optimal values are all below the asymptotic value of the original LP. Hence, the optimal value of any LP (usually a large enough instance) in the new family can be a lower bound on the performance ratio. The results of [15] imply that for the Oblivious Matching Problem on bipartite graphs, Ranking achieves a performance ratio of 0.696.

Recent attempts. No attempts have been made in the literature to theoretically improve the $0.5 + \epsilon$ ratio for arbitrary graphs until two recent papers appeared in FOCS 2012. Poloczek and Szegedy [17] used a technique known as *contrast analysis* to analyze the MRG algorithm and gave a ratio of $\frac{1}{2} + \frac{1}{256} \approx 0.5039$. However, we discover some gaps in their proof; from personal communication with the authors, we are told that they are currently bridging those gaps at the time of writing.

Goel and Tripathi [9] showed a hardness result of 0.7916 for any algorithm and 0.75 for adaptive vertex-iterative algorithms. They also analyzed the Ranking algorithm for a better performance ratio, but later withdrew the paper [10] when they discovered a bug in their proof.

Continuous LPs. Duality and complementary slackness properties of continuous LPs were investigated by Tyndall [19] and Levinson [14]. Anand, Garg, and Kumar [4] used continuous LP relaxation to analyze online scheduling.

1.3. Subsequent work. Since the conference version of this paper was published, there have been subsequent works in this line of research.

In a follow-up paper [1], the node-weighted version of the Oblivious Matching Problem was considered and the first performance ratio strictly above 0.5 has been achieved. Interestingly, it was shown in [1] that applying the techniques developed for the node-weighted case to the unweighted case, the performance analysis of the Ranking algorithm can be slightly improved to 0.5268.

Wang and Wong [20] considered the online fractional matching problem in general graphs (with arbitrary arrival order of nodes), and achieved a 0.526 competitive algorithm.

The online bipartite matching problem with random arrival of edges was considered in [11]. Indeed, they studied a more general online matroid intersection problem, and achieved the first competitive ratio strictly above 0.5 by a randomized algorithm.

Similarly to the continuous relaxation techniques used in this paper, Alaei et al. [3] considered a continuous optimization problem to achieve the worst case approximation factor of anonymous pricing in certain auction settings.

We discuss further future research directions in section 6.

2. Preliminaries. Let $[n] := \{1, 2, \ldots, n\}$, $[a..b] := \{a, a+1, \ldots, b\}$ for $1 \le a \le b$, and Ω be the set of all permutations of the nodes in V, where each permutation is a bijection $\sigma : V \to [n]$. The *rank* of node u in σ is $\sigma(u)$, where smaller rank means higher priority.

The Ranking *algorithm.* For the Oblivious Matching Problem, the algorithm selects a permutation $\sigma \in \Omega$ uniformly at random, and returns a list L of unordered pairs according to the lexicographical order induced by σ. Specifically, given two pairs e_1 and e_2 (where for each i, $e_i = \{u_i, v_i\}$ and $\sigma(u_i) < \sigma(v_i)$), the pair e_1 has higher priority than e_2 if (i) $\sigma(u_1) < \sigma(u_2)$, or (ii) $u_1 = u_2$ and $\sigma(v_1) < \sigma(v_2)$. Each pair of nodes in $G(V, E)$ is probed according to the order given by L; initially, all nodes are *unmatched*. In the round when the pair $e = \{u, v\}$ is probed, if both nodes are currently unmatched and the edge e is in E, then each of u and v is matched, and they are each other's *partner* in σ; moreover, if $\sigma(u) < \sigma(v)$ in this case, we say that u *chooses* v. Otherwise, if at least one of u and v is already matched or there is no edge between them in G, we skip to the next pair in L until all pairs in L are probed.

After running Ranking with σ (or, in general, probing with list L), we denote the resulting matching by $M(\sigma)$ (or $M(L)$), and we say that a node is matched in σ (or L) if it is matched in $M(\sigma)$ (or $M(L)$). Given a probing list L, let L_u denote the probing list obtained by removing all occurrences of u in L such that u always remains unmatched. The following lemma is similar to [9, Claim 1] and [17, Lemma 3]. We present here a formal proof for completeness.

LEMMA 2 (removing one node). *The symmetric difference $M(L) \oplus M(L_u)$, i.e., the set of edges that appear in exactly one of $M(L)$ and $M(L_u)$, is an alternating path consisting of edges alternating between $M(L)$ and $M(L_u)$. This path contains at least one edge* iff *u is matched in L.*

Proof. Observe that probing G with L_u is equivalent to probing G_u with L, where G_u is exactly the same as G except that the node u is labeled *occupied* and will not be matched in any case. Hence, we will use the same L to probe G and G_u, and compare what happens in each round to the corresponding matchings $M = M(L)$ and $M_u = M(L_u)$. For the sake of this proof, "occupied" and "matched" nodes are considered to be "unavailable" and have the same *availability status*, while an "unmatched" node is considered to be "available."

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 $M(L) \oplus M(L_u)$ is an alternating path connecting u to the crucial node; initially, this path is degenerate.

Consider the inductive step. Observe that the crucial node and $M(L) \oplus M(L_u)$ 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 w with another currently unmatched node v in G. Observe that in this case, v is also unmatched in G_u, as the induction hypothesis states that every other node apart from the crucial node has the same availability in both graphs. Hence, this edge is added to exactly one of M and M_u. Therefore, w is unavailable in both graphs (so no longer crucial), and v becomes the new crucial node; moreover, the edge $\{w, v\}$ is added to $M(L) \oplus M(L_u)$, which now is a path connecting u to v. This completes the inductive step.

Observe that u is matched in M in the end, *iff* in some round an edge involving u must be added to M but not to M_u, which is equivalent to the case when $M \oplus M_u$ contains at least one edge. ∎

The performance ratio r of Ranking on G is the expected number of nodes matched by the algorithm to the number of nodes in a maximum matching in G, where the randomness comes from the random permutation in Ω. We consider the set $\mathcal{U} := \Omega \times V$ of instances; an *event* is a subset of instances. An instance $(\sigma, u) \in \mathcal{U}$ is good if u is matched in σ, and bad otherwise.

Perfect matching assumption. According to Corollary 2 of [17] (and also implied by our Lemma 2), without loss of generality, we can assume that the graph $G(V, E)$ has a perfect matching $M^* \subseteq E$ that matches all nodes in V. For a node u, we denote by u^* the partner of u in M^* and we call u^* the *perfect partner* of u. From now on, we consider Ranking on such a graph G without mentioning it explicitly again. Observe that for all $\sigma \in \Omega$, $(\sigma, \sigma^{-1}(1))$ is always good; moreover, the performance ratio is the fraction of good instances.

DEFINITION 3 (σ_u, σ_u^i). *For a permutation σ, let σ_u be the permutation obtained by removing u from σ while keeping the relative order of other nodes unchanged; running Ranking with σ_u means running σ while keeping u always unavailable (or simply deleting u in G). Let σ_u^i be the permutation obtained by inserting u into σ_u at rank i and keeping the relative order of other nodes unchanged.*

FACT 1 (Ranking is greedy). *Suppose Ranking is run with permutation σ. If u is unmatched in σ, then each neighbor w of u (in G) is matched to some node v in σ with $\sigma(v) < \sigma(u)$.*

Similarly to [17, Lemma 3], the following fact is an easy corollary of Lemma 2, by observing that if (σ, u) is bad, then $M(\sigma) = M(\sigma_u)$.

