Towards a Better Understanding of Randomized Greedy Matching

ZHIHAO GAVIN TANG, ITCS, Shanghai University of Finance and Economics, China and Key Laboratory of Interdisciplinary Research of Computation and Economics (Shanghai University of Finance and Economics), Ministry of Education, China
XIAOWEI WU, IOTSC, University of Macau, China
YUHAO ZHANG, John Hopcroft Center for Computer Science, Shanghai Jiao Tong University, China

There has been a long history of studying randomized greedy matching algorithms since the work by Dyer and Frieze (RSA 1991). We follow this trend and consider the problem formulated in the oblivious setting, in which the vertex set of a graph is known to the algorithm, but not the edge set. The algorithm can make queries for the existence of the edge between any pair of vertices but must include the edge into the matching if it exists, i.e., as in the query-commit model by Gamlath et al. (SODA 2019). We revisit the Modified Randomized Greedy (MRG) algorithm by Aronson et al. (RSA 1995) that is proved to achieve a $(0.5 + \epsilon)$-approximation. In each step of the algorithm, an unmatched vertex is chosen uniformly at random and matched to a randomly chosen neighbor (if exists). We study a weaker version of the algorithm named Random Decision Order (RDO) that, in each step, randomly picks an unmatched vertex and matches it to an arbitrary neighbor (if exists). We prove that the RDO algorithm provides a 0.639-approximation for bipartite graphs and 0.531-approximation for general graphs. As a corollary, we substantially improve the approximation ratio of MRG.

Furthermore, we generalize the RDO algorithm to the edge-weighted case and prove that it achieves a 0.501 approximation ratio. This result solves the open question by Chan et al. (SICOMP 2018) and Gamlath et al. (SODA 2019) about the existence of an algorithm that beats greedy in edge-weighted general graphs, where the greedy algorithm probes the edges in descending order of edge-weights. We also present a variant of the algorithm that achieves a $(1 - 1/e)$-approximation for edge-weighted bipartite graphs, which generalizes the $(1 - 1/e)$ approximation ratio of Gamlath et al. (SODA 2019) for the stochastic setting to the case when the realizations of edges are arbitrarily correlated, where in the stochastic setting, there is a known probability associated with each pair of vertices that indicates the probability that an edge exists between the two vertices, when the pair is probed.

CCS Concepts: • **Theory of computation → Graph algorithms analysis**; **Approximation algorithms analysis**.

Additional Key Words and Phrases: Oblivious Matching, Randomized Greedy, Approximation Algorithms

A preliminary version of this article appeared in STOC 2020 [28] and the current article is the result of a merge with [27]. Zhihao Gavin Tang is supported by National Natural Science Foundation of China, Grant No. 61902233. Xiaowei Wu is funded by the Science and Technology Development Fund (FDCT) Macau SAR (File no. 0014/2022/AFJ, 0085/2022/A, 0143/2020/A3, SKL-IOTSC-2021-2023). Yuhao Zhang is supported by National Natural Science Foundation of China, Grant No. 62102251.

Authors' addresses: Zhihao Gavin Tang, ITCS, Shanghai University of Finance and Economics, Shanghai, China and Key Laboratory of Interdisciplinary Research of Computation and Economics (Shanghai University of Finance and Economics), Ministry of Education, Shanghai, China, tang.zhihao@mail.shufe.edu.cn; Xiaowei Wu, IOTSC, University of Macau, Macau, China, xiaoweiwu@um.edu.mo; Yuhao Zhang, John Hopcroft Center for Computer Science, Shanghai Jiao Tong University, Shanghai, China, zhang_yuhao@sjtu.edu.cn.

0004-5411/2023/8-ART111 $15.00
https://doi.org/XXXXXXX.XXXXXXX

ACM Reference Format:
Zhihao Gavin Tang, Xiaowei Wu, and Yuhao Zhang. 2023. Towards a Better Understanding of Randomized Greedy Matching. *J. ACM* 37, 4, Article 111 (August 2023), 32 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Maximum matching is a fundamental problem in combinatorial optimization. Although the problem admits an efficient polynomial time algorithm, the greedy heuristic that repeatedly includes an edge sharing no common endpoints with the selected edges to the matching (until no edge can be included) is widely used and observed to have good performance [22, 29]. In addition, greedy algorithms have attracted attention due to their application in kidney exchange problems [25]. In such scenarios, information about the graph is incomplete and greedy algorithms are the most commonly implemented algorithms as there are no structural properties about the graph to exploit.

To this end, the *oblivious matching* model is formulated [4, 14]. Consider a graph $G = (V, E)$ in which the vertices in V are revealed, while the edges in E are unknown. The algorithm picks a permutation of the unordered pairs of vertices $\binom{V}{2}$. Each pair is probed one-by-one according to the permutation to form a matching greedily. In particular, when the pair (u, v) is probed, if the edge exists and both u, v are unmatched, then we match the two vertices; otherwise, we continue to the next pair. This setting is also known as the *query-commit* model. We compare the performance of an algorithm to the size of a maximum matching. Note that any permutation induces a maximal matching and, hence, any probing algorithm provides a 0.5-approximation. On the other hand, no deterministic algorithms can do better than this ratio. The interesting question then is to design randomized algorithms with approximation ratios greater than 0.5.

1.1 Prior Works

The first natural attempt at randomization is to permute all pairs of vertices uniformly at random. Unfortunately, Dyer and Frieze [9] proved that it fails to beat the 0.5 approximation ratio. The first non-trivial theoretical guarantee for the problem is provided by Aronson et al. [1]. In the paper, they proposed the Modified Randomized Greedy (MRG) algorithm, in each step of which an unmatched vertex u is chosen uniformly at random and matched to a randomly chosen unmatched neighbor. If u has no unmatched neighbor, then it will be left unmatched permanently. They proved a $(0.5+1/400000)$ lower bound on the approximation ratio of MRG. The analysis was later improved by Poloczek and Szegedy [24] to $0.5 + 1/256$.[1] Different randomized greedy algorithms are also studied, including Ranking [4, 20, 21, 23], FRanking [16–18], etc. In the Ranking algorithm, a random permutation of the vertices is chosen, which induces a lexicographical ordering on the pairs of vertices for probing. Similarly, in the FRanking algorithm, a random permutation of the vertices is chosen. However, instead of inducing a lexicographical ordering of pairs, in each step of the algorithm an *arbitrary* unmatched vertex is chosen and matched to the first unmatched neighbor (if exists) according to the (randomly chosen) permutation of vertices. Interestingly, all these algorithms fall into the family of vertex-iterative algorithms [14]:

(1) Iterates through the vertices according to a *decision order*.
(2) In the iteration of u, probe u with other vertices according to the *preference order* of u.

There are $|V| + 1$ different orders to be specified, including a decision order in the first step and $|V|$ individual preference orders in the second step. The names are chosen based on the following equivalent statement of the framework: 1) let the vertices make decisions sequentially according to the decision order; 2) if a vertex is not matched at its decision time, it would choose its favorite unmatched neighbor, according to its individual preference.

[1] It was pointed out by Chan et al. [4] that their paper contains some gaps in the proof.

As introduced above, existing algorithms differ in the way the orders are generated. In particular, MRG samples the decision order and the preference orders independently and uniformly at random. Ranking [21] samples only one random permutation uniformly and uses it as the decision order and the common preference order. The FRanking algorithm also samples only one random permutation and uses it as the common preference order, but uses an arbitrary decision order.

The oblivious matching problem is closely related to the online bipartite matching problem, in which vertices on one side of a bipartite graph is given offline while the other side of vertices arrive online. The algorithm must decide which offline neighbor each online vertex matches to upon its arrival immediately and irrevocably. It was observed [14, 24] that the approximation ratio of Ranking for the oblivious matching problem on bipartite graphs is the same as that for online bipartite matching when the online vertices arrive following an order chosen uniformly at random. Therefore, the result of Mahdian and Yan [23] translates to a 0.696-approximation ratio for Ranking on bipartite graphs in the oblivious setting. For general graphs, Ranking is proved to be 0.526-approximation [3, 4][2]. The FRanking algorithm[3] has been studied in the fully online matching model [17]. The fully online matching model generalizes the online bipartite matching problem by allowing all vertices of the graph to arrive online, where each vertex has an arrival time and a deadline (both are decided by the adversary). The online algorithm can make irrevocable matching decisions at any moment between two vertices that have arrived but have not reached their deadlines. Huang et al. [17] studied the FRanking algorithm, in which the decision order is given by ordering the vertices from earliest deadline to the latest. Their results can be transformed to an oblivious matching algorithm that uses an arbitrary decision order[4] and a common random preference order. The competitive ratios 0.567 and 0.521 achieved for bipartite graphs [18] and general graphs [16] are then inherited in our setting.

1.2 Our Contributions

Despite the successful progress of these results, we lack a systematic understanding of the roles of the decision order and preference orders. In this paper, we revisit the MRG algorithm and study separately the two kinds of randomness on the decision order and the preference orders.

Consider a variant of the MRG algorithm in which the decision order is fixed arbitrarily while the preference orders are drawn independently and uniformly for all vertices. We refer to this algorithm as Independent Random Preferences (IRP). Due to the instance from [9], IRP does not guarantee an approximation ratio better than 0.5.[5] In other words, the random decision order is necessary and crucial for MRG to achieve an approximation ratio strictly larger than 0.5. On the other hand, it is not clear from the previous analysis [1, 24] whether the random decision order alone is sufficient. In this paper, we answer this question affirmatively and conclude that the randomness of decision order plays a more important role than the preference orders for the MRG algorithm. Consider the following variant of the MRG algorithm.

Random Decision Order (RDO). We sample the decision order uniformly at random and fix the individual preference orders arbitrarily.

Theorem 1.1. *RDO is a* 0.639*-approximation algorithm for the oblivious matching problem on bipartite graphs.*

[2]Goel and Tripathi (FOCS 2012) claimed that Ranking is 0.56-approximation, but later withdrew the paper when they discovered a bug in their proof.

[3]In their paper, the algorithm is named Ranking. We use FRanking to distinguish it from the original Ranking algorithm, since they have different behavior when adapted to the oblivious matching setting.

[4]The feature of arbitrary decision time comes from the online nature of their setting, which the algorithm has no control of.

[5]This is observed in the online matching literature. For completeness, we include a formal discussion in Appendix B.

Theorem 1.2. *RDO is a* 0.531*-approximation algorithm for the oblivious matching problem on general graphs.*

Since MRG can be regarded as sampling a preference order uniformly at random for each of the vertices and running the RDO algorithm with the sampled preference orders, we have the following corollary on the approximation ratio of MRG.

Corollary 1.3. *MRG is a* 0.531*-approximation algorithm for the oblivious matching problem.*

Our results substantially improve the $(0.5 + 1/400000)$ ratio by Aronson et al. [1] and the $(0.5 + 1/256)$ ratio by Poloczek and Szegedy [24] for general graphs. Besides, our result beats the state-of-the-art 0.526-approximation by Ranking [3] for the oblivious matching problem on general graphs. We also show some hardness results for the RDO algorithm. First, we give a 0.646 upper bound on the approximation ratio of RDO on bipartite graphs by experiments, which is very close to our 0.639 lower bound. Second, we prove that RDO does not guarantee an approximation ratio larger than 0.625 on general graphs. Together with Theorem 1.1, it gives a separation on the approximation ratio of RDO on bipartite and general graphs. To the best of our knowledge, no such separation has been shown to exist for other algorithms, including MRG, Ranking and FRanking.

We summarize our results (together with some existing results) in Table 1.

	Decision	Preferences	Bipartite	General
Ranking	Common Random		0.696 [23]	0.526 [3, 4]
FRanking	Arbitrary	Common Random	0.567 [18]	0.521 [16]
MRG	Independent Random		$0.5 + \epsilon$ [1, 24] $\rightarrow$ **0.531** (Section 4)	
IRP	Arbitrary	Independent Random	0.5 [9]	
RDO	Random	Arbitrary	**0.639** (Section 3)	**0.531** (Section 4)

Table 1. A summary of prior works and our results on unweighted graphs. Our results are marked in bold. "Common random" means a single random permutation is used for all orders. "Independent random" means all orders are generated independently.

Extension to Edge-weighted Case. Interestingly, the better understanding of MRG towards RDO leads us to a natural generalization of the algorithm to the edge-weighted oblivious matching problem. In the edge-weighted case, the graph is associated with a weight function w defined on all pairs of vertices that are known to the algorithm. If an edge (u, v) exists, its weight is given by w_{uv}. It is straightforward to check that the greedy algorithm that probes the pairs in descending order of the weights provides a 0.5-approximation. It was proposed as an open question in [4] to study the existence of a better than 0.5-approximation algorithm. In this paper, we answer it positively by studying the following generalization of RDO.

Perturbed Greedy. Each vertex $u \in V$ draws a rank $y_u \in [0, 1]$ independently and uniformly. Then we probe all pairs of vertices (u, v) in descending order of their perturbed weights[6], which is defined as $(1 - g(\min\{y_u, y_v\})) \cdot w_{uv}$, where g is a non-decreasing function to be fixed later.

Theorem 1.4. *There exists a function* g *with which Perturbed Greedy provides a* 0.501*-approximation for the edge-weighted oblivious matching problem.*

It is easy to show that the Perturbed Greedy algorithm degenerates to RDO for unweighted graphs, for any increasing function g. In particular, consider y_u as the *decision time* of vertex u. We list all vertices in ascending order of their decision times. This is equivalent to sampling a random

[6]Break ties arbitrarily and consistently.

decision order. Upon the decision of u, all remaining edges incident to u would have the same perturbed weight $1 - g(y_u)$. By breaking ties arbitrarily and consistently, it is equivalent to having arbitrary individual preference orders, i.e. the RDO algorithm.

Furthermore, we show that for bipartite graphs, a variant of Perturbed Greedy that uses only one side of the ranks achieves an approximation ratio of $(1 - 1/e)$.

Theorem 1.5. *There exists a variant of Perturbed Greedy that achieves a $(1 - 1/e)$-approximation for the edge-weighted oblivious matching problem on bipartite graphs.*

Extension to Stochastic Matching. Another well-studied maximum matching model is the *stochastic matching* problem [2, 5, 6, 13, 26]. In the stochastic setting, in addition to the weight w_{uv} associated with each pair (u, v), there is a probability p_{uv}. When a pair (u, v) is probed, the edge exists with probability p_{uv}, and the existences of all edges are independent random events. Equivalently, it is assumed that the underlying graph is generated by sampling each edge independently with the existence probability. Recently, Gamlath et al. [13] proposed a $(1 - 1/e)$-approximation algorithm for the problem on bipartite graphs, which is the first result bypassing the 0.5 barrier.

They proposed two open questions as follows.[7]

(1) Does algorithm with approximation ratio strictly above 0.5 exist on general graphs?
(2) What approximation ratio can be obtained if the existences of edges are correlated?

Observe that any algorithm in the oblivious setting can be applied in the stochastic setting by simply ignoring the extra stochastic information, e.g., existence probabilities of edges. Moreover, the approximation ratio is preserved since it holds for every realization of the underlying graph, even when the edges are sampled from correlated distributions. Thus, as a corollary of Theorem 1.4 and Theorem 1.5, we answer both questions positively.

Corollary 1.6. *For the edge-weighted stochastic matching problem, there exists a 0.501-approximation algorithm for general graphs, and a $(1 - 1/e)$-approximation algorithm for bipartite graphs, even when the existences of edges are arbitrarily correlated.*

1.3 Our Techniques

We will prove our results starting with RDO on bipartite graphs, then RDO on general graphs, and finally Perturbed Greedy on edge-weighted general graphs, in progressive order of difficulty. Our analysis builds on the randomized primal dual framework introduced by Devanur et al. [8] and recently developed in a sequence of results [16, 18, 19]. Roughly speaking, we split the gain of each matched edge to its two endpoints. By proving that the expected combined gain of any pair of neighbors (u, v) is at least $r \cdot w_{uv}$, we show that the approximation ratio is at least r. However, our approach differs from previous works in a way that we only look at the pairs that appear in some fixed maximum matching. We refer to such pairs as perfect partners. In order to prove that the algorithm achieves a r-approximation, we observe that it suffices to show the expected combined gain of (u, v) is at least $r \cdot w_{uv}$ for all perfect partners (u, v). As far as we know, our result is the first successful application of the randomized primal dual technique to the oblivious matching problem, or to any edge-weighted maximum matching problems[8].

