Improved Algorithms for Online Matching

Brian Brubach[*], Karthik A. Sankararaman[†], Aravind Srinivasan[‡], and Pan Xu[§]

Department of Computer Science, University of Maryland, College Park, MD 20742, USA

Abstract

Online matching has received significant attention over the last 15 years due to its close connection to Internet advertising. As the seminal work of Karp, Vazirani, and Vazirani has an optimal $(1 - 1/e)$ competitive ratio in the standard adversarial online model, much effort has gone into developing useful online models that incorporate some stochasticity in the arrival process. One such popular model is the "known I.I.D. model" where different customer-types arrive online from a known distribution. We develop algorithms with improved competitive ratios for some basic variants of this model with integral arrival rates, including: (a) the case of general weighted edges, where we improve the best-known ratio of 0.667 due to Haeupler, Mirrokni and Zadimoghaddam to 0.705; and (b) the vertex-weighted case, where we improve the 0.7250 ratio of Jaillet and Lu to 0.7299.

One of the key ingredients of our improvement is the following (offline) approach to bipartite-matching polytopes with additional constraints. We first add several valid constraints in order to get a good fractional solution $\mathbf{f}$; however, these give us less control over the structure of $\mathbf{f}$. We next *remove* all these additional constraints and randomly move from $\mathbf{f}$ to a feasible point on the matching polytope with all coordinates being from the set $\{0, 1/k, 2/k, \ldots, 1\}$ for a chosen integer k. This, of course, is a type of solution with tractable structure for algorithm design and analysis. The appropriate random move preserves many of the removed constraints (approximately/exactly, and with high probability or in expectation). This underlies some of our improvements, and, we hope, could be of independent interest.

[*]**Email:** `bbrubach@cs.umd.edu`

[†]**Email:** `kabinav@cs.umd.edu`

[‡]**Email:** `srin@cs.umd.edu`

Supported in part by NSF Awards CNS 1010789 and CCF 1422569, and by a research award from Adobe, Inc.

[§]**Email:** `panxu@cs.umd.edu`

Supported in part by NSF Awards CNS 1010789 and CCF 1422569

1 Introduction

Applications to Internet advertising have driven the study of online matching problems in recent years [14]. In these problems, we consider a bipartite graph $G = (U, V, E)$ in which the set U is available offline while the vertices in V arrive online. Whenever some vertex v arrives, it must be matched immediately to at most one vertex in U; each vertex u can be matched to at most one v. In the context of Internet advertising, U is the set of advertisers, V is a set of impressions, and the edges E define the impressions that interest a particular advertiser. When v arrives, we must choose an available advertiser (if any) to match with it. Since advertising forms the key source of revenue for many large Internet companies, finding good matching algorithms and obtaining even small performance gains can have high impact.

In the *stochastic known I.I.D.* model of arrival studied in this paper, we are given a bipartite graph $G = (U, V, E)$ in advance. During the online phase, each arriving vertex v is drawn with replacement from a known distribution on the vertices in V. This known distribution model captures the fact that we often have background data about the impressions, which allows us to predict the frequency with which each type of impression will arrive. Edge-weighted matching [4] is a general model in the context of advertising: every advertiser gains some given revenue for being matched to a particular type of client. Here, a *type* of client refers to a class of users (e.g., a demographic group) who are interested in the same subset of advertisements. A special case of this model is vertex-weighted matching [1], where the weights are associated only with the advertisers. In other words, a particular advertiser has the same revenue generated for matching any of the user types interested in it.

In some modern business models, revenue is generated not upon matching advertisements, but only when a user *clicks* on the particular advertisement: this is referred to as the *pay-per-click* model. From background data, one can assign the probability of a particular advertisement being clicked by a particular type of user. Works including [15, 16] capture this notion using a stochastic model on the edges: each edge has a probability associated with it denoting the probability of (independent) reward that particular assignment yields.

2 Preliminaries

In the *Unweighted Online Known I.I.D. Stochastic Bipartite Matching* problem, we are given a bipartite graph $G = (U, V, E)$. The set U is available offline while the vertices v arrive online and drawn with replacement from an I.I.D. distribution on V. For each $v \in V$, we are given an *arrival rate* r_v, which is the expected number of times v will arrive. With the exception of Section 5, this paper will focus on the integral-arrival-rates setting where all $r_v \in \mathbb{Z}^+$. As described in [7], WLOG we can assume that $\forall v \in V, r_v = 1$. Let $n = \sum_{v \in V} r_v$ be the number of vertices arriving during the online phase. Throughout we will use "WS" to refer to the worst case for various algorithms.

In the vertex-weighted variant, every vertex $u \in U$ has a weight w_u and we seek a maximum vertex weighted matching. In the edge-weighted variant, every edge $e \in E$ has a weight w_e and we seek a maximum weight matching.