FACT 2 (symmetric difference). *Suppose (σ, u) is bad, and (σ_u^i, u) is good for some i. Then, the symmetric difference $M(\sigma) \oplus M(\sigma_u^i)$ is an alternating path P with at least one edge, where except for the endpoints of P (of which u is one), every other node in G is either matched in both σ and σ_u^i, or unmatched in both.*

We adopt the following definitions as used in [2] for the Online Bipartite Matching problem.

DEFINITION 4 (Q_t, R_t, and S_t). *For each $t \in [n]$, let Q_t be the good event that the node at rank t is matched, where $Q_t := \{(\sigma, u) : \sigma \in \Omega, u = \sigma^{-1}(t)$ is matched in $\sigma\}$; similarly, let R_t be the bad event that the node at rank t is unmatched, where $R_t := \{(\sigma, u) : \sigma \in \Omega, u = \sigma^{-1}(t)$ is unmatched in $\sigma\}$.*

Moreover, we define the marginally bad event *S_t at rank $t \in [2..n]$ by $S_t := \{(\sigma, u) \in R_t : (\sigma_u^{t-1}, u) \notin R_{t-1}\}$; observe that $S_1 = R_1 = \emptyset$.*

Given any $(\sigma, u) \in \mathcal{U}$, the marginal position *of u with respect to σ is the (unique) rank t such that $(\sigma_u^t, u) \in S_t$, and is* null *if no such t exists.*

Note that for each $t \in [n]$, Q_t and R_t are disjoint and $|Q_t \cup R_t| = n!$.

DEFINITION 5 (x_t, α_t). *For each $t \in [n]$, let $x_t = \frac{|Q_t|}{n!}$ be the probability that a node at rank t is matched, over the random choice of permutation σ. Similarly, we let $\alpha_t = \frac{|S_t|}{n!}$; observe that $1 - x_t = \frac{|R_t|}{n!}$.*

Note that the performance ratio is $\frac{1}{n}\sum_{t=1}^{n} x_t$, which will be the objective function of our minimization LP. Observe that all x_t's and α_t's are between 0 and 1, and $x_1 = 1$ and $\alpha_1 = 0$. We derive constraints for the variables in the next section.

3. Relating bad and good events to form linear programming constraints. In this section we define some relations between bad and good events to form linear programming constraints. The high level idea is as follows. Suppose f is a relation between A and B, where $f(a)$ is the set of elements in B related to $a \in A$, and $f^{-1}(b)$ is the set of elements in A related to $b \in B$. The *injectivity* of f is the minimum integer q such that for all $b \in B$, $|f^{-1}(b)| \leq q$. If f has injectivity q, we have the inequality $\sum_{a \in A} |f(a)| \leq q|B|$, which follows from counting the number of edges in the bipartite graph induced by f on A and B. In our constructions, usually calculating $|f(a)|$ is straightforward, but sometimes special attention is required to bound the injectivity.

3.1. Monotone constraints: $\boldsymbol{x_{t-1} \geq x_t,\ t \in [2..n]}$. These constraints follow immediately from Lemma 6, which is similar to [2, Claim 2] (for bipartite graphs), as the α_t's are nonnegative. The constraints say that nodes with smaller ranks are more likely to be matched (over the choice of random permutations).

LEMMA 6 (bad–to–marginally-bad). *For all $t \in [n]$, we have $1 - x_t = \sum_{i=1}^{t} \alpha_i$; this implies that for $t \in [2..n]$, $x_{t-1} - x_t = \alpha_t$.*

Proof. Fix $t \in [n]$. From the definitions of x_t and α_t, it suffices to provide a bijection f from R_t to $\cup_{i=1}^{t} S_i$. Suppose $(\sigma, u) \in R_t$. This means (σ, u) is bad, and hence u has a marginal position $t_u \leq t$ with respect to σ. We define $f(\sigma, u) := (\sigma_u^{t_u}, u) \in \cup_{i=1}^{t} S_i$.

Surjective: for each $(\rho, v) \in \cup_{i=1}^{t} S_i$, the marginal position of v with respect to ρ is some $i \leq t$; hence, it follows that $(\rho_v^t, v) \in R_t$ is bad, and we have $f(\rho_v^t, v) = (\rho, v)$.

Injective: if we have $f(\sigma, u) = (\rho, v)$, it must be the case that $u = v$, $\sigma(u) = t$, and $\rho = \sigma_u^i$ for some i; this implies that σ must be ρ_v^t.

Hence, $|R_t| = |\cup_{i=1}^{t} S_i| = \sum_{i=1}^{t} |S_i|$, which is equivalent to $1 - x_t = \sum_{i=1}^{t} \alpha_i$ if we divide the equation by $n!$ on both sides. ∎

3.2. Evolving constraints: $\boldsymbol{\left(1 - \frac{t-1}{n}\right) x_t + \frac{2}{n} \sum_{i=1}^{t-1} x_i \geq 1,\ t \in [2..n]}$. The monotone constraints require that the x_t's do not increase. We next derive the evolving constraints that prevent the x_t's from dropping too fast. Fix $t \in [2..n]$. We shall define a relation f between $\cup_{i=1}^{t} S_i$ and $\cup_{i=1}^{t-1} Q_i$ such that f has injectivity 1, and for $(\sigma, u) \in S_i$, $|f(\sigma, u)| = n - i + 1$. This implies Lemma 7; from Lemma 6, we can express $\alpha_i = x_{i-1} - x_i$ (recall $\alpha_1 = 0$), and rearrange the terms to obtain the required constraint.

Proof intuition. Given a bad instance $(\sigma, u) \in R_t$, the perfect partner u^* must be matched to some node v such that $\sigma(v) < t$. Hence, demoting the rank of u further to $[t..n]$ will produce the same matching, and u^* will still be matched to the same v. If the graph is bipartite, then after promoting the rank of u, it can be shown that u^* will be matched to some node (perhaps a different one) with rank in $[1..t-1]$. Therefore, if the graph is bipartite, we could have produced the constraint $1 - x_t \leq \frac{1}{n} \sum_{i=1}^{t-1} x_i$, which would lead to a performance ratio of $1 - \frac{1}{e}$.

However, for general graphs, promoting the rank of u might cause its perfect partner u^* to become unmatched. Hence, we can only derive a weaker evolving constraint that is obtained by considering only demotion of an unmatched node.

LEMMA 7 (1-to-$(n-i+1)$ mapping). *For all $t \in [2..n]$, we have $\sum_{i=1}^{t} (n-i+1)\alpha_i \leq \sum_{i=1}^{t-1} x_i$.*

Proof. Let $(\sigma, u) \in \cup_{i=1}^{t} S_i$ be a marginally bad instance. Then, there exists a unique $i \in [2..t]$ such that $(\sigma, u) \in S_i$. If we move u to any position $j \in [i..n]$, (σ_u^j, u) is still bad, because i is the marginal position of u with respect to σ. Moreover, observe that $M(\sigma_u) = M(\sigma) = M(\sigma_u^j)$ for all $j \in [i..n]$.