Unweighted Bipartite Graphs. As discussed in the previous subsection, the random decision order can be generated by sampling decision times independently and uniformly from $[0, 1]$ for all vertices and listing the vertices in ascending order of their decision times. We shall use this

[7]The open questions were raised in their SODA 2019 conference presentation.
[8]Independent from our conference version, Fahrbach et al. [10, 15] studied the edge-weighted online bipartite matching problem using the randomized primal-dual technique and show that it obtains a competitive ratio of 0.5086.

interpretation for the purpose of analysis. We use y_u to denote the decision time of vertex u, and it plays a similar role as the rank of u in the Ranking algorithm in our analysis. In the Ranking algorithm, the decision order and all preference orders are given by the common random order given by listing the vertices in ascending order of their ranks. Consider a pair of neighbors (u, v), where u has an earlier decision time than v. Existing analysis of Ranking relies on an important structural property that whenever v is unmatched, u must be matched to some vertex with a rank smaller than y_v. However, we have nothing to say about u's choice on its decision time, since we assume it makes the decision based on its individual preference. In particular, when u matches a vertex at the decision time of u, we say that u is *active* and the other vertex is *passive*, e.g., imagine that the matching decision is initiated by vertex u. Since the matching decision by the active vertex can be arbitrary, whenever an edge is matched, we split the gain between the two endpoints based on the current decision time, instead of the rank/decision time of the passive vertex. We specify such a gain sharing rule so that by fixing the decision times of all vertices other than u, v arbitrarily and taking the randomness over y_u, y_v, the expected combined gain of u and v is at least 0.639.

Unweighted General Graphs. For bipartite graphs, our analysis relies on a crucial structural property that if u is matched when its neighbor v is removed from the graph, then u remains matched when v is added back, no matter what decision time vertex v has. Going from bipartite graphs to general graphs takes away this nice property. A similar issue arises in the analysis of Ranking [4] and FRanking [16]. The recent work by Huang et al. [16] introduces a notion of *victim* to tackle the problem. Roughly speaking, they call an unmatched vertex v the victim of its neighbor z if v becomes matched when z is removed. Then they define a compensation rule in which each active vertex sends an amount of gain, named *compensation*, to its victim (if any). In this work, we propose a new definition of victim together with a compensation rule that is arguably more intuitive and has a clearer structure. Suppose vertex z actively matches u in some matching. If u is matched with its *perfect partner* v when z is removed, we call v the *victim* of z. After sharing the gain as we have described for bipartite graphs, we let each active vertex send a portion of its gain to its victim if the victim is unmatched. This compensation rule is designed to retrieve some extra gain for vertices u, v when the aforementioned property for bipartite graphs fails to hold.

Weighted General Graphs. For unweighted graphs, the contribution of each matched vertex to the object is uniform. Therefore, it suffices to characterize the status of a vertex as matched or unmatched. However, being matched is no longer a meaningful signature on edge-weighted graphs. E.g., being matched in an ϵ-weighted edge and being matched in a large weight edge should be treated differently. This observation prevents the victim notion of Huang et al. [16] from generalizing to the edge-weighted case. More specifically, under their notion, the victim v of z will become matched when z is removed. However, there is no guarantee on the weight of the edge that v matches. In contrast, we define v as the victim of z only if v matches its perfect partner u when z is removed, in which case the edge v matches has weight w_{uv}. Consequently, our definition of victim generalizes naturally to the weighted case, and the analysis for unweighted graphs extends with a minimum change of the compensation rule. In particular, we fix a function h that decides the amount of compensation one would like to send. The compensation rule works in the following way. Suppose v is the victim of z. We know that z is matched with v's perfect partner u. Our compensation rule ensures that, after the compensation, the gain of v is at least $h(y_z) \cdot w_{zu}$. Note that the amount of compensation that z sends depends on the current gain of v. This generalization further justifies the advantage of our new notion of victim, and we believe the new notion will find more applications in other matching problems on general graphs.

1.4 Other Related Works

For the hardness results of the oblivious matching problem, Goel and Tripathi [14] showed a 0.7916 upper bound on the approximation ratio of any algorithm for unweighted graphs. When restricted to the family of vertex iterative algorithms, they give a stronger bound of 0.75. For the MRG algorithm, Dyer and Frieze [9]'s bomb graphs give a 2/3 upper bound on its approximation ratio. For the Ranking algorithm, the hard instance provided by Mahdian and Yan [23] implies a 0.727 upper bound, which was later improved to 0.724 by Chan et al. [4].

Since the publication of the preliminary versions of this paper [27, 28], there have been subsequent works on the edge-weighted stochastic matching problem. For the problem on bipartite graphs, Derakhshan and Farhadi [7] proposed an algorithm with an approximation ratio of 0.6335, beating the $(1 - 1/e)$ ratio by Gamlath et al. [13]. For the problem on general graphs, our approximation ratio 0.501 shown in Corollary 1.6 has been improved to 0.533 by Fu et al. [12].

1.5 Organization

We give the formal definition of our algorithms, together with some basic properties, in Section 2. As a warm-up analysis, we present our general framework for proving the approximation ratio using randomized primal-dual in Section 3, where Theorem 1.1 (for unweighted bipartite graphs) is proved. Then we extend our framework to unweighted general graphs in Section 4, where we introduce our new compensation rule, and prove Theorem 1.2. The analysis for weighted general graphs is presented in Section 5, where we prove Theorem 1.4. The proof of Theorem 1.5 (for weighted bipartite graphs) is included in Section 6. We conclude the paper and propose some open questions in Section 7.

2 PRELIMINARIES

Recall the general algorithm for the edge-weighted oblivious matching problem as follows.

Algorithm 1 Perturbed Greedy

Fix a non-decreasing function $g : [0, 1] \to [0, 1]$.
Each vertex u independently draws a rank $y_u \in [0, 1]$ uniformly at random.
Probe pairs in descending order of their perturbed weights $(1 - g(\min\{y_u, y_v\}))w_{uv}$.

We use $\vec{y}$ to denote the ranks of vertices, and $M(\vec{y})$ to denote the matching produced by our algorithm with ranks $\vec{y}$. For any vertex u, we use $M_u(\vec{y})$ to denote the matching produced by our algorithm on $G - \{u\}$, i.e. when u is removed from the graph. Fix any maximum weight matching M^*. The approximation ratio is the ratio between the expected total weight of edges in $M(\vec{y})$, and the total weight of edges in M^*.

The gain sharing framework we use in this paper is formally defined as follows.

Lemma 2.1. *If there exist non-negative random variables $\{\alpha_u\}_{u\in V}$ depending on $\vec{y}$ such that*

- *for every $\vec{y}$, $\sum_{u\in V} \alpha_u = \sum_{(u,v)\in M(\vec{y})} w_{uv}$;*
- *for every $(u, v) \in M^*$, $\underset{\vec{y}}{\mathbf{E}}[\alpha_u + \alpha_v] \geq r \cdot w_{uv}$,*

then the approximation ratio of our algorithm is at least r.

Proof. The approximation ratio of Perturbed Greedy is given by

$$\frac{\underset{\vec{y}}{\mathbf{E}}\left[\sum_{(u,v)\in M(\vec{y})} w_{uv}\right]}{\sum_{(u,v)\in M^*} w_{uv}} = \frac{\underset{\vec{y}}{\mathbf{E}}\left[\sum_{u\in V} \alpha_u\right]}{\sum_{(u,v)\in M^*} w_{uv}} \geq \frac{\underset{\vec{y}}{\mathbf{E}}\left[\sum_{(u,v)\in M^*} (\alpha_u + \alpha_v)\right]}{\sum_{(u,v)\in M^*} w_{uv}} \geq \frac{\sum_{(u,v)\in M^*} r \cdot w_{uv}}{\sum_{(u,v)\in M^*} w_{uv}} = r,$$

where the first equality and first inequality hold using the assumptions from the lemma. □

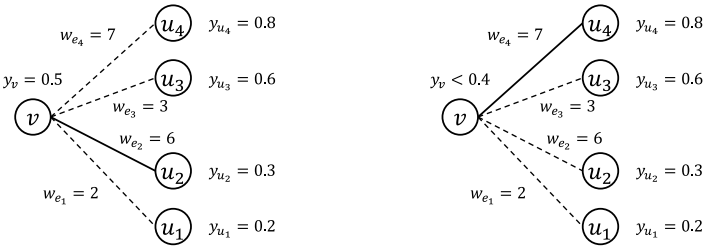
Next, we define the notion of active and passive vertices for edge-weighted general graphs and provide the monotonicity on the ranks.

Definition 2.2 (Active, Passive). For every edge (u, v) added to the matching by Perturbed Greedy with $y_u < y_v$, we say that u is active and v is passive.

Lemma 2.3 (Monotonicity). *Consider any matching $M(\vec{y})$ and any vertex v. If v is passive or unmatched, then there exists some threshold $y \le y_v$ such that if we change y_v to be any value in $(y, 1)$, the matching remains unchanged; if we change y_v to be any value in $(0, y)$, then v becomes active. Moreover, the weight of the edge v actively matches is non-increasing w.r.t. $y_v \in (0, y)$.*

Proof. Fix any vertex v and let E_v be the set of edges $(u, v) \in E$ such that $y_v < y_u$. By definition, each edge $e \in E_v$ has perturbed weight $(1 - g(y_v)) \cdot w_{uv}$ and thus if we increase y_v, the perturbed weights of edges in E_v decrease. Note that $|E_v|$ may also decrease when y_v increases.

Suppose v is passive in $M(\vec{y})$, which means that v is matched with a neighbor z with $y_z < y_v$. Now consider the moment when edge (z, v) is probed. Since the probing is successful, we know that if any edge e in E_v has been probed before this moment, the probing must be unsuccessful, i.e., the endpoint other than v of edge e is already matched. Hence if we increase y_v and observe the change in the resulting matching, then all these probes remain unsuccessful. Therefore v gets matched by the same vertex z, and nothing is changed to the matching. The case when v is unmatched is similar. On the other hand, if we decrease y_v then edges in E_v will have larger perturbed weights and will be probed earlier[9]. If the probing of any of these edges is successful (when y_v is sufficiently small), then v will become active (see Figure 1 for an example). Consequently, imagine that we increase y_v gradually from 0 to 1, then once v becomes passive or unmatched, the matching remains unchanged afterward. Thus there exists threshold $y < y_v$ such that v is active when $y_v \in (0, y)$; passive or unmatched otherwise.



(a) Matching $M(\vec{y})$ with $y_v = 0.5$. (b) The resulting matching if $y_v < 0.4$.

Fig. 1. An illustration on how different values of y_v change the matching. Suppose that $g(y) = y$ and in matching $M(\vec{y})$ we have $y_v = 0.5$ (see Figure 1a). Then the perturbed weights of edges e_1, e_2, e_3, e_4 are $1.6, 4.2, 1.5, 3.5$, respectively. Therefore, edge e_2 is probed first, and v is passively matched to u_2. The matching remains the same when we increase y_v because the increase in y_v only decreases the perturbed weights of edges in $E_v = \{e_3, e_4\}$, which does not affect the matching result. However, if we decrease y_v slightly, then the perturbed weights of edges e_1, e_2 remain the same while the perturbed weights of edges e_3, e_4 would increase. Specifically, when y_v is slightly below 0.4, the perturbed weight of e_4 is larger than 4.2 and hence will be probed before e_2 (whose perturbed weight is 4.2). Then v will actively match u_4 in the resulting matching.

Finally, for the last argument, suppose v actively matches u when $y_v = y' \in (0, y)$. Consider the resulting matching if we decrease y_v. Observe that the perturbed weight of edges in E_v will

[9]It is possible that more edges are included into E_v as y_v decreases.

increase. Thus when edge (v, u) is probed, either v is already matched (with some edge with a larger perturbed weight than (v, u)), or v matches u. Since all edges in E_v is perturbed by the same factor $1 - g(y_v)$, for a smaller value of $y_v < y'$, the weight of the edge v matches is not smaller. □

2.1 Unweighted Graphs.

In this subsection, we focus on unweighted graphs and provide the standard *alternating path* property. An analog for edge-weighted graphs will be provided in Section 5.

As we have discussed in Section 1, we can equivalently interpret the algorithm as follows.

We call the rank y_u of vertex u the *decision time* of u, and let vertices act in ascending order of their decision times. If a vertex is not matched yet at its decision time, then it will match the unmatched neighbor according to its own preference order. We also call $\min\{y_u, y_v\}$ the decision time of edge $(u, v) \in E$, which is the only possible time the edge is included in the matching.

Algorithm 2 Random Decision Order

Each vertex u independently draws a rank $y_u \in [0, 1]$ uniformly at random.
At decision time y_u of u, if u is unmatched, u chooses its favourite unmatched neighbor.

Note that for each vertex, the preference order of its neighbors is arbitrary but fixed. In other words, it does not depend on the ranks $\vec{y}$ of the vertices. It is easy to see that for the unweighted case, if a vertex v is passively matched by u, then the threshold of v in Lemma 2.3 coincides with the rank y_u of u. Thus we have the following.

Corollary 2.4 (Unweighted Monotonicity). *Consider any matching $M(\vec{y})$ and any vertex v.*

- *If v is passively matched by u in $M(\vec{y})$, then if we change the decision time y_v to any value in $(y_u, 1)$, the resulting matching remains the same as $M(\vec{y})$; if we change y_v to any value in $(0, y_u)$, then v becomes active in the resulting matching.*
- *If v is actively matched in $M(\vec{y})$, then the set of unmatched neighbors v sees at its decision time grows when y_v decreases.*

The following important property characterizes the effect of the removal of a single vertex.

Lemma 2.5 (Alternating Path). *If u is matched in $M(\vec{y})$, then the symmetric difference between $M(\vec{y})$ and $M_u(\vec{y})$ is an alternating path $(u_0 = u, u_1, u_2, \ldots)$ such that for all even i, $(u_i, u_{i+1}) \in M(\vec{y})$ and for all odd i, $(u_i, u_{i+1}) \in M_u(\vec{y})$. Moreover, the decision times of edges along the path are non-decreasing. Consequently, vertices $u_1, u_3, \ldots$ are matched no later in $M(\vec{y})$ than in $M_u(\vec{y})$.*

Proof. Suppose u is matched with u_1 in $M(\vec{y})$, then after removing u, at decision time of edge (u, u_1), u_1 will no longer be matched. If u_1 is unmatched in $M_u(\vec{y})$, then the symmetric difference is a single edge (u, u_1) and the statements trivially hold. Otherwise let u_2 be matched with u_1 in $M_u(\vec{y})$. Observe that the decision time of edge (u_1, u_2) is not earlier than (u, u_1), as otherwise u_1 will remain matched with u_2 in $M(\vec{y})$. Then by induction on the number of remaining vertices, the symmetric difference between $M(\vec{y})$ and $M_{u_2}(\vec{y})$ is an alternating path P starting from u_2 such that the decision times of edges are non-decreasing. Thus the symmetric difference between $M(\vec{y})$ and $M_u(\vec{y})$ is an alternating path starting from u, u_1, u_2, followed by path P. Moreover, the decision time of (u, u_1) is not later than (u_1, u_2), and the first edge of P has decision time not earlier than (u_1, u_2), which implies the first statement. For every vertex u_i with odd index in the alternating path, since the decision time of $(u_{i-1}, u_i) \in M(\vec{y})$ is not later than $(u_i, u_{i+1}) \in M_u(\vec{y})$, compared with $M_u(\vec{y})$, u_i is matched no later in $M(\vec{y})$. □

Given the above lemma, we show the following useful property, which roughly says that any vertex v can not affect the matching status of another vertex u if u is matched before y_v.