Asymptotic assumption and notation. We will always assume n is large and analyze algorithms as n goes to infinity: e.g., if $x \leq 1 - (1 - 2/n)^n$, we will just write this as "$x \leq 1 - 1/e^2$" instead of the more-accurate

"$x \leq 1 - 1/e^2 + o(1)$". These suppressed $o(1)$ terms will subtract at most a $o(1)$ term from our competitive ratios.

2.1 LP Benchmark

We will use the following LP to upper bound the optimal offline solution and guide our algorithm. We will first show an LP for the unweighted variant, then describe changes for the vertex-weighted and edge-weighted settings. As usual, we have a variable f_e for each edge. Let $\partial(w)$ be the set of edges adjacent to a vertex $w \in U \cup V$ and let $f_w = \sum_{e \in \partial(w)} f_e$. Recall that $r_v \equiv 1$ WLOG.

$$\text{maximize} \sum_{e \in E} f_e \tag{2.1}$$

$$\text{subject to} \sum_{e \in \partial(u)} f_e \leq 1 \qquad \forall u \in U \tag{2.2}$$

$$\sum_{e \in \partial(v)} f_e \leq 1 \qquad \forall v \in V \tag{2.3}$$

$$0 \leq f_e \leq 1 - 1/e \qquad \forall e \in E \tag{2.4}$$

$$f_e + f_{e'} \leq 1 - 1/e^2 \qquad \forall e, e' \in \partial(u), e \neq e', \forall u \in U \tag{2.5}$$

Vertex-Weighted: In this variant the objective function is: maximize $\sum_{u \in U} \sum_{e \in \partial(u)} f_e w_u$.

Edge-Weighted: In this variant the objective function is: maximize $\sum_{e \in E} f_e w_e$.

Constraint 2.2 is the matching constraint for vertices in U. Constraint 2.3 is valid because each vertex in V has an arrival rate of 1. Constraint 2.4 (sometimes referred to as a *discounted* LP constraint) is used in [13] and [7]. It captures the fact that expected number of matches for any edge is at most $1 - 1/e$. This is a valid constraint for large n because the probability that a given vertex v doesn't arrive after n rounds is $1/e$. Constraint 2.5 is similar to the previous one, but for pairs of edges. For any two neighbors of a given $u \in U$, the probability that neither of its neighbors arrive will be $1/e^2$. Therefore, the sum of variables for any two distinct edges in $\partial(u)$ cannot exceed $1 - 1/e^2$. Notice that constraints 2.4 and 2.5 reduces the gap between the optimal LP solution and the performance of the optimal online algorithm. In fact, without constraint 2.4, we cannot in general achieve a competitive ratio better than $1 - 1/e$.

2.2 Related work

The study of online matching began with the seminal work of Karp, Vazirani, Vazirani [9], where they gave an optimal online algorithm for a version of the unweighted bipartite matching problem in which vertices arrive in adversarial order. Following that, a series of work has studied various related models. The book by Mehta [14] gives a detailed overview of the various work that have been done. The vertex-weighted version of this problem was first introduced by Aggarwal, Goel and Karande [1], where they give an $\left(1 - \frac{1}{e}\right)$ optimal ratio for the adversarial arrival model. The edge-weighted setting has been studied in the adversarial model by Feldman, Korula, Mirrokni and Muthukrishnan [4], where they consider an additional relaxation of "free-disposal".

Beyond the adversarial model, these problems have been studied under the name of *stochastic matching*, where the online vertices are assumed to either have a random order or be drawn I.I.D. from a known distribution. The works [3, 10, 11, 12] among others, study the random arrival-order model; papers including [2, 5, 7, 8, 13] study the problem under the I.I.D. arrival order.

Another variant of this problem is when the edges have stochastic rewards. In Section 5 of this paper, we give a simple $\left(1 - \frac{1}{e}\right)$-competitive algorithm for the case where the edges have arbitrary stochastic rewards and the online vertices arrive I.I.D. with arbitrary arrival rates. Models with stochastic rewards have been previously studied by [15], [16] among others, but not in the known I.I.D. model.

2.2.1 Related Work in the Vertex-Weighted and Unweighted Settings

The vertex-weighted and unweighted variants are similar and often studied together. In the vertex-weighted setting, each vertex $u \in U$ has a weight w_u and we seek a matching that maximizes this weight. The unweighted model can be thought of as a special case of vertex-weighted with all $w_u = 1$. Analysis of algorithms in these settings is typically done from the perspective of vertices in U.