Hence, it follows that for all $j \in [i..n]$, node u's perfect partner u^* is matched in σ_u^j to the same node v such that $\sigma(v) = \sigma_u^j(v) \le i-1 \le t-1$, where the first inequality follows from Fact 1. In this case, we define $f(\sigma, u) := \{(\sigma_u^j, v) : j \in [i..n]\} \subset \cup_{i=1}^{t-1} Q_i$, and it is immediate that $|f(\sigma, u)| = n - i + 1$.

Injectivity. Suppose $(\rho, v) \in \cup_{i=1}^{t-1} Q_i$ is related to some $(\sigma, u) \in \cup_{i=1}^{t} S_i$. It follows that v must be matched to u^* in ρ; hence, u is uniquely determined by (ρ, v). Moreover, (ρ, u) must be bad, and suppose the marginal position of u with respect to ρ is i, which is also uniquely determined. Then, it follows that σ must be ρ_u^i. Hence, (ρ, v) can be related to at most one element in $\cup_{i=1}^{t} S_i$.

Observing that $S_1 = \emptyset$, the result follows from

$$\sum_{i=1}^{t} (n - i + 1)|S_i| = \sum_{a \in \cup_{i=1}^{t} S_i} |f(a)| \le |\cup_{i=1}^{t-1} Q_i| = \sum_{i=1}^{t-1} |Q_i|,$$

since $|S_i| = n!\alpha_i$ and $|Q_i| = n!x_i$. ∎

3.3. Boundary constraint: $x_n + \frac{3}{2n}\sum_{i=1}^{n} x_i \ge 1$. One can check (for instance, by experiments) that the monotone and the evolving constraints alone cannot give a ratio better than 0.5. Indeed, by setting $x_t = \frac{n-t+1}{n}$ for all $t \in [n]$, all previous constraints are satisfied while the performance ratio is $\frac{1}{2} + \frac{1}{2n}$.

However, it can be observed (for instance, by experiments) that given the monotone and evolving constraints, we have $\sum_{i=1}^{t} x_i \ge \sum_{i=1}^{t} \frac{n-t+1}{n}$. Hence, in some sense, the above solution is the "worst" possible. Hence, as long as we can show that $x_n \ge \delta$ for some small constant $\delta > 0$, we can achieve a performance ratio strictly larger than 0.5.

The boundary constraint provides such a lower bound in terms of the performance ratio, i.e., x_n is lower bounded by some constant whenever the performance ratio is strictly smaller than $\frac{2}{3}$. The boundary constraint is crucial to the success of our LP, and hence we analyze our construction carefully.

The high level idea is that we define a relation f between R_n and $Q := \cup_{i=1}^{n} Q_i$. As we shall see, it will be straightforward to show that $|f(a)| = 2n$ for each $a \in R_n$, but it will require some work to show that the injectivity is at most 3. Once we have established these results, the boundary constraint follows immediately from $\sum_{a \in R_n} |f(a)| \le 3|Q|$, because $\frac{|R_n|}{n!} = 1 - x_n$ and $\frac{|Q_i|}{n!} = x_i$.

Defining relation f between R_n and Q. Consider a bad instance $(\sigma, u) \in R_n$. We define $f(\sigma, u)$ such that for each $i \in [n]$, (σ, u) *produces* exactly two good instances of the form $(\sigma_u^i, *)$.

For each $i \in [n]$, we consider σ_u^i:

1. if u is unmatched in σ_u^i: *(u and u^* cannot be both unmatched)*:
   R(1): produce (σ_u^i, u^*) and include it in $f(\sigma, u)$;
   R(2): let v be the partner of u^* in σ_u^i; produce (σ_u^i, v) and include it in $f(\sigma, u)$.
2. if u is matched in σ_u^i:
   R(3): produce (σ_u^i, u) and include it in $f(\sigma, u)$;
   (a) if u^* is matched to u in σ_u^i:
   R(4): produce (σ_u^i, u^*) and include it in $f(\sigma, u)$;
   (b) if u^* is matched to $v \ne u$ in σ_u^i:
   R(5): produce (σ_u^i, v) and include it in $f(\sigma, u)$;
   (c) if u^* is unmatched in σ_u^i: *(all neighbors of u^* in G must be matched)*:

R(6): let v_o be the partner of u^* in σ, produce (σ_u^i, v_o) and include it in $f(\sigma, u)$.

Observe that for $k \in [6]$, applying each rule R(k) produces exactly one good instance. Moreover, for each $i \in [n]$, when we consider σ_u^i, exactly 2 rules will be applied: if u is unmatched in σ_u^i, then R(1) and R(2) will be applied; if u is matched in σ_u^i, then R(3) and one of R(4), R(5), R(6)} will be applied.

Observation 1. For each $(\sigma, u) \in R_n$, we have $|f(\sigma, u)| = 2n$.

Observation 2. If $(\rho, x) \in f(\sigma, u)$, then $\sigma = \rho_u^n$ and exactly one rule can be applied to (σ, u) to produce (ρ, x).

Bounding injectivity. We first show that different bad instances in R_n cannot produce the same good instance using the same rule.

LEMMA 8 (rule injectivity). *For each $k \in [6]$, any $(\rho, x) \in Q$ can be produced by at most one $(\sigma, u) \in R_n$ using* R(k).

Proof. Suppose $(\rho, x) \in Q$ is produced using a particular rule R(k) by some $(\sigma, u) \in R_n$. We wish to show that in each case $k \in [6]$, we can recover u uniquely, in which case σ must be ρ_u^n.

The first 5 cases are simple. Let y be the partner of x in ρ. If $k = 1$ or $k = 4$, we know that $x = u^*$ and hence we can recover $u = x^*$; if $k = 2$ or $k = 5$, we know that $y = u^*$ and hence we can recover $u = y^*$; if $k = 3$, we know that $u = x$.

For the case when $k = 6$, we need to do a more careful analysis. Suppose R(6) is applied to $(\sigma, u) \in R_n$ to produce (ρ, x). Then, we can conclude the following: (i) in $\sigma = \rho_u^n$, u is unmatched, and u^* is matched to x; (ii) in ρ, u is matched, u^* is unmatched, and x is matched.

For contradiction's sake, assume that u is not unique and there are two $u_1 \neq u_2$ that satisfy the above properties. It follows that $u_1^* \neq u_2^*$ and according to property (ii), in ρ, both u_1 and u_2 are matched, and both u_1^* and u_2^* are unmatched; hence, all 4 nodes are distinct. Without loss of generality, we assume that $\rho(u_1^*) < \rho(u_2^*)$. Let $\sigma_2 := \rho_{u_2}^n$, and observe that $\sigma_2(u_1^*) < \sigma_2(u_2^*)$.