Corollary 2.6. *Suppose at time y, vertex u is matched while vertex v is still unmatched. Then changing y_v to be any value in $(y, 1)$ does not change the matching status of u.*

Proof. Since v is unmatched when u gets matched, if v is removed, then the matching status of u is not affected. Now suppose that we insert v with $y_v \in (y, 1)$ and observe the resulting matching. If v is matched, then it must be matched at the time later than y (the time when u gets matched). In other words, the insertion of v triggers a (possibly empty) alternating path in which all edges (and therefore vertices) have decision times at least y. Hence, u is not included in the alternating path and its matching status is not affected. □

For bipartite graphs, Lemma 2.5 also implies the following nice property.

Corollary 2.7. *For bipartite graphs, if u is matched in $M_v(\vec{y})$ and v is a neighbor of u, then u is also matched in $M(\vec{y})$. Moreover, the time u is matched in $M(\vec{y})$ is not later than that in $M_v(\vec{y})$.*

Proof. By Lemma 2.5, inserting v (at any rank) creates a (possibly empty) alternating path $(v, v_1, v_2, \ldots)$. As the graph is bipartite, if u appears in the path, then it must be one of $\{v_1, v_3, \ldots\}$, which is matched not later than that in $M_v(\vec{y})$. Otherwise, its matching status is not affected. □

3 UNWEIGHTED BIPARTITE GRAPHS

Recall that for the unweighted case, vertices act in ascending order of their decision times, and each unmatched vertex chooses its favorite unmatched neighbor. We first give the gain sharing rule for every matched pair $(u, v) \in M(\vec{y})$.

Gain Sharing. Whenever u actively chooses v at time y_u, let $\alpha_u = g(y_u)$ and $\alpha_v = 1 - g(y_u)$, where g is a non-decreasing function to be fixed later.

General Framework. Recall that to prove an approximation ratio of r, it suffices to show that $\mathop{\mathbf{E}}_{\vec{y}}[\alpha_u + \alpha_v] \geq r$ for every $(u, v) \in M^*$. Fix such a pair (u, v), and fix the decision times of all vertices other than u, v arbitrarily. For ease of notation, we use $M(y_u, y_v)$ to denote the matching produced by our algorithm when u, v's decision times are y_u and y_v, respectively. In the following, we will give a lower bound $f(y_u, y_v)$ of $\alpha_u + \alpha_v$ for each $M(y_u, y_v)$, and show that there exists an appropriate function g such that $\int_0^1 \int_0^1 f(y_u, y_v) dy_u dy_v \geq 0.639$, finishing the proof of Theorem 1.1.

From now on, we consider an arbitrarily fixed pair of perfect partners $(u, v) \in M^*$, and lower bound $\mathbf{E}[\alpha_u + \alpha_v]$ using the randomness of y_u and y_v. We first consider the matching $M(1, 1)$, i.e., when u and v have the latest decision times compared with other vertices. Depending on whether u, v are matched to each other, we divide our analysis into two parts.

3.1 Symmetric Case: $(u, v) \in M(1, 1)$

In this case, u and v are not chosen by any other vertex. At time 1, u and v are matched together.

Lemma 3.1. *If $(u, v) \in M(1, 1)$, then when $y_u < y_v$, u is active in $M(y_u, y_v)$ and v is unmatched before time y_u; when $y_v < y_u$, v is active in $M(y_u, y_v)$, and u is unmatched before time y_v.*

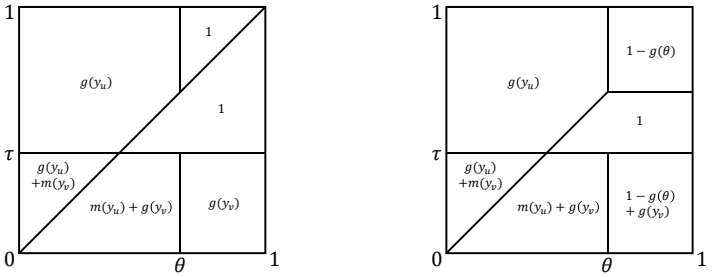
Proof. Suppose $y_u < y_v$. Consider the first time t when one of u, v (say, $x \in \{u, v\}$) is matched. Obviously, $t \leq y_u$. If $t < y_u$, then x must be chosen by a vertex z at its decision time $y_z = t$. By Lemma 2.3, we have $(x, z) \in M(1, 1)$, which violates the assumption of the lemma. Thus $t = y_u$, i.e, u is active, and v is unmatched before time y_u. The case when $y_v < y_u$ is similar. □

Now suppose that we decrease y_u gradually from 1 to 0 and observe the resulting matching $M(y_u, 1)$. By Corollary 2.4, the set of unmatched neighbors of u at time y_u grows when y_u decreases. Hence there exists a transition time θ such that v is no longer u's favorite vertex when $y_u < \theta$. In other words, when $y_u > \theta$, u actively matches v in $M(y_u, 1)$; when $y_u < \theta$, u actively matches a vertex other than v in $M(y_u, 1)$. Moreover, by Corollary 2.6, for all $y_v > y_u$, the matching status of u in $M(y_u, y_v)$ is the same as in $M(y_u, 1)$. Thus we have (refer to Figure 2a)

- $\alpha_u + \alpha_v = 1$ when $y_u > \theta$ and $y_v > y_u$;
- $\alpha_u = g(y_u)$ when $y_u < \theta$ and $y_v > y_u$.

Similarly, we decrease y_v gradually from 1 to 0 and study $M(1, y_v)$. Let τ be the transition time such that u is v's favorite neighbor if and only if $y_v > \tau$. Then we have (refer to Figure 2a)

- $\alpha_u + \alpha_v = 1$ when $y_v > \tau$ and $y_u > y_v$;
- $\alpha_v = g(y_v)$ when $y_v < \tau$ and $y_u > y_v$.



(a) Symmetric case: u, v match each other (b) Asymmetric case: u is chosen at θ

Fig. 2. Unweighted bipartite graphs: the horizontal and vertical axes correspond to y_u and y_v respectively. For each region, the formula written serves as a lower bound of $\alpha_u + \alpha_v$.

We refer to these gains as the basic gains of our analysis as they come immediately after we properly define θ, τ. Next, we study the matching status of the vertex with later decision time and achieve some extra gains, where we crucially use the bipartiteness of the graph.

Lemma 3.2 (Extra Gain). *For all $y_u < \theta$ and $y_v < \tau$, both u and v are matched in $M(y_u, y_v)$.*

Proof. By definition, when $y_u < \theta$, u actively matches a vertex other than v in $M(y_u, 1)$. Thus removing v does not change the matching status of u. In other words, u is matched in $M_v(y_u, y_v)$. By Corollary 2.7, u remains matched in $M(y_u, y_v)$. Similarly, we have that v is matched in $M(y_u, y_v)$ for all $y_v < \tau$, which finishes the proof. □

Let $m(y) \stackrel{\text{def}}{=} \min\{g(y), 1 - g(y)\}$. It is easy to see that whenever u is matched (actively or passively), $\alpha_u \geq m(y_u)$. In summary, we have the following lower bound (refer to Figure 2a).

$$\begin{aligned} \mathop{\mathbf{E}}_{y_u, y_v}[\alpha_u + \alpha_v] \geq & \int_0^\theta (1 - y_u) g(y_u) dy_u + \frac{1}{2}(1 - \theta)^2 + \int_0^\tau (1 - y_v) g(y_v) dy_v + \frac{1}{2}(1 - \tau)^2 \\ & + \int_0^\theta \min\{y_u, \tau\} m(y_u) dy_u + \int_0^\tau \min\{y_v, \theta\} m(y_v) dy_v. \end{aligned} \tag{1}$$

3.2 Asymmetric Case: $(u, v) \notin M(1, 1)$

In this case, at least one of u, v is matched before time 1. Without loss of generality (w.l.o.g.), suppose that u is matched at time $\theta < 1$, and strictly earlier than v. Observe that u must be passive since

$y_u = 1$. Let z be the vertex that actively matches u. Then we have $y_z = \theta$. Intuitively, u is the "luckier" vertex compared with v since it is favored by a vertex with early decision time. Indeed, u would remain matched even when v is removed from the graph.

First, observe that when both u and v have decision times larger than θ, u is always matched by z, and thus $\alpha_u = 1 - g(\theta)$. When $y_u < \theta$, u must be active in $M(y_u, 1)$, since at time y_u, u is unmatched and has unmatched neighbors z and v (with later decision times). Moreover, by Corollary 2.6, u is active in $M(y_u, y_v)$ as long as $y_u < \theta$ and $y_v > y_u$. Thus $\alpha_u = g(y_u)$. Similarly, for all $y_v < \theta$ and $y_u > y_v$, v is active and $\alpha_v = g(y_v)$.

Again, we decrease y_v gradually from θ to 0 and study $M(1, y_v)$. Observe that at time y_v, u is an unmatched neighbor of v. Then there exists a transition time τ such that u is the favourite neighbor of v when $y_v \in (\tau, \theta)$; and v matches a vertex other than u when $y_v < \tau$. In summary, we have the following basic gains (refer to Figure 2b)

- $\alpha_u = 1 - g(\theta)$ when $y_u, y_v > \theta$;
- $\alpha_u + \alpha_v = 1$ when $y_v \in (\tau, \theta)$ and $y_u > y_v$;
- $\alpha_u = g(y_u)$ when $y_u < \theta$ and $y_v > y_u$;
- $\alpha_v = g(y_v)$ when $y_v < \tau$ and $y_u > y_v$.

Next, we retrieve some extra gains. Again, the following holds only for bipartite graphs.

Lemma 3.3 (Extra Gain). *When $y_u < \theta$ and $y_v < \tau$, both u and v are matched in $M(y_u, y_v)$. When $y_u > \theta$ and $y_v < \tau$, $\alpha_u \geq 1 - g(\theta)$.*

Proof. Consider when $y_u < \theta$ and $y_v = 1$. According to the previous discussion, u has two unmatched neighbors z and v at time y_u. Thus u would still be actively matched even if we remove v from the graph. That is, u is active in $M_v(y_u, 1)$. Then by Corollary 2.7, after inserting v at any rank, u remains matched. In other words, u is matched in $M(y_u, y_v)$ for all $y_u < \theta$.

By definition of τ, v actively matches a vertex other than u in $M(1, y_v)$ for all $y_v < \tau$. Thus v is active in $M_u(1, y_v)$, and matched in $M(y_u, y_v)$ for every y_u by Corollary 2.7.

Now we consider the second statement when $y_u > \theta$ and $y_v < \tau$. Observe that in $M(y_u, 1)$, z matches u at time θ while at this moment v is unmatched. Removing v does not affect z and u, i.e. z actively matches u in $M_v(y_u, 1)$. Then by Corollary 2.7, inserting v at any rank does not increase the time at which u gets matched. Hence in $M(y_u, y_v)$, u is passively matched at time no later than θ, which implies $\alpha_u \geq 1 - g(\theta)$ by the monotonicity of g. □

Adding these extra gains to the basic gains, we have (refer to Figure 2b)

$$\begin{aligned}\mathop{\mathbf{E}}_{y_u, y_v}[\alpha_u + \alpha_v] \geq \int_0^\theta (1 - y_u) g(y_u) dy_u + \int_0^\tau (1 - y_v) g(y_v) dy_v + (1 - \theta)(1 - \theta + \tau)(1 - g(\theta)) \\ + \int_0^\theta \min\{y_u, \tau\} m(y_u) dy_u + \int_0^\tau y_v m(y_v) dy_v + \frac{1}{2}(2 - \tau - \theta)(\theta - \tau).\end{aligned} \tag{2}$$

Lower Bound of Approximation Ratio. To complete the analysis, it remains to find a non-decreasing function $g : [0, 1] \to [0, 1]$ so that the minimum (over all possible values of θ, τ) of Equations (1) and (2) is at least 0.639. We prove Theorem 1.1 by fixing g to be a step function and running a factor revealing LP. See Appendix A for a detailed discussion.

4 UNWEIGHTED GENERAL GRAPHS

In this section, we consider the case of unweighted general graphs. Since Corollary 2.7 holds only for bipartite graphs, the extra gains we proved in the previous section cease to hold for general graphs. It is easy to check that, by applying the previous analysis while only having the basic gains, we are not able to break the 0.5 barrier on the approximation ratio.

The same difficulty arises in the fully online matching problem [16]. The authors bypass it by introducing a novel concept of "victim". They call a vertex v the victim of vertex w in $M(\vec{y})$ if (1) v is a neighbor of w; (2) w is active and v is unmatched; (3) v is matched in $M_w(\vec{y})$. Intuitively, v is unmatched in $M(\vec{y})$ because of the existence of w. It is then shown that in every $M(y_u, y_v)$, either u, v are both matched, or the unmatched vertex is the victim of some vertex and receives compensation. In either case, the improved analysis breaks the 0.5 barrier.

In this paper, we introduce a new notion of victim and compensation, which is arguably clearer and more fundamental than the notion given in [16]. Recall that we fixed a maximum matching M^* and call u and v perfect partners of each other if $(u, v) \in M^*$.

Definition 4.1 (Victim). Suppose in $M(\vec{y})$, z actively matches u and v is the perfect partner of u. Then we call v the *victim* of z in $M(\vec{y})$ if u and v match each other in $M_z(\vec{y})$.

Note that the definition of victim is with respect to a fixed $\vec{y}$. For example, it possible that v is the victim of z in $M(\vec{y})$ while z is the victim of v in $M(\vec{y}')$, for another $\vec{y}'$. Moreover, by definition, a victim is not necessarily unmatched[10]. Intuitively, the existence of z prevents the algorithm from making the correct decision of matching u, v together. Compared with the definition of Huang et al. [16], we regard v as the victim of z even when v is matched in $M(\vec{y})$. The same definition will be applied to edge-weighted graphs in Section 5. Building upon this definition, we define the following gain sharing rule.

Gain Sharing. Let g be a non-decreasing function and h be a function that is pointwise smaller than g. Consider the following two-step gain sharing procedure in matching $M(\vec{y})$:

- Whenever u actively matches v at time y_u, let $\alpha_u = g(y_u)$ and $\alpha_v = 1 - g(y_u)$.
- For each active vertex z that has an *unmatched* victim v, decrease α_z and increase α_v by the same amount $h(y_z)$.

We refer to the second step of gain sharing as the compensation step, and the amount $h(y_z)$ of gain as the *compensation* sent from z to v. Note that the compensation step does not change $\sum_{u \in V} \alpha_u$, which means that Lemma 2.1 can still be applied. It is easy to see that the passive gain of vertex u is at least $1 - g(y_u)$ and the active gain is at least $g(y_u) - h(y_u)$. Therefore, we have the following immediately.

Fact 4.1. *If u is matched in $M(\vec{y})$, then $\alpha_u \geq m(y_u) \stackrel{def}{=} \min\{g(y_u) - h(y_u), 1 - g(y_u)\}$.*

Moreover, if v is the victim of vertex z, then either v is matched, which gives $\alpha_v \geq m(y_v)$, or v receives compensation from z, which gives $\alpha_v \geq h(y_z)$. For analysis purposes, we choose g, h so that $\min_y\{m(y)\} \geq \max_y\{h(y)\}$, i.e., the gain of a matched vertex is at least the compensation of an unmatched vertex. To help to understand, one can imagine that the compensation is a very small amount of gain compared with $m(\cdot)$. Consequently, we have the following.

Fact 4.2. *If v is the victim of vertex z in $M(\vec{y})$, then $\alpha_v \geq h(y_z)$.*

Following the same framework as for bipartite graphs, we fix a pair of perfect partners $(u, v) \in M^*$, and fix the decision times of all vertices other than u, v arbitrarily. Using the randomness of y_u and y_v, we lower bound the combined gain of vertices u and v. As before, we use $M(y_u, y_v)$ to denote the realized matching when u, v have decision times y_u and y_v, respectively. Again, we consider whether $(u, v) \in M(1, 1)$ and proceed differently.

[10]For the unweighted case, our whole analysis goes through if we require a victim to be an unmatched vertex. However, to be consistent with the weighted case, in which being matched is no longer a meaningful signature, we adopt the same definition of victim (regardless of whether it is matched or not) in the unweighted case.

4.1 Symmetric Case: $(u, v) \in M(1, 1)$

The analysis is similar to the bipartite case. Let θ be the transition time such that u actively matches v in $M(y_u, 1)$ when $y_u > \theta$; matches a vertex other than v in $M(y_u, 1)$ when $y_u < \theta$. The transition time τ of y_v is defined analogously.