Now, suppose we start with σ_2, and consider what happens when u_2 is promoted in σ_2 resulting in ρ. Observe that u_2 changes from unmatched in σ_2 to matched in ρ, and by property (i), u_2^* changes from matched in σ_2 to unmatched in ρ. From Fact 2, every other node must remain matched or unmatched in both σ_2 and ρ; in particular, u_1^* is unmatched in σ_2. However, x is a neighbor of both u_1^* and u_2^* (in G), and $\sigma_2(u_1^*) < \sigma_2(u_2^*)$, but x is matched to u_2^* in σ_2; this contradicts Fact 1. $\square$

Lemma 8 immediately implies that the injectivity of f is at most 6. However, to show a better bound of 3, we need to show that some of the rules cannot be simultaneously applied to produce the same good instance (ρ, x). We consider two cases for the remaining analysis.

Case (1): *x is matched to x^* in ρ.*

LEMMA 9. *For $(\rho, x) \in Q$, if x is matched to x^* in ρ, then we have $|f^{-1}(\rho, x)| \leq 3$.*

Proof. If (ρ, x) is produced using R(1), then x^* must be unmatched in ρ; if (ρ, x) is produced by (σ, u) using R(2), then x must be matched to $u^* \, (\neq x^*)$ in ρ since $x \neq u$; similarly, if (ρ, x) is produced by (σ, u) using R(5), then $x \, (\neq u)$ must be matched to $u^* \, (\neq x^*)$ in ρ.

Hence, (ρ, x) cannot be produced by R(1), R(2), or R(5), and at most three remaining rules can produce it. It follows from Lemma 8 that $|f^{-1}(\rho, x)| \leq 3$. $\square$

Case (2): *x is not matched to x^* in ρ.*

Observation 3 (unused rule). For $(\rho, x) \in Q$, if x is not matched to x^* in ρ, then (ρ, x) cannot be produced by applying R(4).

Out of the remaining 5 rules, we show that (ρ, x) can be produced from at most one of $\{\mathrm{R}(2), \mathrm{R}(5)\}$, and at most two of $\{\mathrm{R}(1), \mathrm{R}(3), \mathrm{R}(6)\}$. After we show these two lemmas, we can immediately conclude from Lemma 8 that $|f^{-1}(\rho, x)| \leq 3$ and complete the case analysis.

LEMMA 10 (one in R(2), R(5)). *Each $(\rho, x) \in Q$ cannot be produced from both* R(2) *and* R(5).

Proof. Suppose the opposite is true: (σ_1, u_1) produces (ρ, x) according to R(2), and (σ_2, u_2) produces (ρ, x) according to R(5). This implies that in ρ, x is matched to both u_1^* and u_2^*, which means $u_1 = u_2$. By Observation 2, this means $\sigma_1 = \sigma_2$, which contradicts the fact that the same $(\sigma, u) \in R_n$ cannot use two different rules to produce the same $(\rho, x) \in Q$. □

LEMMA 11 (two in R(1), R(3), R(6)). *Each $(\rho, x) \in Q$ cannot be produced from all three of* R(1), R(3), *and* R(6).

Proof. Assume the opposite is true. Suppose (σ_1, u_1) produces (ρ, x) using R(1); then $x = u_1^*$ (hence, x is a neighbor of u_1 in G) and u_1 is unmatched in ρ. Suppose (σ_2, u_2) produces (ρ, x) using R(3); then $x = u_2$ is unmatched in σ_2, and matched in ρ. Suppose (σ_3, u_3) produces (ρ, x) using R(6); then u_3 is matched in ρ, u_3^* is unmatched in ρ, and x is a neighbor (in G) of u_3^*.

By Observation 2, all of u_1, u_2, and u_3 are distinct. In particular, observe that $u_1 = x^* = u_2^* \neq u_3^*$; hence, all of u_1, u_2, and u_3^* are distinct (since u_2 is matched in ρ, but the other two are not).

Now, suppose we start from $\sigma_2 = \rho_x^n$ and promote $x = u_2$ resulting in ρ. Observe that u_2 changes from unmatched in σ_2 to matched in ρ, and both u_1 and u_3^* are unmatched in ρ. By Fact 2, at least one of u_1 and u_3^* is unmatched in σ_2; however, both u_1 and u_3^* are neighbors of $x = u_2$ (in G), which is unmatched in σ_2. This contradicts that fact that in any permutation, two unmatched nodes cannot be neighbors in G. □

We have finally finished the case analysis, and can conclude the f has injectivity at most 3, thereby achieving the boundary constraint.

3.4. Lower bounding the performance ratio by linear programming formulation. Combining all the proved constraints, the following LP_n gives a lower bound on the performance ratio when Ranking is run on a graph with n nodes. It is not surprising that the optimal value of LP_n decreases as n increases (although our proof does not rely on this). In section 4, we analyze the continuous relaxation LP_∞ in order to give a lower bound for all finite LP_n, thereby proving a lower bound on the performance ratio of Ranking:

$$
\begin{array}{llrl}
\mathsf{LP}_n & \min & \frac{1}{n}\sum_{t=1}^{n} x_t & \\
& \text{s.t.} & x_1 = 1, & \\
& & x_{t-1} - x_t \geq 0, & t \in [2..n], \\
& & \left(1 - \frac{t-1}{n}\right) x_t + \frac{2}{n}\sum_{i=1}^{t-1} x_i \geq 1, & t \in [2..n], \\
& & x_n + \frac{3}{2n}\sum_{t=1}^{n} x_t \geq 1, & \\
& & x_t \geq 0, & t \in [n].
\end{array}
$$

4. Analyzing LP_n via continuous LP_∞ relaxation. In this section, we analyze the limiting behavior of LP_n by solving its continuous LP_∞ relaxation, which contains both monotone and boundary condition constraints. We develop new duality and complementary slackness characterizations to solve for the optimal value of LP_∞, thereby giving a lower bound on the performance ratio of Ranking.

4.1. Continuous linear programming relaxation. To form a continuous linear program LP_∞ from LP_n, we replace the variables x_t's with a function variable z that is continuous in $[0,1]$ and differentiable almost everywhere in $[0,1]$. The dual LD_∞ contains a real variable γ, and function variables w and y, where y is continuous in $[0,1]$ and differentiable almost everywhere in $[0,1]$. In the rest of this paper, we use "$\forall\theta$" to denote "for almost all θ," which means for all but a measure zero set.

Continuity requirement. In other literature [19, 14] concerning continuous linear programming, it is often only required that the functions concerned are measurable. However, we require z and y to be continuous *everywhere* in $[0,1]$, which is essential in deriving weak duality for LP_∞ and LD_∞.

It is not hard to see that x_i corresponds to $z(\frac{i}{n})$, but perhaps it is less obvious how LD_∞ is formed. We remark that one could consider the limiting behavior of the dual of LP_n to conclude that LD_∞ is the resulting program. We show in section 4.2 that the pair $(\mathsf{LP}_\infty, \mathsf{LD}_\infty)$ is actually a special case of a more general class of primal-dual continuous linear programming. First, we show in Lemma 12 that LP_∞ is a relaxation of LP_n:

$$
\begin{array}{lll}
\mathsf{LP}_\infty & \min & \int_0^1 z(\theta)d\theta \\
 & \text{s.t.} & z(0) = 1, \\
 & & z'(\theta) \le 0 \qquad \forall\theta \in [0,1], \\
 & & (1-\theta)z(\theta) + 2\int_0^\theta z(\lambda)d\lambda \ge 1 \qquad \forall\theta \in [0,1], \\
 & & z(1) + \frac{3}{2}\int_0^1 z(\theta)d\theta \ge 1, \\
 & & z(\theta) \ge 0 \qquad \forall\theta \in [0,1].
\end{array}
$$

$$
\begin{array}{lll}
\mathsf{LD}_\infty & \max & \int_0^1 w(\theta)d\theta + \gamma - y(0) \\
 & \text{s.t.} & (1-\theta)w(\theta) + 2\int_\theta^1 w(\lambda)d\lambda + \frac{3\gamma}{2} + y'(\theta) \le 1 \qquad \forall\theta \in [0,1], \\
 & & \gamma - y(1) \le 0, \\
 & & \gamma, y(\theta), w(\theta) \ge 0 \qquad \forall\theta \in [0,1].
\end{array}
$$

LEMMA 12 (continuous linear programming relaxation). *The optimal value of* LP_n *is at least the optimal value of* LP_∞.

Proof. We fix n, and let p_n and p_∞ be the optimal values for LP_n and LP_∞, respectively. For the sake of contradiction, suppose $p_\infty = p_n + \delta$ for some $\delta > 0$, which may be dependent on n. Let x be an optimal solution for LP_n. In order to obtain a contradiction, our goal is to construct a feasible solution z (from x) for LP_∞ that has an objective value smaller than $p_n + \delta$.

The rest of the proof proceeds in the following manner. We first construct a natural step function $\hat{z}$ in $[0,1]$ corresponding to x. Although $\hat{z}$ is not continuous, it satisfies the constraints of LP_∞ and the objective function evaluates to $\int_0^1 \hat{z}(\theta)d\theta = p_n$. Then we modify $\hat{z}$ into a feasible solution z for LP_∞, increasing the objective value by less than δ.

Recall that x is an optimal solution for LP_n. Define a step function $\hat{z}$ in interval $[0,1]$ as follows: $\hat{z}(0) := 1$ and $\hat{z}(\theta) := x_t$ for $\theta \in \left(\frac{t-1}{n}, \frac{t}{n}\right]$ and $t \in [n]$. It follows that

$$\int_0^1 \hat{z}(\theta)d\theta = \sum_{t=1}^{n} \int_{\frac{t-1}{n}}^{\frac{t}{n}} \hat{z}(\theta)d\theta = \frac{1}{n}\sum_{t=1}^{n} x_t = p_n.$$

We now prove that $\hat{z}$ satisfies the constraints of LP_∞. Clearly $\hat{z}(0) = 1$ and $\hat{z}'(\theta) = 0$ for $\theta \in [0,1] \setminus \{\frac{t}{n} : 0 \le t \le n, t \in \mathbb{Z}\}$.

Evolving constraint. For every $\theta \in (0,1]$, suppose $\theta \in \left(\frac{t-1}{n}, \frac{t}{n}\right]$, and we have

$$\begin{aligned}
(1-\theta)\hat{z}(\theta) + 2\int_0^\theta \hat{z}(\lambda)d\lambda &= (1-\theta)x_t + 2\sum_{i=1}^{t-1}\int_{\frac{i-1}{n}}^{\frac{i}{n}} \hat{z}(\lambda)d\lambda + 2\int_{\frac{t-1}{n}}^{\theta} \hat{z}(\lambda)d\lambda \\
&= (1-\theta)x_t + \frac{2}{n}\sum_{i=1}^{t-1} x_i + 2\left(\theta - \frac{t-1}{n}\right)x_t \\
&= \left(1 - \frac{t-1}{n} + \left(\theta - \frac{t-1}{n}\right)\right)x_t + \frac{2}{n}\sum_{i=1}^{t-1} x_i \\
&\ge \left(1 - \frac{t-1}{n}\right)x_t + \frac{2}{n}\sum_{i=1}^{t-1} x_i \\
&\ge 1,
\end{aligned}$$

where the last inequality follows from the feasibility of x in LP_n. The above inequality holds trivially at $\theta = 0$.

Boundary constraint. Using the fact that $\int_0^1 \hat{z}(\theta)d\theta = \frac{1}{n}\sum_{t=1}^n x_t$ we have

$$\hat{z}(1) + \frac{3}{2}\int_0^1 \hat{z}(\theta)d\theta = x_n + \frac{3}{2n}\sum_{t=1}^{n} x_t \ge 1,$$

where the last inequality follows from the feasibility of x in LP_n.

Achieving continuity. Next we define a continuous function z as follows. Let $\epsilon := \min\{\delta, \frac{1}{2n}\}$. The idea is that for $t \in [2..n]$, at the transition point $\frac{t-1}{n}$, we let the function drop gradually from x_{t-1} to x_t, as θ increases from $\frac{t-1}{n}$ to $\frac{t-1}{n} + \epsilon$.

Formally, let $z(\theta) := x_1 = 1$ for $\theta \in [0, \frac{1}{n}]$. For each $t \in \{2, \dots, n\}$, let

$$z(\theta) := \begin{cases} x_t + \frac{x_{t-1} - x_t}{\epsilon}\left(\frac{t-1}{n} + \epsilon - \theta\right), & \theta \in \left(\frac{t-1}{n}, \frac{t-1}{n} + \epsilon\right], \\ x_t, & \theta \in \left(\frac{t-1}{n} + \epsilon, \frac{t}{n}\right]. \end{cases}$$

Observe that z is continuous on $[0,1]$. Moreover, it is differentiable almost everywhere, and has nonpositive derivative whenever it is differentiable. To check that z is feasible, observe that $z \ge \widehat{z}$ on $[0,1]$, and so z also satisfies the evolving and the boundary constraints.

Finally, observe that for each $t \in [2..n]$, when we let the function z drop gradually at the transition point $\frac{t-1}{n}$, the difference in area under the curves z and $\widehat{z}$ on the interval $[\frac{t-1}{n}, \frac{t-1}{n} + \epsilon]$ is $\frac{(x_{t-1}-x_t)\epsilon}{2}$. Hence, the total difference in area under the curves z and $\widehat{z}$ is $\sum_{t=2}^{n} \frac{(x_{t-1}-x_t)\epsilon}{2} = \frac{(x_1-x_n)\epsilon}{2} \le \frac{\epsilon}{2}$.