Following the same analysis for bipartite graphs, we have (refer to Figure 3)

- $\alpha_u + \alpha_v = 1$ when $y_u > \theta$ and $y_v > y_u$; u is active when $y_u < \theta$ and $y_v > y_u$;
- $\alpha_u + \alpha_v = 1$ when $y_v > \tau$ and $y_u > y_v$; v is active when $y_v < \tau$ and $y_u > y_v$.

Observe that for general graphs, the gain of an active vertex u can no longer be lower bounded $g(y_u)$. However, it can be lower bounded by $g(y_u) - h(y_u)$. However, if u, v match each other, then the active vertex does not need to send compensations (recall that u, v are perfect partners).

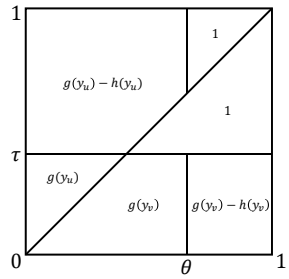


Fig. 3. Unweighted general graphs: $(u, v) \in M(1, 1)$.

For bipartite graphs, we show that both u and v are matched in $M(y_u, y_v)$ when $y_u < \theta$ and $y_v < \tau$. Unfortunately, this is not guaranteed in general graphs. However, we manage to achieve a weaker version of the extra gains that if only one of u, v is matched when $y_u < \theta$ and $y_v < \tau$, then it does not need to send compensation.

Lemma 4.2 (Extra Gain). *For all $y_u < \theta$ and $y_v < \tau$, we have $\alpha_u + \alpha_v \geq g(y_u)$ in $M(y_u, y_v)$ when $y_v > y_u$ and $\alpha_u + \alpha_v \geq g(y_v)$ when $y_u > y_v$.*

Proof. We first consider the case when $y_v > y_u$. If v is matched, then $\alpha_u + \alpha_v \geq g(y_u) - h(y_u) + m(y_v) \geq g(y_u)$. Now suppose v is unmatched. By definition, when $y_v < \tau$, in $M(1, y_v)$, vertex u is unmatched at time y_v but v actively matches a vertex other than u. Thus, v is also active in $M_u(1, y_v)$. In other words, v becomes unmatched after inserting u at decision time $y_u < \theta$. We show that in this case u need not send compensation in $M(y_u, y_v)$, which implies $\alpha_u = g(y_u)$.

Suppose u matches z in $M(y_u, y_v)$. By Lemma 2.5, removing u triggers an alternating path that starts at u and ends at v. Thus the perfect partner of z is either matched in both of $M(y_u, y_v)$ and $M_u(y_u, y_v)$; or unmatched in both. Consequently, u does not have an unmatched victim.

Symmetrically, we have $\alpha_u + \alpha_v \geq g(y_v)$ when $y_u > y_v$. □

In summary, we have the following lower bound on $\mathbf{E}[\alpha_u + \alpha_v]$. Note that u, v are symmetric and we can assume w.l.o.g. that $\tau \leq \theta$ (refer to Figure 3).

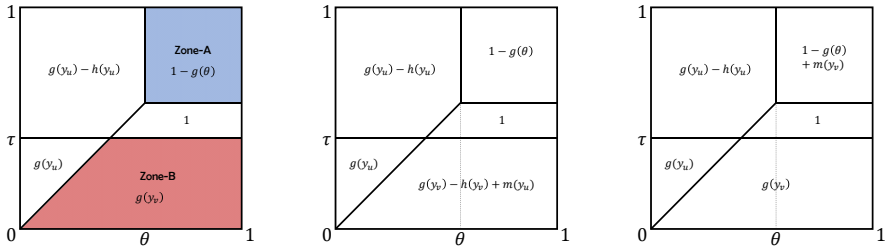
$$\begin{aligned} \mathop{\mathbf{E}}_{y_u, y_v}[\alpha_u + \alpha_v] \geq & \frac{1}{2}(1-\theta)^2 + \frac{1}{2}(1-\tau)^2 + \int_0^\tau \Big((1-y_v)g(y_v) - (1-\theta)h(y_v)\Big)dy_v \\ & + \int_0^\theta \Big((1-y_u)g(y_u) - (1-\max\{\tau, y_u\})h(y_u)\Big)dy_u. \end{aligned} \tag{3}$$

4.2 Asymmetric Case: $(u, v) \notin M(1, 1)$

As before, at least one of u, v is matched before time 1. We assume w.l.o.g. that u is matched strictly earlier than v in $M(1, 1)$, and let z be the active vertex that matches u with decision time $y_z = \theta$. First, when $y_u, y_v > \theta$, u is always matched by z, and thus $\alpha_u = 1 - g(\theta)$. When $y_u < \theta$, we know that u is active in $M(y_u, 1)$, and thus active in $M(y_u, y_v)$ as long as $y_v > y_u$ (by Corollary 2.6). Now consider $M(1, y_v)$ when $y_v < \theta$. Following the same analysis as for the bipartite case, let $\tau < \theta$ be the transition time such that v chooses u when $y_v \in (\tau, \theta)$ and chooses a vertex other than u when $y_v \in (0, \tau)$. Moreover, since v is active in $M_u(1, y_v)$ when $y_v < \tau$, following the same analysis as in Lemma 4.2, it can be shown that $\alpha_u + \alpha_v \geq g(y_u)$ when $y_v < \tau$ and $y_u < y_v$. Similarly, it is easy to show that $\alpha_u + \alpha_v \geq g(y_v)$ when $y_v < \tau$ and $y_u > y_v$[11].

In summary, we have (refer to Figure 4a)

(L1) $\alpha_u = 1 - g(\theta)$ when $y_u > \theta$ and $y_v > \theta$;
(L2) $\alpha_u + \alpha_v = 1$ when $y_v \in (\tau, \theta)$ and $y_u > y_v$;
(L3) $\alpha_u \geq g(y_u) - h(y_u)$ when $y_u < \min\{\theta, y_v\}$ and $y_v > \tau$;
(L4) $\alpha_u + \alpha_v \geq g(y_u)$ when $y_u < y_v$ and $y_v < \tau$;
(L5) $\alpha_u + \alpha_v \geq g(y_v)$ when $y_v < y_u$ and $y_v < \tau$.



(a) basic gains (b) u is matched earlier (c) v is matched earlier

Fig. 4. Simple lower bounds and the case when $(u, v) \notin M_z(1, 1)$.

Unsurprisingly, the above basic gains do not yield an approximation ratio strictly above 0.5.

Extra Gains. For convenience of discussion, we define Zone-A to be the matchings $M(y_u, y_v)$ when $y_u > \theta$ and $y_v > \theta$ (where lower bound (L1) is applied), and Zone-B to be the matchings $M(y_u, y_v)$ when $y_v < \tau, y_u > y_v$ (where lower bound (L5) is applied). In the following, we show that better lower bounds can be obtained for either (L1) or (L5). Roughly speaking, if v is unmatched in Zone-A, then either it is compensated by z (in which case (L1) can be improved), or u will be matched in Zone-B (in which case (L5) can be improved). Hence depending on the matching status of u and v in $M_z(1, 1)$, i.e., when z is removed, we divide our analysis into two cases.

4.2.1 Case 1: $(u, v) \notin M_z(1, 1)$. In this case, at least one of u, v is matched passively before time 1 in $M_z(1, 1)$. We first consider the case when u is matched strictly earlier than v. We show that in this case u is matched in Zone-B, and thus (L5) can be improved (see Figure 4b).

Lemma 4.3. *If u is matched strictly earlier than v in $M_z(1, 1)$, then u is matched in Zone-B.*

Proof. To show that u is matched in Zone-B, by Lemma 2.3 it suffices to show that u is matched in $M(1, y_v)$, for all $y_v < \tau$. Suppose otherwise, i.e., u is unmatched in $M(1, y_v)$ for some $y_v < \tau$.

[11]The key observation is that, u is matched in $M_v(1, y_v)$, and thus in every $M_v(y_u, y_v)$. This implies that after inserting v with $y_v < \tau$, either u is matched, or v does not need to send compensation.

Since z matches u in $M_v(1, y_v)$, the symmetric difference between $M(1, y_v)$ and $M_v(1, y_v)$ is an alternating path that starts from v and ends with u. Moreover, z is the second last vertex in the alternating path. Now suppose that we remove v and z simultaneously in $M(1, y_v)$. Then in the resulting matching, all vertices between v and z in the alternating path recover their matching status in $M_v(1, y_v)$, while all other vertices remain the same matching status. In particular, u remains unmatched when both v and z are removed from $M(1, y_v)$. However, since u is matched strictly earlier than v in $M_z(1, 1)$, u should remain matched to the same vertex when we further remove v, which is a contradiction. □

Given the lemma, we improve (L5) and obtain the following (refer to Figure 4b). As we will show later, this is not the bottleneck case since this lower bound is strictly larger than (7).

$$\begin{aligned}\mathop{\mathbf{E}}_{y_u, y_v}[\alpha_u + \alpha_v] \geq &(1-\theta)^2(1-g(\theta)) + \frac{1}{2}(2-\theta-\tau)(\theta-\tau) + \int_0^\tau (1-y_v)(g(y_v)-h(y_v))dy_v \\ &+ \int_0^\theta \Big((1-y_u)g(y_u) - (1-\max\{\tau, y_u\})h(y_u)\Big)dy_u + \int_0^1 \min\{\tau, y_u\}m(y_u)dy_u.\end{aligned} \tag{4}$$

Next, we consider the case when v is matched strictly earlier than u in $M_z(1, 1)$. We show that in this case, v is matched in Zone-A, which improves (L1).

Lemma 4.4. *If v is matched earlier than u in $M_z(1, 1)$, then v is matched in Zone-A.*

Proof. Consider adding z back to $M_z(1, 1)$. Since z chooses u, v remains matched in $M(1, 1)$. Moreover, all matchings $M(y_u, 1)$ for $y_u > \theta$ are the same and hence, v is matched in $M(y_u, 1)$ for all $y_u > \theta$. Finally, by Lemma 2.3 v is matched in Zone-A. □

Thus, we obtain the following lower bound (refer to Figure 4c). As we will show later, this is neither the bottleneck case since the lower bound is strictly larger than (6).

$$\begin{aligned}\mathop{\mathbf{E}}_{y_u, y_v}[\alpha_u + \alpha_v] \geq &(1-\theta)^2(1-g(\theta)) + (1-\theta)\int_\theta^1 m(y_v)dy_v + \frac{1}{2}(2-\theta-\tau)(\theta-\tau) \\ &+ \int_0^\theta (1-y_v)g(y_v)dy_v + \int_0^\theta \Big((1-y_u)\cdot g(y_u) - (1-\max\{\tau, y_u\})h(y_u)\Big)dy_u.\end{aligned} \tag{5}$$

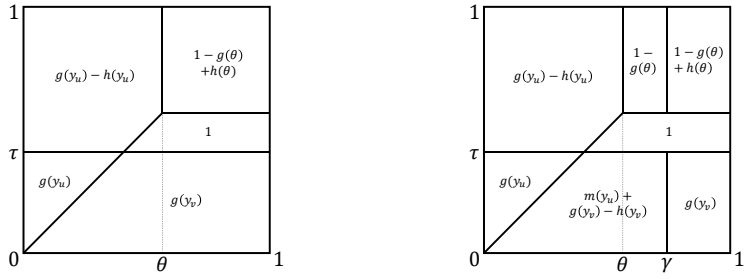
4.2.2 Case 2: $(u, v) \in M_z(1, 1)$. Now we study the second case when u, v match each other in $M_z(1, 1)$. This is where the notion of victim applies. Formally, we have the following lower bound for α_v in Zone-A.

Lemma 4.5 (Compensation). *For all $y_u > \theta$, if u matches v in $M_z(y_u, 1)$, then we have $\alpha_v \geq h(\theta)$ in $M(y_u, y_v)$, for all $y_v > \theta$.*

Proof. Consider any $y_u > \theta$ and suppose that u matches v in $M_z(y_u, 1)$. When $y_u < y_v$, u actively matches v in $M_z(y_u, y_v)$. Hence v is the victim of z and $\alpha_v \geq h(\theta)$. When $y_v < y_u$, either v actively matches u in $M_z(y_u, y_v)$ and thus v is the victim of z in $M(y_u, y_v)$, or v actively matches a vertex other than u. In the first case, $\alpha_v \geq h(\theta)$. In the second case, when we add back z to the graph, z chooses u and does not affect the matching status of v. Hence, $\alpha_v \geq m(y_v) \geq h(\theta)$. □

To apply this lemma, we first consider the case when u matches v in all $M_z(y_u, 1)$, where $y_u > \theta$. The lemma implies that $\alpha_v \geq h(\theta)$ in Zone-A. Refer to Figure 5a, we have

$$\begin{aligned}\mathop{\mathbf{E}}_{y_u, y_v}[\alpha_u + \alpha_v] \geq &(1-\theta)^2(1-g(\theta)+h(\theta)) + \frac{1}{2}(2-\tau-\theta)(\theta-\tau) + \int_0^\tau (1-y_v)g(y_v)dy_v \\ &+ \int_0^\theta \Big((1-y_u)g(y_u) - (1-\max\{\tau, y_u\})h(y_u)\Big)dy_u.\end{aligned} \tag{6}$$



(a) u matches v in $M_z(y_u, 1)$ when $y_u > \theta$ (b) u matches v in $M_z(y_u, 1)$ when $y_u > \gamma$

Fig. 5. u, v match each other in $M_z(1, 1)$.

Since $h(x) \leq m(y)$ for all $x, y \in [0, 1]$, it is obvious that this lower bound is not larger than (5). We would like to remark that in this case, we can see clearly how the compensation rule helps us achieve an approximation ratio strictly above 0.5. Without the compensation v receives in Zone-A, the lower bounds become Figure 4a, which cannot beat 0.5.

Finally, we consider the case when u does not always match v in $M_z(y_u, 1)$ for all $y_u > \theta$. Let $\gamma > \theta$ be the transition time such that u matches v in $M_z(y_u, 1)$ when $y_u > \gamma$ and matches a vertex other than v in $M_z(y_u, 1)$ when $y_u \in (\theta, \gamma)$ (see Figure 5b).

Applying Lemma 4.5 to $M(y_u, y_v)$ where $y_u > \gamma$ and $y_v > \theta$, we have $\alpha_v \geq h(\theta)$. Now consider the matching $M_z(y_u, 1)$, where $y_u < \gamma$. Intuitively, since u is matched strictly earlier than v, (in the same spirit of Lemma 4.3) u has another neighbor as a backup, and hence is still matched when we decrease y_v. Formally, we have the following.

Lemma 4.6. *When $y_u < \gamma$ and $y_v < \tau$, u is matched in $M(y_u, y_v)$.*

Proof. The proof is similar to that of Lemma 4.3. It suffices to show that u is matched in $M(\gamma, y_v)$ for all $y_v < \tau$. By definition of γ, u actively matches a vertex other than v in $M_z(\gamma, 1)$. Thus u remains active if we further remove v from the graph.

On the other hand, if u is unmatched in $M(\gamma, y_v)$, then the symmetric difference between $M(\gamma, y_v)$ and $M_v(\gamma, y_v)$ is an alternating path that starts from v and ends at u. Moreover, z is the second last vertex in the alternating path. Consequently, u remains unmatched if we remove both v and z from the graph, which is a contradiction. □

Plugging in the improved version of (L1) and (L5), we obtain the following (refer to Figure 5b).

$$\begin{aligned}\mathop{\mathbf{E}}_{y_u, y_v}[\alpha_u + \alpha_v] \geq & (1-\theta)^2(1-g(\theta)) + (1-\gamma)(1-\theta)h(\theta) + \frac{1}{2}(2-\tau-\theta)(\theta-\tau) \\ & + \int_0^\theta \Big((1-y_u)g(y_u) - (1-\max\{\tau, y_u\})h(y_u)\Big)dy_u + \int_0^\gamma \min\{y_u, \tau\}m(y_u)dy_u \\ & + \int_0^\tau \Big((1-y_v)g(y_v) - (\gamma-y_v)h(y_v)\Big)dy_v. \end{aligned} \tag{7}$$

Observe that when $\gamma = 1$, this bound degenerates to (4). It is easy to see this by comparing Figure 5b and Figure 4b.

Lower Bound of Approximation Ratio. Equipped with the previous lower bounds, it suffices to find functions $g, h : [0,1] \to [0,1]$ so that the minimum of (3), (6), (7) over all possible θ, τ, γ is maximized. Again, a factor revealing LP shows that RDO provides a 0.531-approximation, finishing the proof of Theorem 1.2.

5 WEIGHTED GENERAL GRAPH

We analyze the approximation ratio of our algorithm on weighted general graphs in this section. Recall that our algorithm probes pairs (u, v) in descending order of the perturbed weights $(1 - g(\min\{y_u, y_v\}))w_{uv}$. Sometimes it will be helpful to interpret the algorithm as replacing every (potential) edge (u, v) with two directed edges (u, v) and (v, u). Then we set the perturbed weight of a directed edge (u, v) to be $(1 - g(y_u))w_{uv}$ and probe the directed edges in descending order of their perturbed weights.

The structure of our analysis is similar to that of the unweighted case. However, we will see many of the previous properties fail when the edges are weighted. We first provide some basic properties of Perturbed Greedy for weighted graphs, which will be the building blocks of our analysis. Notably, we generalize the gain sharing and compensation rule to edge-weighted graphs.

First, we observe the following property that is analogous to Lemma 2.5 for the unweighted case, where we substitute *decision times* with *perturbed weights*. The proof is almost identical to that of Lemma 2.5, thus is omitted.

Lemma 5.1 (Weighted Alternating Path). *If u is matched in $M(\vec{y})$, the symmetric difference between $M(\vec{y})$ and $M_u(\vec{y})$ is an alternating path $(u, u_1, u_2, \ldots)$ in which the perturbed weights of edges are decreasing. Consequently, vertices $u_1, u_3, \ldots$ are matched earlier[12] in $M(\vec{y})$ than in $M_u(\vec{y})$.*

Following the same definition of victim, we define the gain sharing rules for edge-weighted graphs. For technical reasons, we set $h(y) = \frac{1}{10}(1 - g(y))$ and restrict ourselves to non-decreasing function g such that $g(y) \in [0.4, 0.6]$ for all $y \in [0, 1]$. As a consequence, for all $y \in [0, 1]$ we have

$$m(y) \stackrel{\text{def}}{=} \min\{g(y) - h(y), 1 - g(y)\} = \min\{1.1g(y) - 0.1, 1 - g(y)\} \geq 0.34 > 5 \cdot \max_x\{h(x)\}.$$

Gain Sharing. Consider the following two-step approach for gain sharing in matching $M(\vec{y})$:

- Whenever an edge (u, v) is added to the matching with u being active and v being passive, let $\alpha_u = g(y_u) \cdot w_{uv}$ and $\alpha_v = (1 - g(y_u)) \cdot w_{uv}$.
- For each active vertex z, if z has a victim v, decrease α_z and increase α_v by the same amount such that $\alpha_v \geq h(y_z) \cdot w_{uz}$ afterwards, where u is the vertex matched by z. More specifically, the amount of compensation is (where $[t]^+ \stackrel{\text{def}}{=} \max\{t, 0\}$)
  - $h(y_z) \cdot w_{uz}$ if v is unmatched;
  - at most $[h(y_z) \cdot w_{uz} - (g(y_v) - h(y_v)) \cdot w_{vx}]^+$ if v actively matches some vertex x;
  - $[h(y_z) \cdot w_{uz} - (1 - g(y_x)) \cdot w_{vx}]^+$ if v is passively matched by x.

Note that the above compensation step is consistent with the unweighted case: if the victim v is matched, then the amount of compensation is 0, since $m(y_v) > h(y_z)$; otherwise, it is $h(y_z)$.

Lower Bounds of Gains. Suppose that vertex u is matched with vertex v. By the gain sharing rules, if u is active, then its gain is at least $(g(y_u) - h(y_u))w_{uv}$; if it is passive, then its gain is $(1 - g(y_u))w_{uv}$. Thus the gain of a matched vertex u is lower bounded by $m(y_u)w_{uv}$. While the amount of compensation a victim v (of z) receives depends on the gain of v in the first step, our

[12]There is no explicit concept of time. However, since the edges are probed in descending order of their perturbed weights, a vertex being matched earlier means that the new edge has a larger perturbed weight than the old one.

compensation rule guarantees $\alpha_v \geq h(y_z) \cdot w_{zu}$ afterward, where u is the perfect partner of v and is matched by z in $M(\vec{y})$.

We follow the previous framework by fixing an arbitrary pair of perfect partners (u, v), and fixing the ranks of all vertices other than u, v arbitrarily. We derive lower bounds on $\alpha_u + \alpha_v$ for every $M(y_u, y_v)$, and show that the integration (over y_u and y_v) of the lower bound is at least 0.5014. For convenience, in the following, we assume $w_{uv} = 1$.

In the remaining part of this section, we say that u is matched *strictly earlier* than v in matching $M(\vec{y})$ if v remains unmatched after u is matched; we say that u is matched *earlier* than v if either u is strictly earlier than v, or u actively matches v.

Fact 5.1. *Suppose u is matched earlier than v in $M(y_u, y_v)$.*
1. *Increasing the rank of v does not change the matching status of u.*
2. *If u is active, $\alpha_u \geq g(y_u) - h(y_u)$; if u is passive, $\alpha_u \geq 1 - g(y_u)$.*

Proof. For the first statement, increasing the rank of v can not increase the perturbed weights of edges adjacent to v. Thus before u is matched, all probes have the same results before and after the increment of y_v, which means u is matched to the same neighbor.

For the second statement, suppose that u is matched with some vertex z (which may possibly be v). If u is active, then $(1 - g(y_u))w_{uz} \geq (1 - g(y_u))w_{uv}$ since edge (u, z) is probed no later than (u, v). Hence $\alpha_u \geq (g(y_u) - h(y_u))w_{uz} \geq g(y_u) - h(y_u)$. When u is passive, the perturbed weight of (u, z) equals $(1 - g(y_z))w_{uz}$, which is at least the perturbed weight of (u, v). Thus we have $\alpha_u \geq (1 - g(y_z))w_{uz} \geq 1 - g(\min\{y_u, y_v\}) \geq 1 - g(y_u)$. □

Lemma 5.2. *Suppose u is active and matched earlier than v in $M(y_u, y_v)$. If v is matched with some x such that $w_{vx} \geq \frac{1}{2}$ in $M_u(y_u, y_v)$, then we have $\alpha_u + \alpha_v \geq g(y_u)$ in $M(y_u, y_v)$.*

Proof. By Fact 5.1, we have $\alpha_u \geq g(y_u) - h(y_u)$. If v remains matched with x in $M(y_u, y_v)$, then the lemma holds since $\alpha_u + \alpha_v \geq g(y_u) - h(y_u) + \frac{1}{2}m(y_v) \geq g(y_u)$ (recall that $\min_y\{m(y)\} \geq 5 \cdot \max_x\{h(x)\}$.). If u does not have a victim or only needs to send 0 compensation to its victim, then we are also done.

Otherwise, by Lemma 5.1, the symmetric difference between $M(y_u, y_v)$ and $M_u(y_u, y_v)$ is an alternating path starting from u that contains (v, x). Let z be matched by u in $M(y_u, y_v)$ and z^* be the victim of u. Since the edge z^* matches in $M(y_u, y_v)$ appears no later than (v, x) in the alternating path, by Lemma 5.1, the perturbed weight of this edge is at least $(1 - g(\min\{y_v, y_x\}))\frac{1}{2} \geq 0.2$ (recall that $g(\cdot) \in [0.4, 0.6]$). Hence the gain of z^* in $M(y_u, y_v)$ before the compensation step is at least $m(y_{z^*}) \cdot \frac{0.2}{1-g(0)} = m(y_{z^*}) \cdot \frac{1}{3}$. Consequently the gain of u after the compensation step is $\alpha_u \geq (g(y_u) - h(y_u))w_{uz} + m(y_{z^*}) \cdot \frac{1}{3} \geq g(y_u)$. □

We regard the above lemma as a weighted generalization of Lemma 4.2, which states that when u is active, it need not send compensation if v matches an edge that is not too bad when u is removed.

Now we are ready to derive lower bounds on $\alpha_u + \alpha_v$ for every $M(y_u, y_v)$. Similar to before, we divide our analysis into two cases depending on whether $(u, v) \in M(1, 1)$.

5.1 Symmetric Case: $(u, v) \in M(1, 1)$

Since u, v are matched to each other in $M(1, 1)$, we define θ to be the transition rank such that u matches v in $M(y_u, 1)$ when $y_u > \theta$; matches a vertex other than v in $M(y_u, 1)$ when $y_u \in (0, \theta)$. We define λ analogously for v. Assume w.l.o.g. that $\theta \geq \lambda$.

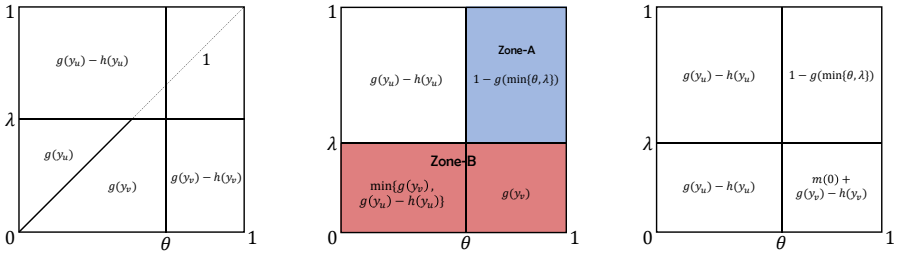
First, we show that for $y_u > \theta$ and $y_v > \lambda$, u, v are matched together in $M(y_u, y_v)$. Suppose otherwise, and assume u is matched strictly earlier. By Fact 5.1, if we increase y_v to 1, the matching status of u should not be affected. Hence u is not matched with v in $M(y_u, 1)$, which contradicts

the definition of θ (recall that $y_u > \theta$). The same argument implies that v is not matched strictly earlier. Hence $\alpha_u + \alpha_v = 1$ when $y_u > \theta$ and $y_v > \lambda$ (recall that u, v are perfect partners, and hence they do not need to send any compensation).

Next, we consider the case when $y_u < \theta$ and $y_v > \lambda$. We have (1) v is not matched strictly earlier than u; and (2) u is not passively matched strictly earlier than v. Because in either case, increasing y_u does not change the matching status of v, which means that v is not matched with u in $M(1, y_v)$, contradicting the definition of λ. Hence when $y_u < \theta$ and $y_v > \lambda$, either u is passively matched by v, or u is active and matched earlier than v. By Lemma 5.1 we have $\alpha_u + \alpha_v \geq g(y_u) - h(y_u)$[13]. Symmetrically, we have $\alpha_u + \alpha_v \geq g(y_v) - h(y_v)$ when $y_u > \theta$ and $y_v < \lambda$.

Finally, we consider the case when $y_u < \theta$ and $y_v < \lambda$. We show that in this case we must have $\alpha_u + \alpha_v \geq \min\{g(y_u), g(y_v)\}$. Suppose u is matched earlier than v. Then u must be active, as otherwise u is also passive in $M(y_u, 1)$. By definition of λ, we know that in $M_u(y_u, y_v)$, the edge v matches has weight at least $w_{uv} = 1$. Applying Lemma 5.2, we have $\alpha_u + \alpha_v \geq g(y_u)$. Similarly, when v is matched earlier than u, we have $\alpha_u + \alpha_v \geq g(y_v)$. Therefore, $\alpha_u + \alpha_v \geq \min\{g(y_u), g(y_v)\}$. Given that g is non-decreasing, we have $\alpha_u + \alpha_v \geq g(y_u)$ when $y_u < y_v$ and $\alpha_u + \alpha_v \geq g(y_v)$ when $y_v < y_u$. In summary, we have (refer to Figure 6a)

$$\begin{aligned}\mathop{\mathbf{E}}_{y_u, y_v}[\alpha_u + \alpha_v] \geq &(1-\lambda)\int_0^\theta \Big(g(y_u) - h(y_u)\Big)dy_u + (1-\theta)\int_0^\lambda \Big(g(y_v) - h(y_v)\Big)dy_v \\ &+ \int_0^\lambda (\lambda - y_u)g(y_u)dy_u + \int_0^\lambda (\theta - y_v)g(y_v)dy_v + (1-\theta)(1-\lambda).\end{aligned} \tag{8}$$



(a) $(u, v) \in M(1, 1)$. (b) Basic lower bounds (c) u strictly earlier in $M_z(1, 1)$

Fig. 6. Lower bounds on $\alpha_u + \alpha_v$.

5.2 Asymmetric Case: $(u, v) \notin M(1, 1)$

W.l.o.g., suppose that u is matched strictly earlier than v in $M(1, 1)$ by z. Let θ be the transition rank such that u is active in $M(y_u, 1)$ when $y_u < \theta$ and passive when $y_u > \theta$. Note that when $y_u > \theta$, u is always passively matched by z in $M(y_u, 1)$. Let λ be the transition rank such that v is matched earlier than u in $M(1, y_v)$ when $y_v < \lambda$ and later than u when $y_v > \lambda$. Notably, we have $\lambda = \theta = y_z$ in the unweighted case, while all three parameters might differ for weighted graphs. For example, suppose u has another neighbor x and θ is the critical rank such that the perturbed weight of edge (u, x) beats the perturbed weight of the edge x is matched within $M(1, 1)$.

First of all, we know that u is matched by z when $y_u > \theta$ and $y_v > \lambda$, which means that $\alpha_u = (1 - g(y_z))w_{uz}$. Since edge (u, z) is probed earlier than (u, v) when $y_u > \theta$, we have $(1 - g(y_z))w_{uz} \geq$

[13]It is possible that u is active and matched strictly earlier than v even when $y_u > y_v$. This is a key difference between the weighted and unweighted case: a smaller rank does not necessarily imply earlier decision time.

$(1-g(\theta))w_{uv}$. Similarly $(1-g(y_z))w_{uz} \geq (1-g(\lambda))w_{uv}$. Thus we have $\alpha_u \geq 1-g(\min\{\theta,\lambda\})$ when $y_u > \theta$ and $y_v > \lambda$.

When $y_u < \theta$ and $y_v > \lambda$, we first show that u must be active in $M(y_u, y_v)$. Suppose u is passive, then the matching $M(y_u, y_v)$ should be the same as $M(1, y_v)$, in which v is matched later than u (by definition of λ). Therefore increasing y_v in matching $M(y_u, y_v)$ will not change the matching status of u, which implies that u is also passive in $M(y_u, 1)$, violating the definition of θ. Moreover, we have that u is matched earlier than v in $M(y_u, y_v)$, as otherwise v would be matched strictly earlier than u in $M(1, y_v)$, which again violates the definition of λ. Then by Fact 5.1, we have $\alpha_u \geq g(y_u) - h(y_u)$. Similarly, when $y_u > \theta$ and $y_v < \lambda$, v must be active and matched earlier than u. Observe that u is matched by z in $M_v(y_u, y_v)$ and $w_{uz} \geq \frac{1-g(1)}{1-g(y_z)} > \frac{1}{2}$. Applying Lemma 5.2, we have $\alpha_u + \alpha_v \geq g(y_v)$.

Finally, when $y_u < \theta$ and $y_v < \lambda$, the vertex matched earlier is active. Moreover, since u is matched strictly earlier than v in $M(1,1)$, the edge u matches in every $M_v(y_u, 1)$ has weight at least $w_{uz} \geq \frac{1}{2}$. Hence by Lemma 5.2, for all $y_u < \theta$ and $y_v < \lambda$, if v is matched earlier then we have $\alpha_u + \alpha_v \geq g(y_v)$. By Fact 5.1, if u is matched earlier we have $\alpha_u \geq g(y_u) - h(y_u)$. Thus, $\alpha_u + \alpha_v \geq \min\{g(y_u) - h(y_u), g(y_v)\}$. Note that if we can show that the edge v matches in every $M_u(1, y_v)$, where $y_v < \lambda$, has weight at least $\frac{1}{2}$, then by applying Lemma 5.2 we can improve the lower bound to $\alpha_u + \alpha_v \geq \min\{g(y_u), g(y_v)\}$.