It follows that $\int_0^1 z(\theta)d\theta \le \int_0^1 \widehat{z}(\theta)d\theta + \frac{\epsilon}{2} = p_n + \frac{\epsilon}{2} < p_n + \delta$, obtaining the desired contradiction. ∎

4.2. Primal-dual for a general class of continuous linear programming. We study a class of continuous linear programs CP that includes LP_∞ as a special case. In particular, CP contains monotone and boundary conditions as constraints. Let $K, L > 0$ be two real constants. Let A, B, C, F be measurable functions on $[0, 1]$. Let D be a nonnegative measurable function on $[0, 1]^2$. We describe CP and its dual CD, following which we present weak duality and complementary slackness conditions. In CP, the variable is a function z that is continuous on $[0, 1]$ and differentiable almost everywhere in $[0, 1]$; in CD, the variables are a real number γ, and measurable functions w and y, where y is continuous on $[0, 1]$ and differentiable almost everywhere in $[0, 1]$.

$$\mathsf{CP} \quad \min \ p(z) = \int_0^1 A(\theta)z(\theta)d\theta$$

$$\text{s.t.} \quad z(0) = K, \tag{1}$$

$$z'(\theta) \le 0 \qquad \forall\theta \in [0, 1], \tag{2}$$

$$B(\theta)z(\theta) + \int_0^\theta D(\theta, \lambda)z(\lambda)d\lambda \ge C(\theta) \qquad \forall\theta \in [0, 1], \tag{3}$$

$$z(1) + \int_0^1 F(\theta)z(\theta)d\theta \ge L, \tag{4}$$

$$z(\theta) \ge 0 \qquad \forall\theta \in [0, 1].$$

$$\mathsf{CD} \quad \max \ d(w, y, \gamma) = \int_0^1 C(\theta)w(\theta)d\theta + L\gamma - Ky(0)$$

$$\text{s.t.} \quad B(\theta)w(\theta) + \int_\theta^1 D(\lambda, \theta)w(\lambda)d\lambda + F(\theta)\gamma + y'(\theta) \le A(\theta) \qquad \forall\theta \in [0, 1], \tag{5}$$

$$\gamma - y(1) \le 0, \tag{6}$$

$$\gamma, y(\theta), w(\theta) \ge 0 \qquad \forall\theta \in [0, 1].$$

Lemma 13 (weak duality and complementary slackness). *Suppose z and (w, y, γ) are feasible solutions to* CP *and* CD*, respectively. Then, $d(w, y, \gamma) \le p(z)$. Moreover, suppose z and (w, y, γ) satisfy the following complementary slackness conditions:*

$$z'(\theta)y(\theta) = 0 \ \ \forall\theta \in [0, 1], \tag{7}$$

$$\left[B(\theta)z(\theta) + \int_0^\theta D(\theta, \lambda)z(\lambda)d\lambda - C(\theta)\right] w(\theta) = 0 \ \ \forall\theta \in [0, 1], \tag{8}$$

$$\left[z(1) + \int_0^1 F(\theta)z(\theta)d\theta - L\right] \gamma = 0, \tag{9}$$

$$\left[B(\theta)w(\theta) + \int_\theta^1 D(\lambda, \theta)w(\lambda)d\lambda + F(\theta)\gamma + y'(\theta) - A(\theta)\right] z(\theta) = 0 \ \ \forall\theta \in [0, 1], \tag{10}$$

$$(\gamma - y(1))z(1) = 0. \tag{11}$$

Then, z and (w, y, γ) are optimal for CP *and* CD*, respectively, and achieve the same optimal value.*

Proof. Using the primal and dual constraints, we obtain

$$
\begin{aligned}
d(w,y,\gamma) &= \int_0^1 C(\theta)w(\theta)d\theta + L\gamma - Ky(0) \\
&\le \int_0^1 \left[B(\theta)z(\theta) + \int_0^\theta D(\theta,\lambda)z(\lambda)d\lambda\right] w(\theta)d\theta + L\gamma - Ky(0) && \text{by (3)} \\
&= \int_0^1 \left[B(\theta)w(\theta) + \int_\theta^1 D(\lambda,\theta)w(\lambda)d\lambda\right] z(\theta)d\theta + L\gamma - Ky(0) && (*) \\
&\le \int_0^1 [A(\theta) - F(\theta)\gamma - y'(\theta)]\, z(\theta)d\theta + L\gamma - Ky(0) && \text{by (5)} \\
&= \int_0^1 A(\theta)z(\theta)d\theta - \int_0^1 y'(\theta)z(\theta)d\theta + \left[L - \int_0^1 F(\theta)z(\theta)d\theta\right]\gamma - Ky(0) \\
&\le \int_0^1 A(\theta)z(\theta)d\theta - \int_0^1 y'(\theta)z(\theta)d\theta + z(1)\gamma - Ky(0) && \text{by (4)} \\
&= \int_0^1 A(\theta)z(\theta)d\theta - y(1)z(1) + y(0)z(0) + \int_0^1 z'(\theta)y(\theta)d\theta + z(1)\gamma - Ky(0) && (**) \\
&\le \int_0^1 A(\theta)z(\theta)d\theta + (\gamma - y(1))z(1) && \text{by (1), (2)} \\
&\le \int_0^1 A(\theta)z(\theta)d\theta && \text{by (6)} \\
&= p(z),
\end{aligned}
$$

where in (*) we change the order of integration by using Tonelli's theorem on nonnegative measurable function g: $\int_0^1 \int_0^\theta g(\theta,\lambda)d\lambda d\theta = \int_0^1 \int_\theta^1 g(\lambda,\theta)d\lambda d\theta$; and in (**) we use integration by parts and the fundamental theorem of calculus, as both y and z are continuous everywhere in $[0,1]$. Moreover, if z and (w,y,γ) satisfy conditions (7)–(11), then all the inequalities above hold with equality. Hence, $d(w,y,\gamma) = p(z)$; so z and (w,y,γ) are optimal for CP and CD, respectively. ∎

4.3. Lower bound for the performance ratio. The performance ratio of $\mathsf{Ranking}$ is lower bounded by the optimal value of LP_∞. We analyze this optimal value by applying the primal-dual method to LP_∞. In particular, we construct a primal feasible solution z and a dual feasible solution (w,y,γ) that satisfy the complementary slackness conditions presented in Lemma 13. Note that LP_∞ and LD_∞ are achieved from CP and CD by setting $K := 1$, $L := 1$, $A(\theta) := 1$, $B(\theta) := 1-\theta$, $C(\theta) := 1$, $D(\lambda,\theta) := 2$, $F(\theta) := \frac{3}{2}$.

We give some intuition on how z is constructed. An optimal solution to LP_∞ should satisfy the primal constraints with equality for some θ. Setting the constraint $(1-\theta)z(\theta) + 2\int_0^\theta z(\lambda)d\lambda \ge 1$ to equality, we get $z(\theta) = 1-\theta$. However this function violates the last constraint $z(1) + \frac{3}{2}\int_0^1 z(\theta)d\theta \ge 1$. Since z is decreasing, we need to balance between $z(1)$ and $\int_0^1 z(\theta)d\theta$.

The intuition is that we set $z(\theta) := 1-\theta$ for $\theta \in [0,\mu]$ and allow z to decrease until θ reaches some value $\mu \in (0,1)$, and then $z(\theta) := 1-\mu$ stays constant for $\theta \in [\mu,1]$. To determine the value of μ, note that the equation $z(1) + \frac{3}{2}\int_0^1 z(\theta)d\theta = 1$ should be satisfied, since otherwise we could construct a feasible solution with smaller objective value by decreasing the value of $z(\theta)$ for $\theta \in (\mu,1]$. It follows that $(1-\mu) + \frac{3}{2}(1-\mu+\frac{\mu^2}{2}) = 1$, that is, the value of $\mu \in (0,1)$ is determined by the equation $3\mu^2 - 10\mu + 6 = 0$.