In summary, we have the following lower bounds that serve as our basic gains (refer to Fig. 6b).

(L1) $\alpha_u \geq 1 - g(\min\{\theta,\lambda\})$ when $y_u > \theta$ and $y_v > \lambda$;
(L2) $\alpha_u \geq g(y_u) - h(y_u)$ when $y_u < \theta$ and $y_v > \lambda$;
(L3) $\alpha_u + \alpha_v \geq g(y_v)$ when $y_u > \theta$ and $y_v < \lambda$;
(L4) $\alpha_u + \alpha_v \geq \min\{g(y_u) - h(y_u), g(y_v)\}$ when $y_u < \theta$ and $y_v < \lambda$.

Extra Gains. Similar to our analysis for the unweighted case, we define Zone-A to be the matchings $M(y_u, y_v)$ when $y_u > \theta$ and $y_v > \lambda$ (where lower bound (L1) is applied), and Zone-B to be the matchings $M(y_u, y_v)$ when $y_v < \lambda$ (where lower bounds (L3) (L4) are applied). In the following, we show that better lower bounds can be obtained for at least one of these bounds. Roughly speaking, (like the unweighted case) if only one of u, v is matched in Zone-B, then we should be able to recover some extra gain in Zone-A, e.g., the compensation received by v. We continue our analysis according to the matching status of u and v in $M_z(1,1)$, i.e., when z is removed from the graph.

5.2.1 Case 1: $(u, v) \notin M_z(1,1)$. In this case, at least one of u, v is passively matched by another vertex in $M_z(1,1)$. We first consider the case when u is matched strictly earlier. In such case, we show that u has gain at least $m(y_u)$ in Zone-B, as it has a "backup" neighbor other than z, v. The following lemma is a weighted version of Lemma 4.3, and the proof is similar. We formalize it in a general way so that we can also apply it in later analysis.

Lemma 5.3. *When $y_u > \theta$ and $y_v < \lambda$, if u is matched strictly earlier than v in $M_z(y_u, 1)$, we have $\alpha_u \geq m(y_u)$ in $M(y_u, y_v)$.*

Proof. First, by definition u is passively matched by z in $M_v(y_u, y_v)$. Now consider $M(y_u, y_v)$.

If the insertion of v does not change the matching status of u, i.e., u remains passively matched with z, then we have $\alpha_u \geq 1 - g(\min\{\theta,\lambda\}) \geq m(y_u)$.

If u is matched even earlier after the insertion, then the edge (u, x) vertex u matches has perturbed weight larger than $1 - g(\min\{\theta,\lambda\})$. Hence if u is passive, we have $\alpha_u \geq 1 - g(\min\{\theta,\lambda\}) \geq m(y_u)$; if u is active, we have $\alpha_u \geq (g(y_u) - h(y_u))\frac{1-g(\min\{\theta,\lambda\})}{1-g(y_u)} \geq g(y_u) - h(y_u) \geq m(y_u)$.

Otherwise, u appears in the alternating path triggered by the insertion of v as a vertex v_{2i}, which is matched later after the insertion (refer to Lemma 5.1). Note that in this case, the vertex v_{2i-1} right before $v_{2i} = u$ must be z. Hence the matching status of u is not changed if we remove both v

and z simultaneously in $M(y_u, y_v)$ (following the same argument as in the proof of Lemma 4.3). On the other hand, by Fact 5.1, we have $\alpha_u \geq m(y_u)$ in $M_z(y_u, 1)$. Moreover, the matching status of u is not changed if we further remove v in $M_z(y_u, 1)$. Thus we have $\alpha_u \geq m(y_u)$ in $M(y_u, y_v)$. □

Note that if u is matched strictly earlier than v in $M_z(1, 1)$, then by the above lemma, for all $y_v < \lambda$, the gain of u in $M(1, y_v)$ is at least $m(1)$. Now consider decreasing the rank of u. By Lemma 2.3, when u is passive, the gain of u does not change until it becomes active, which gives $\alpha_u \geq 1 - g(y_u)$; when u is active, decreasing u's rank would not decrease the edge weight it matches, and hence $\alpha_u \geq g(y_u) - h(y_u)$. For ease of analysis, we choose function g such that m achieves its minimum at $m(0)$. To sum up, we have $\alpha_u \geq m(0)$ in $M(y_u, y_v)$ for all $y_v < \lambda$.

Therefore, we improve (L3) to $\alpha_u + \alpha_v \geq m(0) + g(y_v) - h(y_v)$ and (L4) to $\alpha_u + \alpha_v \geq \min\{g(y_u) - h(y_u), m(0) + g(y_v) - h(y_v)\}$. Note that by restricting $g(y) \in [0.4, 0.6]$, $m(0) + g(y_v) - h(y_v)$ is larger than $g(y_u) - h(y_u)$ for all y_u, y_v. Consequently, we have the following (refer to Figure 6c).

$$\begin{aligned} \mathop{\mathbf{E}}_{y_u, y_v}[\alpha_u + \alpha_v] \geq \int_0^\theta \Big(g(y_u) - h(y_u)\Big) dy_u + (1-\theta) \int_0^\lambda \Big(g(y_v) - h(y_v)\Big) dy_v \\ + (1-\theta)(1-\lambda)(1 - g(\min\{\lambda, \theta\})) + (1-\theta)\lambda \cdot m(0). \end{aligned} \tag{9}$$

It remains to consider the case when v is matched strictly earlier than u in $M_z(1, 1)$. As we will show later, this case can actually be regarded as a "better" case of $(u, v) \in M_z(1, 1)$, and thus we defer its analysis to the next subsection (see Remark 5.1).

5.2.2 Case 2: $(u, v) \in M_z(1, 1)$. Finally, we consider the case when u, v match each other in $M_z(1, 1)$. By definition, v is the victim of z in $M(1, 1)$, and hence

$$\alpha_v \geq h(y_z) w_{uz} = \tfrac{1}{10}(1 - g(y_z)) w_{uz} \geq \tfrac{1}{10}(1 - g(\min\{\theta, \lambda\})) = h(\min\{\theta, \lambda\}).$$

We remark that this is the major reason for restricting $h = \frac{1}{10}(1 - g)$, as otherwise we do not have an immediate connection between $h(y_z) w_{uz}$ and $h(\min\{\theta, \lambda\})$. Next, we show that v receives this compensation in (part of) Zone-A.

Lemma 5.4 (Compensation). *For all $y_u > \theta$, if u matches v in $M_z(y_u, 1)$, then for all $y_v > \lambda$ we have $\alpha_v \geq h(\min\{\theta, \lambda\})$ in $M(y_u, y_v)$.*

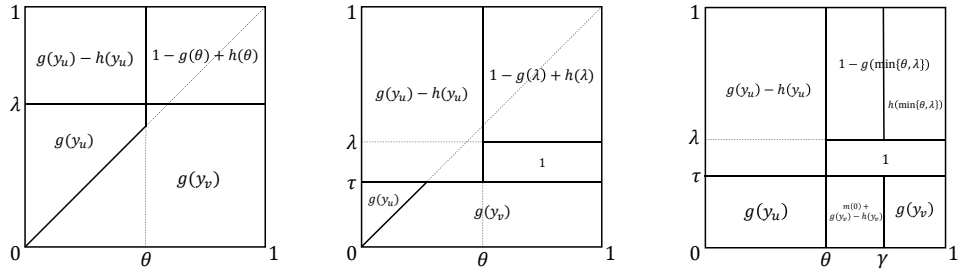
Proof. Consider $M_z(y_u, y_v)$. If u, v match each other, then v is the victim of z in $M(y_u, y_v)$, and the lemma holds. Otherwise, we know that v must be matched strictly earlier than u: if u is matched strictly earlier, then it will also be matched strictly earlier than v in $M_z(y_u, 1)$, which contradicts the assumption that u matches v in $M_z(y_u, 1)$.

Thus we have $\alpha_v \geq m(y_v)$ in $M_z(y_u, 1)$. Moreover, if we further remove u in $M_z(y_u, 1)$, the matching status of v is not changed. Since z matches u in $M(y_u, y_v)$, removing both z and u does not change the matching status of v, which means that $\alpha_v \geq m(y_v) > h(\min\{\theta, \lambda\})$ in $M(y_u, y_v)$. □

The above lemma indicates that it is crucial to determine whether u matches v in $M_z(y_u, 1)$ when $y_u > \theta$. Recall that $(u, v) \in M_z(1, 1)$. We define $\gamma \in [\theta, 1]$ to be the transition rank such that u matches v in $M_z(y_u, 1)$ when $y_u \in (\gamma, 1)$ and matches other vertex when $y_u \in (\theta, \gamma)$.

We first consider the case that u matches v in $M_z(y_u, 1)$ for all $y_u > \theta$. In this case $\gamma = \theta$, and thus by Lemma 5.4, we have $\alpha_u + \alpha_v \geq 1 - g(\min\{\theta, \lambda\}) + h(\min\{\theta, \lambda\})$ in Zone-A. The gain in Zone-B is more complicated. In particular, we need to consider whether u, v match each other in

$M(1, y_v)$ when $y_v < \lambda$. Let $\tau \in [0, \lambda]$ be the transition rank such that v matches u in $M(1, y_v)$ when $y_v \in (\tau, \lambda)$ and matches strictly earlier than u when $y_v \in (0, \tau)$.



(a) $\gamma = \theta$ and $\lambda > \theta$ (b) $\gamma = \theta$ and $\lambda \le \theta$ (c) $\gamma \in (\theta, 1)$

Fig. 7. $(u, v) \in M_z(1, 1)$.

To proceed, we compare θ and λ and consider the following two cases.

When $\lambda > \theta$. First, observe that when $\lambda > \theta$, we must have $\tau = \lambda$. In other words, v never matches u in $M(1, y_v)$ when $y_v < \lambda$. The reason is, when $y_v = \lambda^-$, the perturbed weight of edge (u, v) is $1 - g(y_v) < 1 - g(\theta) < (1 - g(y_z))w_{zu}$. However, by definition of λ we know that v is matched earlier than z in $M(1, y_v)$ when $y_v = \lambda^-$. Thus, v actively matches an edge with a weight larger than 1. Consequently by Lemma 2.3, v does not match u in $M(1, y_v)$ for all $y_v < \lambda$.

In other words, v is matched in $M_u(1, y_v)$ with an edge of weight at least 1 for all $y_v < \lambda$. Thus we apply Lemma 5.2 and improve (L4) to $\alpha_u + \alpha_v \ge \min\{g(y_u), g(y_v)\}$. To sum up, we have the following lower bound (refer to Figure 7a).

$$\begin{aligned}\mathop{\mathbf{E}}_{y_u, y_v}[\alpha_u + \alpha_v] \ge &(1-\theta)(1-\lambda)(1 - g(\theta) + h(\theta)) + (1-\lambda)\int_0^\theta \Big(g(y_u) - h(y_u)\Big)dy_u \\ &+ (1-\theta)\int_0^\lambda g(y_v)dy_v + \int_0^\theta (\theta + \lambda - 2y)g(y)dy.\end{aligned} \tag{10}$$

When $\lambda \le \theta$. In this case v matches u in $M(1, y_v)$ when $y_v \in (\tau, \lambda)$. Moreover, u remains matched by v in $M(y_u, y_v)$ when $y_u > \theta$ and $y_v \in (\tau, \lambda)$. When $y_v < \tau$, we have $\alpha_u + \alpha_v \ge \min\{g(y_u), g(y_v)\}$, by a similar argument as before. In summary, we have (refer to Figure 7b)

$$\begin{aligned}\mathop{\mathbf{E}}_{y_u, y_v}[\alpha_u + \alpha_v] \ge &(1-\theta)(1-\lambda)(1 - g(\lambda) + h(\lambda)) + (1-\tau)\int_0^\theta \Big(g(y_u) - h(y_u)\Big)dy_u \\ &+ (1-\theta)(\lambda - \tau) + (1-\theta)\int_0^\tau g(y_v)dy_v + \int_0^\tau (\theta + \tau - 2y)g(y)dy.\end{aligned} \tag{11}$$

The derivative over λ of the RHS of above is

$$(1-\theta) \cdot \left(g(\lambda) - h(\lambda) - (1-\lambda)(g'(\lambda) - h'(\lambda))\right).$$

Our choice of g, h guarantees that $g(y) - h(y)$ is always larger than $g'(y) - h'(y)$, and thus this derivative is positive. Therefore, the minimum is achieved when $\lambda = \tau$.

To sum up, the two lower bounds can be unified as follows.

$$\begin{aligned}\mathop{\mathbf{E}}_{y_u, y_v}[\alpha_u + \alpha_v] \ge &(1-\theta)(1-\lambda)(1 - g(\min\{\theta, \lambda\}) + h(\min\{\theta, \lambda\})) + (1-\theta)\int_0^\lambda g(y_v)dy_v \\ &+ (1-\lambda)\int_0^\theta \Big(g(y_u) - h(y_u)\Big)dy_u + \int_0^{\min\{\theta,\lambda\}} (\lambda + \theta - 2y)g(y)dy.\end{aligned} \tag{12}$$

Remark 5.1. *Now if we turn our attention back to the case when v is matched strictly earlier than u in $M_z(1,1)$, we can see that Equation* (12) *also serves as a lower bound. For Zone-A, the lower bound $1-g(\min\{\theta,\lambda\})+h(\min\{\theta,\lambda\})$ holds since v is matched strictly earlier than u in $M_z(1,y_v)$ when $y_v>\lambda$, which implies $\alpha_v \geq m(y_v)$. Exactly the same analysis on lower bounds for Zone-B can also be applied to this case.*

Finally, we consider the case when $\gamma > \theta$. By the definition of γ and Lemma 5.4, $\alpha_v \geq h(\min\{\theta,\lambda\})$ in the part of Zone-A when $y_u > \gamma$. When $y_u > \theta$ and $y_v \in (\tau,\lambda)$, u,v match each other and $\alpha_u+\alpha_v=1$. It remains to give lower bounds when $y_v<\tau$.

Since v is matched strictly earlier than u in $M(1,y_v)$, Lemma 5.2 applies and we have $\alpha_u+\alpha_v \geq \min\{g(y_u),g(y_v)\}$ for all $y_u\in(0,1)$ and $y_v<\tau$. Furthermore, since $\gamma>\theta$, we can actually improve this bound further. First, by Lemma 5.3, we have $\alpha_u \geq m(y_u) \geq m(0)$ when $y_u\in(\theta,\gamma)$ and $y_v<\tau$. Second, when $y_u<\theta$ and $y_v<\tau$, we can improve (L3) to

$$\alpha_u+\alpha_v \geq \min\{g(y_u), g(y_v)-h(y_v)+m(y_u)\} = g(y_u).$$

In summary, we have the lower bounds as shown in Figure 7c.

$$\begin{aligned}\mathbf{E}\left[\alpha_u+\alpha_v\right] \geq & (1-\theta)(1-\lambda)(1-g(\min\{\theta,\lambda\})) + (1-\gamma)(1-\lambda)h(\min\{\theta,\lambda\}) \\ & + \int_0^\theta \Big(g(y_u)-(1-\tau)h(y_u)\Big)dy_u + \int_0^\tau \Big(g(y_v)-(\gamma-\theta)h(y_v)\Big)dy_v \\ & + (1-\theta)(\lambda-\tau)+(\gamma-\theta)\tau\cdot m(0).\end{aligned} \tag{13}$$

It can be verified that for every fixed τ, the minimum of the above lower bound is achieved when $\lambda=\tau$ (given that $1-g(y)+h(y)$ is always smaller than 1). Moreover, for every fixed $\lambda=\tau$, the derivative over γ is a constant. Hence the minimum must be achieved when $\gamma\in\{\theta,1\}$. It can be further verified that when $\gamma=1$, Equation (9) serves as an lower bound for Equation (13); when $\gamma=\theta$, Equation (12) serves as an lower bound.