After setting z, we construct (w,y,γ) carefully to fit the complementary slackness conditions. Formally, we set z and (w,y,γ) as follows with their plots in Figure 1.

LEMMA 14 (optimality of z and (w,y,γ)). *The solutions z and (w,y,γ) constructed above are optimal for LP_∞ and LD_∞, respectively. In particular, the optimal value of LP_∞ is $\frac{2(5-\sqrt{7})}{9} \approx 0.523$.*

$$z(\theta) = \begin{cases} 1-\theta, & 0 \le \theta \le \mu, \\ 1-\mu, & \mu < \theta \le 1, \end{cases}$$

$$w(\theta) = \begin{cases} \frac{2(1-\mu)^2}{(5-3\mu)(1-\theta)^3}, & 0 \le \theta \le \mu, \\ 0, & \mu < \theta \le 1, \end{cases}$$

$$y(\theta) = \begin{cases} 0, & 0 \le \theta \le \mu, \\ \frac{2(\theta-\mu)}{5-3\mu}, & \mu < \theta \le 1, \end{cases}$$

$$\gamma = \frac{2(1-\mu)}{5-3\mu},$$

where $\mu = \frac{5-\sqrt{7}}{3}$ is a root of the equation

$$3\mu^2 - 10\mu + 6 = 0.$$

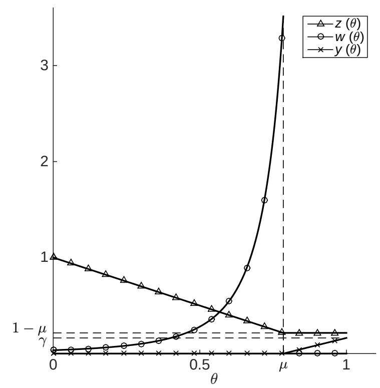

FIG. 1. *Optimal* z *and* (w, y, γ).

Proof. We list the complementary slackness conditions and check that they are satisfied by z and (w, y, γ). Then Lemma 13 gives the optimality of z and (w, y, γ).

(7) $z'(\theta)y(\theta) = 0$: we have $y(\theta) = 0$ for $\theta \in [0, \mu)$ and $z'(\theta) = 0$ for $\theta \in (\mu, 1]$.

(8) $[(1-\theta)z(\theta) + 2\int_0^\theta z(\lambda)d\lambda - 1]w(\theta) = 0$: we have

$$(1-\theta)z(\theta) + 2\int_0^\theta z(\lambda)d\lambda - 1 = (1-\theta)^2 + 2\left(\theta - \frac{\theta^2}{2}\right) - 1 = 0$$

for $\theta \in [0, \mu)$ and $w(\theta) = 0$ for $\theta \in (\mu, 1]$.

(9) $[z(1) + \frac{3}{2}\int_0^1 z(\theta)d\theta - 1]\gamma = 0$: we have

$$z(1) + \frac{3}{2}\int_0^1 z(\theta)d\theta - 1 = (1-\mu) + \frac{3}{2}\left(1 - \mu + \frac{\mu^2}{2}\right) - 1 = 0$$

by the definition of μ.

(10) $[(1-\theta)w(\theta) + 2\int_\theta^1 w(\lambda)d\lambda + \frac{3\gamma}{2} + y'(\theta) - 1]z(\theta) = 0$: for $\theta \in [0, \mu)$, we have

$$\begin{aligned} &(1-\theta)w(\theta) + 2\int_\theta^1 w(\lambda)d\lambda + \frac{3\gamma}{2} + y'(\theta) - 1 \\ &= \frac{2(1-\mu)^2}{(5-3\mu)(1-\theta)^2} + 2\int_\theta^\mu w(\lambda)d\lambda + \frac{3(1-\mu)}{5-3\mu} + 0 - 1 = 0, \end{aligned}$$

and for $\theta \in (\mu, 1]$, we have

$$\begin{aligned} &(1-\theta)w(\theta) + 2\int_\theta^1 w(\lambda)d\lambda + \frac{3\gamma}{2} + y'(\theta) - 1 \\ &= \frac{3\gamma}{2} + y'(\theta) - 1 = \frac{3(1-\mu)}{5-3\mu} + \frac{2}{5-3\mu} - 1 = 0. \end{aligned}$$

(11) $(\gamma - y(1))z(1) = 0$: we have $\gamma - y(1) = \frac{2(1-\mu)}{5-3\mu} - \frac{2(1-\mu)}{5-3\mu} = 0$.

Moreover, the optimal value of LP_∞ is $\int_0^1 z(\theta)d\theta = 1 - \mu + \frac{\mu^2}{2} = \frac{2(5-\sqrt{7})}{9} \approx 0.523$. ∎

Proof of Theorem 1. The expected ratio of Ranking is lower bounded by the optimal value of LP_n. Hence, the theorem follows from Lemmas 12 and 14. ∎

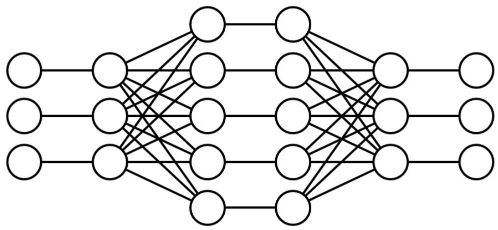

Fig. 2. *Double bomb graph.*

Table 1
Experimental performance results.

n	20	50	100	200	500
$\epsilon = 0.33$	0.7344	0.7297	0.7281	0.7272	0.7267
$\epsilon = 0.63$	0.7314	0.7267	0.7253	0.7244	0.7240
$\epsilon = 0.90$	0.7318	0.7274	0.7260	0.7252	0.7248

5. Hardness example. Our experiments suggest that the hardness result in [12] can be slightly improved by adjusting the parameter of their hard instance. An example of the graph is shown in Figure 2. We define the graph as follows:

Let G be a bipartite graph over $2(3+\epsilon)n$ vertices (u_i's and v_i's). Define the edges by the adjacency matrix A. ($A[i][j] = 1$ if there is an edge between u_i and v_j.)

$$A[i][j] = \begin{cases} 1 & \text{if } i = j, \\ 1 & \text{if } i \in [1, n], j \in (n, (2+\epsilon)n], \\ 1 & \text{if } i \in (n, (2+\epsilon)n], j \in ((2+\epsilon)n, (3+\epsilon)n], \\ 0 & \text{otherwise.} \end{cases}$$

We run experiments on different n's and ϵ's (each for $100,000$ times) and get the following results; see Table 1.

We observe that when $\epsilon \approx 1 - 1/e$ the ratio is minimized for this kind of graph. It is close to 0.724 in this case. We leave as future work to analyze it theoretically.

6. Open problems and future work. In this paper, we show that the Ranking algorithm has a performance ratio strictly above 0.5 for the Oblivious Matching Problem on general graphs. It is shown in [1] that a weighted version of the Ranking algorithm achieves a performance ratio strictly above 0.5 for the problem on general node-weighted graphs, in which each node u has weight w_u, and the objective is to maximize the total weight of matched nodes.

It is interesting to consider edge-weighted versions of oblivious or online matching problems. Partial solutions have been given in [1] for the following variants.