5.3 Lower Bounding the Approximation Ratio

Unlike the unweighted case, the performance of Perturbed Greedy on edge-weighted graphs depends on the design of the function g. To this end, we explicitly construct one and analytically prove the approximation ratio, rather than running a factor revealing LP. In particular, we construct a function g that satisfies all pre-specified constraints such that the lower bounds (8) (9) (12) are at least 0.5014, which completes the proof of Theorem 1.4. We remark that the piece-wise linear function is just an artifact of the proof, and is not optimal for maximizing the approximation ratio.

In the following, we fix function $g(y)=\begin{cases}0.365y+0.48926, & y\leq 0.13\\ 0.067y+0.528, & y\in(0.13,0.4)\\ 0.5548 & y\geq 0.4.\end{cases}$

First, it can be verified that the constraints we put on g are satisfied: for every $y\in(0,1)$, we have $g(y)\in(0.4,0.6)$ and $g(y)-h(y)\geq g'(y)-h'(y)$. It can also be shown that for the function g we fix, we have $\min_y\{m(y)\}=m(0)=g(0)-h(0)=0.438186$.

5.3.1 Equation (8). We prove that the RHS of (8) (shown as follows) is at least 0.5014 (over $\theta\geq\lambda$).

$$\begin{aligned}&(1-\theta)(1-\lambda)+(1-\lambda)\int_0^\theta \Big(g(y_u)-h(y_u)\Big)dy_u + \int_0^\lambda (\lambda-y_u)g(y_u)dy_u \\ &+(1-\theta)\int_0^\lambda \Big(g(y_v)-h(y_v)\Big)dy_v + \int_0^\lambda(\theta-y_v)g(y_v)dy_v.\end{aligned}$$

First, if we take the derivative over θ, we have

$$(1-\lambda)(g(\theta)-h(\theta)-1)+\int_0^{\lambda} h(y)dy,$$

which is negative (for any θ) when $\lambda \le 0.9$.

Thus for $\lambda \le 0.9$, the minimum is achieved when $\theta = 1$:

$$(1-\lambda)\int_0^1 \Big(g(y)-h(y)\Big)dy+\int_0^{\lambda}(\lambda-y)g(y)dy+\int_0^{\lambda}(1-y)g(y)dy.$$

By taking the derivative over λ, we have

$$\int_0^{\lambda} g(y)dy+(1-\lambda)g(\lambda)-\int_0^1 \Big(g(y)-h(y)\Big)dy.$$

Since this derivative is non-decreasing, the minimum is achieved when $\lambda = \lambda^* = 0.0344402$ (solution for $\int_0^{\lambda} g(y)dy+(1-\lambda)g(\lambda) = \int_0^1 g(y)-h(y)dy$), and the value is at least 0.5014.

For $\lambda > 0.9$, we lower bound $(1-\theta)(1-\lambda)$ by $(1-\lambda)\int_{\theta}^1 \Big(g(y_u)-h(y_u)\Big)dy_u$, then the RHS of (8) becomes

$$\begin{aligned}&(1-\lambda)\int_0^1 \Big(g(y_u)-h(y_u)\Big)dy_u+\int_0^{\lambda}(\lambda-y_u)g(y_u)dy_u\\ &+(1-\theta)\int_0^{\lambda}\Big(g(y_v)-h(y_v)\Big)dy_v+\int_0^{\lambda}(\theta-y_v)g(y_v)dy_v,\end{aligned}$$

which attains its minimum when $\theta = \lambda$: (the inequality holds since h is non-increasing)

$$\begin{aligned}&(1-\lambda)\int_0^1 \Big(g(y)-h(y)\Big)dy+(1-\lambda)\int_0^{\lambda}\Big(g(y)-h(y)\Big)dy+2\int_0^{\lambda}(\lambda-y)g(y)dy\\ \ge&(1-\lambda)\int_0^1 \Big(g(y)-h(y)\Big)dy+(1+\lambda)\int_0^{\lambda} g(y)dy-2\int_0^{\lambda} y\cdot g(y)dy-(1-\lambda)\lambda\cdot h(0).\end{aligned}$$

Observe that the derivative (for $\lambda \ge 0.9$)

$$\int_0^{\lambda} g(y)dy+(1-\lambda)g(\lambda)-\int_0^1 \Big(g(y)-h(y)\Big)dy-(1-2\lambda)h(0) > 0.$$

Thus the minimum is achieved when $\lambda = 0.9$, which is at least 0.53.

5.3.2 Equation (9). We prove the RHS of (9) (shown as follows) is at least 0.5016 (over all θ and λ):

$$\begin{aligned}&\int_0^{\theta}\Big(g(y_u)-h(y_u)\Big)dy_u+(1-\theta)\int_0^{\lambda}\Big(g(y_v)-h(y_v)\Big)dy_v\\ &+(1-\theta)(1-\lambda)(1-g(\min\{\lambda,\theta\}))+(1-\theta)\lambda\cdot m(0).\end{aligned}$$

First, observe that if $\lambda > \theta$, then the derivative over λ is non-negative, which means that the minimum in this case is achieved when $\lambda = \theta$. Thus it suffices to consider the case when $\lambda \le \theta$. For this case, the derivative over θ is given by

$$g(\theta)-h(\theta)-(1-\lambda)(1-g(\lambda))-\lambda\cdot m(0)-\int_0^{\lambda}\Big(g(y)-h(y)\Big)dy.$$

It can be verified (by taking another derivative over λ) that the maximum value of the above derivative is achieved when $\lambda = 0$:

$$g(\theta)-h(\theta)-(1-g(0)) \le g(1)-h(1)-(1-g(0)) = 0.51028-0.51074 < 0.$$

Thus the minimum of (9) is attained when $\theta = 1$:

$$\int_0^1 \Big(g(y) - h(y)\Big)dy \geq 0.5016.$$

5.3.3 Equation (12). We prove that the RHS of (12) (shown as follows) is at least 0.5014 (over all θ and λ):

$$(1-\theta)(1-\lambda)(1-g(\min\{\theta,\lambda\}) + h(\min\{\theta,\lambda\})) + (1-\theta)\int_0^\lambda g(y_v)dy_v$$
$$+ (1-\lambda)\int_0^\theta \Big(g(y_u) - h(y_u)\Big)dy_u + \int_0^{\min\{\theta,\lambda\}} (\theta + \lambda - 2y)g(y)dy.$$

First, observe that except for a $(1-\lambda)\int_0^\theta h(y)dy$ term, the lower bound is symmetric for θ and λ. Moreover, if $\theta < \lambda$, then $(1-\theta)\int_0^\lambda h(y)dy > (1-\lambda)\int_0^\theta h(y)dy$, which means that by swapping the values of θ and λ, the lower bound decreases. Hence it suffices to consider the case when $\theta \geq \lambda$.

When $\theta \geq \lambda$, the derivative over θ is given by

$$(1-\lambda)\Big((g(\lambda) - h(\lambda)) + (g(\theta) - h(\theta)) - 1\Big) = \frac{11}{10}(1-\lambda)\Big((g(\lambda) + g(\theta)) - \frac{12}{11}\Big).$$

Let $\lambda_0 \approx 0.12835$ be the solution for $g(\lambda) + g(1) = \frac{12}{11}$. Since $g(y)$ is non-decreasing, for $\lambda < \lambda_0$, we have $g(\lambda) + g(1) < \frac{12}{11}$, which implies that the above derivative is negative. Hence the minimum is achieved when $\theta = 1$:

$$(1-\lambda)\int_0^1 \Big(g(y) - h(y)\Big)dy + \int_0^\lambda (1 + \lambda - 2y)g(y)dy.$$

Then following the same argument as in Section 5.3.1, the minimum value is at least 0.5014.

For $\lambda \geq \lambda_0$, the minimum is achieved when $\theta = \theta^*$, where $g(\lambda) + g(\theta^*) = \frac{12}{11}$. Let $\lambda^* = 0.260516$ be the solution for $g(\lambda) = \frac{6}{11}$. Note that we must have $\lambda \leq \lambda^* \leq \theta^* \leq 0.4$.

For $\lambda \in (\lambda_0, 0.13]$, by definition of function g, $g(\lambda) + g(\theta^*) = \frac{12}{11}$ is equivalent to

$$0.067 \cdot \theta^* + 0.528 + 0.365 \cdot \lambda + 0.48926 = \frac{12}{11}.$$

Plugging in θ^* and using $g(\lambda) = 0.365 \cdot \lambda + 0.48926$, we can explicitly express the lower bound as a cubic function of λ, which achieves its minimum value 0.5026 when $\lambda = 0.13$.

For $\lambda \in (0.13, \lambda^*]$, we have $\theta^* = 2\lambda^* - \lambda$. Plugging in θ^* and using $g(\lambda) = 0.067 \cdot \lambda + 0.528$, we can explicitly express the lower bound as another cubic function of λ, which achieves its minimum value 0.50235 when $\lambda \approx 0.204$.

Thus for all $\lambda, \theta \in [0, 1]$, the lower bound is at least 0.5014.

6 WEIGHTED BIPARTITE GRAPHS

In this section, we prove Theorem 1.5. We remark that our algorithm achieves the same approximation ratio $(1 - 1/e)$ as the algorithm by Gamlath et al. [13], while the two algorithms exploit quite different structures of the problem. The existence probabilities of edges are crucial to [13] in that they can estimate the probability of each edge appearing in the optimal matching in advance. Given that the analysis (of showing $(1 - 1/e)$-approximation) of both algorithms are tight, it remains an interesting open question to see how the two ideas can be combined and how better algorithms can be designed for the stochastic bipartite matching problem.

Let the given bipartite graph be $G = (L \cup R, E)$, where L and R denote the left hand side and right hand side vertex set, respectively. Next, we describe the One-Sided Perturbed Greedy algorithm:

One-Sided Perturbed Greedy. Fix non-decreasing function $g(y) := e^{y-1}$. Each vertex $u \in L$ independently draws a rank $y_u \in [0,1]$ uniformly at random. For each pair of vertices (u,v) where $u \in L$ and $v \in R$, let $(1-g(y_u))w_{uv}$ be the perturbed weight of pair (u,v). We probe all pairs of vertices in descending order of their perturbed weights.

Recall that in the Perturbed Greedy algorithm, vertices from both sides have ranks, and the perturbed weight of edge (u,v) is given by $(1-g(\min\{y_u,y_v\})) \cdot w_{uv}$. We can equivalently think of the One-Sided Perturbed Greedy algorithm as the variant of Perturbed Greedy in which the rank y_v of each vertex $v \in R$ is fixed to be $y_v = 1$.

We follow similar notations as before, and use $\vec{y}$ to denote the rank vector of all vertices in L and $M(\vec{y})$ to denote the corresponding matching produced by our algorithm with the rank vector $\vec{y}$.

Gain Sharing. For each matched edge (u,v), where $u \in L$ and $v \in R$, let $\alpha_u = g(y_u) \cdot w_{uv}$ and $\alpha_v = (1-g(y_u)) \cdot w_{uv}$.

It is obvious to see that the first condition of Lemma 2.1 is satisfied by our gain sharing rule, since for any matched edge (u,v), we have $\alpha_u + \alpha_v = w_{uv}$. Next, we fix a pair of neighbors (u,v) (which are not necessarily perfect partners) and derive a lower bound of the expected gain of $\alpha_u + \alpha_v$. We start with a basic monotonicity property of the ranks of vertices, e.g., higher rank leads to a "better" matching result for every fixed vertex. The proof of the following lemma is almost identical to that of Lemma 2.3. However, since the two algorithms are different, we present proof here for completeness.

Lemma 6.1 (Monotonicity). *Consider any matching $M(\vec{y})$ and any vertex $u \in L$, if we fix the ranks of all vertices expect u, the weight of the edge u matches is non-increasing w.r.t. $y_u \in [0,1]$.*

Proof. Suppose u is matched with v when $y_u = y$. If we fix the ranks of all other vertices, and decrease y_u to be some $y' < y$, the perturbed weight of all edges adjacent to u will decrease. Hence these edges have a more prior order when we probe edges. More specifically, edge (u,v) will be probed at least as early as when $y_u = y$. If u is unmatched when (u,v) is probed, then u and v will match each other and the matching remains the same; otherwise u is matched to some z such that

$$(1-g(y')) \cdot w_{uz} \geq (1-g(y')) \cdot w_{uv},$$

which implies $w_{uz} \geq w_{uv}$. Hence the lemma follows. □

From now on, we fix the ranks of all vertices in L other than u. By Lemma 6.1, there exists a *marginal rank* θ, such that the weight of edge u matches is at least w_{uv} if and only if $y_u \in [0,\theta)$. This implies a basic bound for the gain of u when $y_u \in [0,\theta)$, and the gain of v when $y_u \in [\theta,1]$.

Lemma 6.2. *When $y_u \in [0,\theta)$, $\alpha_u \geq g(y_u) \cdot w_{uv}$, and when $y_u \in [\theta,1]$, $\alpha_v \geq (1-g(\theta)) \cdot w_{uv}$.*

Proof. The first bound is implied from the definition of θ and our gain sharing method. For the second one, consider the case when $y_u = \theta$. By definition, u matches an edge with a weight smaller than w_{uv}, which means that when we probe edge (u,v), which has perturbed weight $(1-g(\theta)) \cdot w_{uv}$, v is already matched, and u is not matched. Thus, the perturbed weight of the edge v matches is larger than $(1-g(\theta)) \cdot w_{uv}$, which implies $\alpha_v \geq (1-g(\theta)) \cdot w_{uv}$.

Since u is not matched when we probe edge (u,v), if we further increase y_u, then u remains unmatched when we probe edge (u,v) (by Lemma 6.1). In other words, all edges adjacent to u that are probed before (u,v) are unsuccessful. Hence increasing y_u to any value in $(\theta,1]$ does not change the matching status of v ,which implies $\alpha_v \geq (1-g(\theta)) \cdot w_{uv}$ for all $y_u \in [\theta,1]$. □

The next lemma characterizes the matching status of v when $y_u < \theta$ and is the only part of the proof that crucially uses the bipartiteness of the graph. Indeed, the lemma is implied by Corollary 2.7

by fixing the ranks of all $v \in R$ to be 1 in the algorithm (in which case the two algorithms One-Sided Perturbed Greedy and Perturbed Greedy are equivalent).

Lemma 6.3 (Extra Gain for Bipartite). *When $y_u \in [0, \theta)$, $\alpha_v \geq (1 - g(\theta)) \cdot w_{uv}$.*

Combining Lemma 6.2 and Lemma 6.3, we have

$$\mathop{\mathbf{E}}_{\vec{y}}[\alpha_u + \alpha_v] \geq w_{uv} \cdot \left(\int_0^{\theta} g(y_u) dy_u + (1 - g(\theta)) \right) = w_{uv} \cdot (1 - \frac{1}{e}),$$

where the equality follows from $g(y) = e^{y-1}$. By Lemma 2.1, the approximation ratio of One-Sided Perturbed Greedy is at least $1 - 1/e$, which finishes the proof of Theorem 1.5.

7 CONCLUSION AND OPEN QUESTIONS

In this paper, we prove that RDO breaks the 0.5 barrier by using a random decision order and arbitrary preference orders. A natural question to ask is whether the random preferences help in MRG, i.e., whether MRG has a strictly larger approximation ratio than RDO in the worst case. We slightly believe so and would like to see techniques extending the current gain sharing framework to analyze random preference orders. Another interesting open question is to give a tight approximation ratio analysis for RDO on unweighted bipartite graphs. We have shown an upper bound of 0.646 and a lower bound of 0.639, which are already very close.

We also propose the first algorithm that achieves approximation ratio strictly greater than 0.5 for the edge-weighted oblivious matching problem. Additionally, we show that a modified version of Perturbed Greedy that uses only the ranks on one side of the graph achieves an approximation ratio of $(1 - 1/e)$. We conjecture that Perturbed Greedy (with an appropriate choice of g) has an approximation ratio strictly greater than $(1 - 1/e)$ on bipartite graphs, and we leave this as an open problem.

REFERENCES

[1] Jonathan Aronson, Martin Dyer, Alan Frieze, and Stephen Suen. 1995. Randomized greedy matching. II. *Random Struct. Algorithms* 6, 1 (Jan. 1995), 55–73. https://doi.org/10.1002/rsa.3240060107

[2] Sepehr Assadi, Sanjeev Khanna, and Yang Li. 2017. The Stochastic Matching Problem: Beating Half with a Non-Adaptive Algorithm. In *EC*. ACM, 99–116.

[3] T.-H. Hubert Chan, Fei Chen, and Xiaowei Wu. 2018. Analyzing Node-Weighted Oblivious Matching Problem via Continuous LP with Jump Discontinuity. *ACM Trans. Algorithms* 14, 2 (2018), 12:1–12:25.

[4] T.-H. Hubert Chan, Fei Chen, Xiaowei Wu, and Zhichao Zhao. 2018. Ranking on Arbitrary Graphs: Rematch via Continuous Linear Programming. *SIAM J. Comput.* 47, 4 (2018), 1529–1546.

[5] Ning Chen, Nicole Immorlica, Anna R. Karlin, Mohammad Mahdian, and Atri Rudra. 2009. Approximating Matches Made in Heaven. In *ICALP (1) (Lecture Notes in Computer Science, Vol. 5555)*. Springer, 266–278.

[6] Kevin P. Costello, Prasad Tetali, and Pushkar Tripathi. 2012. Stochastic Matching with Commitment. In *ICALP (1) (Lecture Notes in Computer Science, Vol. 7391)*. Springer, 822–833.

[7] Mahsa Derakhshan and Alireza Farhadi. 2023. Beating (1 - 1/e)-Approximation for Weighted Stochastic Matching. In *SODA*. SIAM, 1931–1961.

[8] Nikhil R. Devanur, Kamal Jain, and Robert D. Kleinberg. 2013. Randomized Primal-Dual analysis of RANKING for Online BiPartite Matching. In *SODA*. SIAM, 101–107.

[9] Martin E. Dyer and Alan M. Frieze. 1991. Randomized Greedy Matching. *Random Struct. Algorithms* 2, 1 (1991), 29–46.

[10] Matthew Fahrbach, Zhiyi Huang, Runzhou Tao, and Morteza Zadimoghaddam. 2020. Edge-Weighted Online Bipartite Matching. In *FOCS*. IEEE, 412–423.

[11] Lance Fortnow and Salil P. Vadhan (Eds.). 2011. *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*. ACM.

[12] Hu Fu, Zhihao Gavin Tang, Hongxun Wu, Jinzhao Wu, and Qianfan Zhang. 2021. Random Order Vertex Arrival Contention Resolution Schemes for Matching, with Applications. In *ICALP (LIPIcs, Vol. 198)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 68:1–68:20.

[13] Buddhima Gamlath, Sagar Kale, and Ola Svensson. 2019. Beating Greedy for Stochastic Bipartite Matching. In *SODA*. SIAM, 2841–2854.
[14] G. Goel and P. Tripathi. 2012. Matching with Our Eyes Closed. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*. 718–727. https://doi.org/10.1109/FOCS.2012.19
[15] Zhiyi Huang. 2019. Understanding Zadimoghaddam's Edge-weighted Online Matching Algorithm: Weighted Case. *CoRR* abs/1910.03287 (2019).
[16] Zhiyi Huang, Ning Kang, Zhihao Gavin Tang, Xiaowei Wu, Yuhao Zhang, and Xue Zhu. 2018. How to match when all vertices arrive online. In *STOC*. ACM, 17–29.
[17] Zhiyi Huang, Ning Kang, Zhihao Gavin Tang, Xiaowei Wu, Yuhao Zhang, and Xue Zhu. 2020. Fully Online Matching. *J. ACM* 67, 3 (2020), 17:1–17:25.
[18] Zhiyi Huang, Binghui Peng, Zhihao Gavin Tang, Runzhou Tao, Xiaowei Wu, and Yuhao Zhang. 2019. Tight Competitive Ratios of Classic Matching Algorithms in the Fully Online Model. In *SODA*. SIAM, 2875–2886.
[19] Zhiyi Huang, Zhihao Gavin Tang, Xiaowei Wu, and Yuhao Zhang. 2018. Online Vertex-Weighted Bipartite Matching: Beating 1-1/e with Random Arrivals. In *ICALP (LIPIcs, Vol. 107)*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 79:1–79:14.
[20] Chinmay Karande, Aranyak Mehta, and Pushkar Tripathi. 2011. Online bipartite matching with unknown distributions, See [11], 587–596.
[21] Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. 1990. An Optimal Algorithm for On-line Bipartite Matching. In *STOC*, Harriet Ortiz (Ed.). ACM, 352–358.
[22] Jacob Magun. 1998. Greedy Matching Algorithms: An Experimental Study. *ACM Journal of Experimental Algorithmics* 3 (1998), 6. https://doi.org/10.1145/297096.297131
[23] Mohammad Mahdian and Qiqi Yan. 2011. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing LPs, See [11], 597–606.
[24] Matthias Poloczek and Mario Szegedy. 2012. Randomized Greedy Algorithms for the Maximum Matching Problem with New Analysis. In *FOCS*. IEEE Computer Society, 708–717. https://ieeexplore.ieee.org/xpl/conhome/6374356/proceeding
[25] Alvin E. Roth, Tayfun Sönmez, and M. Utku Ünver. 2005. Pairwise kidney exchange. *J. Economic Theory* 125, 2 (2005), 151–188.
[26] Sahil Singla. 2018. The Price of Information in Combinatorial Optimization. In *SODA*. SIAM, 2523–2532.
[27] Zhihao Gavin Tang, Xiaowei Wu, and Yuhao Zhang. 2020. A Simple 1-1/e Approximation for Oblivious Bipartite Matching. *CoRR* abs/2002.06037 (2020).
[28] Zhihao Gavin Tang, Xiaowei Wu, and Yuhao Zhang. 2020. Towards a better understanding of randomized greedy matching. In *STOC*. ACM, 1097–1110.
[29] Gottfried Tinhofer. 1984. A probabilistic analysis of some greedy cardinality matching algorithms. *Annals OR* 1, 3 (1984), 239–254. https://doi.org/10.1007/BF01874391

A FACTOR REVEALING LP

Recall that, to lower bound the approximation ratio, we need to define appropriate functions g and h such that the lower bounds we formulate on $\mathbf{E}[\alpha_u + \alpha_v]$ are at least some ratio $r > 0.5$ for all parameters θ, λ, τ and γ.

For instance, for the unweighted bipartite case[14], we formulated the following two lower bounds:

$$\begin{aligned} L^1(\theta,\tau) = &\int_0^\theta (1-y)g(y)dy + \int_0^\theta \min\{y,\tau\}m(y)dy + (1-\theta)(1-\theta+\tau)(1-g(\theta)) \\ &+ \int_0^\tau (1-y)g(y)dy + \int_0^\tau ym(y)dy + \frac{1}{2}(2-\tau-\theta)(\theta-\tau), \\ L^2(\theta,\tau) = &\int_0^\theta (1-y)g(y)dy + \int_0^\theta \min\{y,\tau\}m(y)dy + \frac{1}{2}(1-\theta)^2 \\ &+ \int_0^\tau (1-y)g(y)dy + \int_0^\tau ym(y)dy + \frac{1}{2}(1-\tau)^2. \end{aligned}$$

To prove Theorem 1.1, it remains to find function g such that

$$\min_{\theta,\tau}\{L^1(\theta,\tau), L^2(\theta,\tau)\} \geq 0.639.$$

This can be proved by solving the following continuous optimization problem and showing that the optimal solution is at least 0.639.

$$\begin{aligned} \text{maximize} \quad & r \\ \text{subject to} \quad & r \leq L^1(\theta,\tau), \quad \forall \theta,\tau \in [0,1] \\ & r \leq L^2(\theta,\tau), \quad \forall \theta,\tau \in [0,1] \\ & g(y) \in [0,1], \quad \forall y \in [0,1] \\ & g(y) \text{ is non-decreasing.} \end{aligned}$$

Since $L^1(\theta,\tau), L^2(\theta,\tau)$ are both linear in $g(y)$, we can discretize them as follows. Let n be the size of discretization. The larger n is, the more accurately we can solve the above continuous optimization problem. Let g be a step function such that $g(y) = g_i$ for all $y \in [\frac{i}{n}, \frac{i+1}{n})$. By doing so, the integrations in the equations L^1, L^2 can be represented by linear summations over g_i's.

Moreover, if we can formulate linear functions $D^1(i,j), D^2(i,j)$ in a way that $D^1(i,j) \leq L^1(\theta,\tau)$ and $D^2(i,j) \leq L^2(\theta,\tau)$ for all $\theta \in [\frac{i}{n}, \frac{i+1}{n})$ and $\tau \in [\frac{j}{n}, \frac{j+1}{n})$, then the optimal solution of the following finite LP provides a lower bound on the approximation ratio.

$$\begin{aligned} \text{maximize} \quad & r \\ \text{subject to} \quad & r \leq D^1(i,j), \quad \forall i,j \in [n] \\ & r \leq D^2(i,j), \quad \forall i,j \in [n] \\ & g_{k-1} \leq g_k, \quad \forall k \in [n] \\ & g_0 \geq 0, \quad g_n \leq 1. \end{aligned}$$

We use $L^1(\theta,\tau)$ to illustrate how we derive the discretized relaxation $D^1(i,j)$. $L^2(\theta,\tau)$ and other lower bounds derived in Section 4 can be relaxed in a similar way. Thus, a similar factor revealing LP can be formulated for unweighted general graphs. We omit the tedious details. Our codes for solving the linear programmings are available upon request.

[14]The same approach can be applied to formulate a factor revealing LP for the unweighted general case.

Suppose $\theta \in [\frac{i}{n}, \frac{i+1}{n})$ and $\tau \in [\frac{j}{n}, \frac{j+1}{n})$. We consider the terms of $L^1(\theta, \tau)$ one by one and obtain the following lower bounds.

(1) $\int_0^\theta (1-y)g(y)dy \geq \frac{1}{n}\sum_{k=0}^{j-1}(1-\frac{k+1}{n})g_k$ and $\int_0^\tau (1-y)g(y)dy \geq \frac{1}{n}\sum_{k=0}^{i-1}(1-\frac{k+1}{n})g_k$;
(2) $\int_0^\theta \min\{y,\tau\}m(y)dy \geq \frac{1}{n}\sum_{k=0}^{i-1}\frac{\min\{k,j\}}{n}\cdot m_k$ and $\int_0^\tau y\cdot m(y)dy \geq \frac{1}{n}\sum_{k=0}^{j-1}\frac{k}{n}\cdot m_k$, where each m_k is a new variable and we introduce two extra constraints $m_k \leq g_k$ and $m_k \leq 1-g_k$;
(3) $(1-\theta)(1-\theta+\tau)(1-g(\theta)) \geq (1-\frac{i+1}{n})(1-\frac{i+1}{n}+\frac{j}{n})(1-g_{i+1})$;
(4) $\frac{1}{2}(2-\tau-\theta)(\theta-\tau) \geq \frac{1}{2}(2-\frac{j+1}{n}-\frac{i+1}{n})(\frac{i}{n}-\frac{j+1}{n})$.

Observe that $D^1(i,j)$ asymptotically approaches $L^1(\theta,\tau)$ when n approaches infinity. Hence the optimal value of the discretized program approaches that of the continuous program when $n \to \infty$.

B HARDNESS RESULTS

B.1 RDO on Bipartite Graphs

In this section, we construct a bipartite graph called Double-Bomb that is similar to the one in [4]. We evaluate the average performance of RDO on the given graph by experiments. By doing so, we suggest an experimental hardness result for RDO.

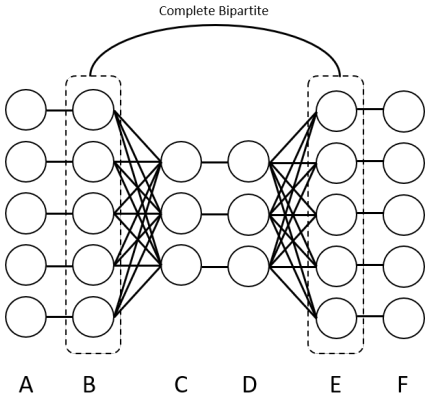


Fig. 8. Double-Bomb

Refer to Figure 8, vertices of Double-Bomb consist of 6 parts (A, B, C, D, E, F). C, D contain n_1 vertices each, and A, B, E, F contain n_2 vertices each. n_1, n_2 will be specified later in the experiment. The edges of the graph are defined as the following:

(1) $\forall i \in [n_1]$, let there be an edge $(C[i], D[i])$;
(2) $\forall i \in [n_2]$, let there be edges $(A[i], B[i])$ and $(E[i], F[i])$;
(3) $\forall i \in [n_1], j \in [n_2]$, let there be edges $(B[j], C[i])$ and $(D[i], E[j])$;
(4) $\forall i, j \in [n_1]$, let there be an edge $(B[i], E[j])$.

Each group of vertices shares the same preference order. Vertices in A, F have only one neighbor. Hence we don't need to specify the preferences. For vertices in B, they prefer vertices in E to vertices in C and finally to vertices in A. For vertices in C, they prefer vertices in B to vertices in D. For vertices in E, they prefer vertices in B to vertices in D and finally to vertices in F. For vertices in D, they prefer vertices in E to vertices in C. Here, within a group of vertices, the preference order is always from small index to large index.

We run experiments for different n_1, n_2 (each for 10^5 times), the average performance is shown in the following table.

$n_1 =$	100	200	500	1000
$n_2/n_1 = 1$	0.6514	0.6504	0.6499	0.6497
$n_2/n_1 = 1.3$	0.6479	0.6471	0.6465	0.6464
$n_2/n_1 = 1.5$	0.6474	0.6467	0.6461	0.646
$n_2/n_1 = 1.8$	0.6477	0.6471	0.6466	0.6465
$n_2/n_1 = 2$	0.6484	0.6478	0.6473	0.6471

We observe that the worst performance is achieved when $n_2/n_1 = 1.5$, which is close to 0.646. We leave as future work to analyze it theoretically.

B.2 RDO on General Graphs

In this section, we construct a non-bipartite graph with 4 vertices for which the maximum matching matches 4 vertices while the RDO algorithm matches 2.5 vertices in expectation. In other words, the approximation ratio of RDO on general graphs is at most 0.625. Together with Theorem 1.1, we show a separation on the approximation ratio of RDO on bipartite and general graphs.

Theorem B.1. *The approximation ratio of RDO for general graphs is at most* 0.625.

Proof. Let the four vertices be $\{a, b, c, d\}$, and let there be 4 edges: (a, b), (a, c), (b, c) and (c, d). Obviously, there exists a perfect matching.

Let the preference of all vertices be $c > b > a > d$. It is easy to check that unless d has the earliest decision time, RDO matches only one edge. Thus the expected size of the matching produced by RDO is $\frac{5}{4}$ while the maximum matching has size 2. □

B.3 IRP

Recall that in the IRP algorithm, the decision order is arbitrary while the preference orders are chosen independently and uniformly at random.

Theorem B.2. *The approximation ratio of IRP is at most* 0.5*, even for bipartite graphs.*

Proof. We state the instance from [9]. Let the set of vertices be $\{u_i, v_i\}_{i\in[n]}$, where u_i is connected to v_i, for all $i \in [n]$ and there is a complete bipartite graph between $\{u_1, \ldots, u_{\frac{n}{2}}\}$ and $\{u_{\frac{n}{2}+1}, \ldots, u_n\}$. Obviously, the maximum matching has size n. Suppose that the decision order is given by $(u_1, u_2, \ldots, u_n, v_1, v_2, \ldots, v_n)$. Then for all $i \le \frac{n}{2} - \sqrt{n}$, the probability of u_i matching v_i in IRP is at most $\frac{1}{\sqrt{n}}$. Therefore the expected size of the resulting matching, which is half the number of matched vertices, is at most

$$\frac{n}{2} + \left(\frac{n}{2} - \sqrt{n}\right) \cdot \frac{1}{\sqrt{n}} + \sqrt{n} < \frac{n}{2} + \frac{3}{2} \cdot \sqrt{n},$$

which gives an approximation ratio of $0.5 + o(1)$. □

Received 07 March 2021; revised 09 October 2022; accepted 26 July 2023