Edge-weighted Oblivious Matching Problem. A set V of nodes is given, together with a weight w_{uv} attached to each pair of nodes u and v. If (after probing the pair $\{u, v\}$) there is an edge $e \in E$ between u and v, then its weight is given by w_{uv}. The goal of the problem is to decide a probing order on the pairs that produces a matching with maximum total edge weights. Note that the node-weighted setting is a special case of the edge-weighted setting when $w_{uv} = w_u + w_v$.

It can be easily shown that the greedy algorithm that probes pairs $\{u, v\}$ in non-increasing order of their weights w_{uv} achieves a performance ratio exactly 0.5.

It is shown in [1] that when the number of distinct weights is bounded by a constant, a performance ratio strictly above 0.5 can be achieved. However, for the problem with arbitrary weights, whether a performance ratio strictly above 0.5 can be achieved remains unknown.

Another closely related open problem is the edge-weighted online bipartite matching problem with arbitrary arrival order of online nodes and free disposal.

Edge-weighted online bipartite matching with free disposal. Suppose we are given a set V of *offline* nodes, while the *online* nodes U come in an arbitrary order. When an online node $u \in U$ 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 can match u to any of the offline nodes v. Even if 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) weight of the final matching to that of a maximum weight matching in hindsight.

It was shown that ratio 0.5 can be achieved by some greedy algorithm [7]. For the special case when each online node has bounded degree, a performance ratio strictly larger than 0.5 has been achieved in [1]. Achieving a performance ratio strictly above 0.5 for the general case remains an open problem.

We believe that the techniques developed in this paper can shed light on solving the above two open problems. We think that it is possible to prove a performance ratio strictly above 0.5 using a randomized algorithm similar to the weighted Ranking algorithm used in [2, 1].

REFERENCES

[1] M. Abolhassani, T. H. Chan, F. Chen, H. Esfandiari, M. Hajiaghayi, H. Mahini, and X. Wu, *Beating ratio* 0.5 *for weighted oblivious matching problems*, in ESA, P. Sankowski and C. D. Zaroliagis, eds., LIPIcs Leibniz Int. Proc. Inform. 57, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Wadern, Germany, 2016, 3, https://doi.org/10.4230/LIPIcs.ESA.2016.3.

[2] G. Aggarwal, G. Goel, C. Karande, and A. Mehta, *Online vertex-weighted bipartite matching and single-bid budgeted allocations*, in SODA, D. Randall, ed., SIAM, Philadelphia, 2011, pp. 1253–1264, https://doi.org/10.1137/1.9781611973082.95.

[3] S. Alaei, J. D. Hartline, R. Niazadeh, E. Pountourakis, and Y. Yuan, *Optimal auctions vs. anonymous pricing*, in FOCS, V. Guruswami, ed., IEEE Computer Society, Los Alamitos, CA, 2015, pp. 1446–1463, https://doi.org/10.1109/FOCS.2015.92.

[4] S. Anand, N. Garg, and A. Kumar, *Resource augmentation for weighted flow-time explained by dual fitting*, in SODA, Y. Rabani, ed., SIAM, Philadelphia, 2012, pp. 1228–1241, http://portal.acm.org/citation.cfm?id=2095213&CFID=63838676&CFTOKEN=79617016.

[5] J. Aronson, M. E. Dyer, A. M. Frieze, and S. Suen, *Randomized greedy matching* II, Random Structures Algorithms, 6 (1995), pp. 55–74, https://doi.org/10.1002/rsa.3240060107.

[6] M. E. Dyer and A. M. Frieze, *Randomized greedy matching*, Random Structures Algorithms, 2 (1991), pp. 29–46, https://doi.org/10.1002/rsa.3240020104.

[7] J. Feldman, N. Korula, V. S. Mirrokni, S. Muthukrishnan, and M. Pál, *Online ad assignment with free disposal*, in WINE, S. Leonardi, ed., Lecture Notes in Comput. Sci. 5929, Springer, Berlin, 2009, pp. 374–385, https://doi.org/10.1007/978-3-642-10841-9_34.

[8] G. Goel and A. Mehta, *Online budgeted matching in random input models with applications to adwords*, in SODA, S. Teng, ed., SIAM, Philadelphia, 2008, pp. 982–991, http://dl.acm.org/citation.cfm?id=1347082.1347189.

[9] G. Goel and P. Tripathi, *Matching with our eyes closed*, in FOCS, IEEE Computer Society, Los Alamitos, CA, 2012, pp. 718–727, https://doi.org/10.1109/FOCS.2012.19.

[10] G. Goel and P. Tripathi, *Matching with our Eyes Closed*, arXiv:1306.2988, 2013.

[11] G. P. Guruganesh and S. Singla, *Online matroid intersection: Beating half for random arrival*, http://arxiv.org/abs/1512.06271.

[12] C. Karande, A. Mehta, and P. Tripathi, *Online bipartite matching with unknown distributions*, in STOC, L. Fortnow and S. P. Vadhan, eds., ACM, New York, 2011, pp. 587–596, https://doi.org/10.1145/1993636.1993715.

[13] R. M. Karp, U. V. Vazirani, and V. V. Vazirani, *An optimal algorithm for on-line bipartite matching*, in STOC, H. Ortiz, ed., ACM, New York, 1990, pp. 352–358, https://doi.org/10.1145/100216.100262.

[14] N. Levinson, *A class of continuous linear programming problems*, J. Math. Anal. Appl., 16 (1966), pp. 73–83, https://doi.org/10.1016/0022-247X(66)90187-9.

[15] M. Mahdian and Q. Yan, *Online bipartite matching with random arrivals: An approach based on strongly factor-revealing LPs*, in STOC, L. Fortnow and S. P. Vadhan, eds., ACM, New York, 2011, pp. 597–606, https://doi.org/10.1145/1993636.1993716.

[16] S. Micali and V. V. Vazirani, *An O(sqrt(|v|) |E|) algorithm for finding maximum matching in general graphs*, in FOCS, IEEE Computer Society, Long Beach, CA, 1980, pp. 17–27, https://doi.org/10.1109/SFCS.1980.12.

[17] M. Poloczek and M. Szegedy, *Randomized greedy algorithms for the maximum matching problem with new analysis*, in FOCS, IEEE Computer Society, Los Alamitos, CA, 2012, pp. 708–717, https://doi.org/10.1109/FOCS.2012.20.

[18] A. E. Roth, T. Sönmez, and M. U. Ünver, *Pairwise kidney exchange*, J. Econom. Theory, 125 (2005), pp. 151–188, https://doi.org/10.1016/j.jet.2005.04.004.

[19] W. F. Tyndall, *A duality theorem for a class of continuous linear programming problems*, SIAM, 13 (1965), pp. 644–666, http://www.jstor.org/stable/2946458.

[20] Y. Wang and S. C. Wong, *Two-sided online bipartite matching and vertex cover: Beating the greedy algorithm*, in ICALP, M. M. Halldórsson, K. Iwama, N. Kobayashi, and B. Speckmann, eds., Lecture Notes in Comput. Sci. 9134, Springer, Berlin, 2015, pp. 1070–1081, https://doi.org/10.1007/978-3-662-47672-7_87.