These settings have seen a long line of results starting with Feldman, Mehta, Mirrokni and Muthukrishnan[5] who were the first to beat $1 - 1/e$ with a competitive ratio of 0.67 for the unweighted problem. This was improved by Manshadi, Gharan, and Saberi [13] to 0.705 using an *adaptive* algorithm. The term adaptive here means that when v arrives, it first checks which neighbors are available to be matched and only chooses among those available neighbors. By contrast, a *non-adaptive* algorithm may attempt to match v to an unavailable neighbor resulting in no gain. In addition, Manshadi, Gharan, and Saberi [13] showed that even in the unweighted variant with integral arrival rates, no algorithm can achieve a ratio better than $1 - e^{-2} \approx 0.86$.

Finally, Jaillet and Lu [8] presented an adaptive algorithm which crucially uses a clever LP to achieve 0.725 and 0.729 for the vertex-weighted and unweighted problems, respectively. They used this special LP to ensure that they could always find an optimal LP solution with $f_e \in \{0, 1/3, 2/3\}$ for all edges $e \in E$. This simplified fractional solution allowed for easier analysis at the expense of using a slightly weaker LP benchmark.

2.2.2 Related Work in the Edge-Weighted Setting

In the more general, edge-weighted version of the problem, each edge e has a weight w_e and we seek a matching that maximizes this weight. In contrast to the vertex-weighted and unweighted settings, analysis of algorithms in this setting is typically done from the perspective of the edges in E.

Haeupler, Mirrokni, Zadimoghaddam [7] were the first to beat $1 - 1/e$ by achieving a competitive ratio of 0.667. They employed two techniques commonly used in randomized algorithms. First, they use a *discounted LP* with tighter constraints compared to the basic matching LP. A similar LP can be seen in 2.1. Second, they used the *power of two choices* by constructing two matchings offline to guide their online algorithm. A vertex type from V arriving for the first time attempts to match with its neighbor in M_1; the second time with its neighbor in M_2. If some v arrives and its designated neighbor has already been matched, nothing happens.

2.3 Our Contributions

Problem	Previous Work	This Paper
Edge-Weighted (Section 3)	0.667 (Haeupler *et al.* [7])	0.705
Vertex-Weighted (Section 4)	0.725 (Jaillet and Lu [8])	0.7299
Unweighted	$1 - 2e^{-2}(\sim 0.7293)$ (Jaillet and Lu [8])	0.7299
Non-integral Stochastic Rewards (Section 5)	N/A	$1 - e^{-1}$

Theorem 2.1. *For vertex-weighted online stochastic matching with integral arrival rates, online algorithm* VW *achieves a competitive ratio of at least* 0.7299.

Theorem 2.2. *For edge-weighted online stochastic matching with integral arrival rates, there exists an algorithm which achieves a competitive ratio of at least* 0.7.

Theorem 2.3. *For edge-weighted online stochastic matching with integral arrival rates, online algorithm* EW$[q]$ (2) *with* $q = 0.149251$ *achieves a competitive ratio of at least* 0.70546.

Theorem 2.4. *For edge-weighted online stochastic matching with arbitrary arrival rates and fractional matching probabilities, online algorithm* SM (9) *achieves a competitive ratio of* $1 - 1/e$.

2.4 Overview of vertex-weighted algorithm and contributions

A key challenge encountered by [8] was that their special LP could lead to length four cycles of type C_1 shown in Figure 1. In fact, they used this cycle to show that no algorithm could perform better than $1 - 2/e^2 \approx 0.7293$ using their LP. They mentioned that tighter LP constraints such as 2.4 and 2.5 in the LP from Section 2 could avoid this bottleneck, but they did not propose a technique to use them. Note that the $\{0, 1/3, 2/3\}$ solution produced by their LP was an essential component of their Random List algorithm.

We show a randomized rounding algorithm to construct a similar, simplified $\{0, 1/3, 2/3\}$ vector from the solution of a stricter benchmark LP. This allows for the inclusion of additional constraints, most importantly constraint 2.5. Using this rounding algorithm combined with tighter constraints, we will upper bound the probability of a vertex appearing in the cycle C_1 from Figure 1 at $2 - 3/e \approx 0.89$. Additionally, we show how to deterministically break all other length four cycles which are not of type C_1 without creating any new cycles of type C_1. Finally, we describe an algorithm which utilizes these techniques to improve previous results in both the vertex-weighted and unweighted settings.

For this algorithm, we first solve the LP in Section 2 on the input graph. In Section 4, we show how to use the technique in sub-section 2.7 to obtain a sparse fractional vector. We then present a randomized online algorithm (similar to the one in [8]) which uses the sparse fractional vector as a guide to achieve a competitive ratio of 0.7299. Previously, there was gap between the best unweighted algorithm with a ratio of $1 - 2e^{-2}$ due to [8] and the negative result of $1 - e^{-2}$ due to [13]. We take a step towards closing that gap by showing that an algorithm can achieve $0.7299 > 1 - 2e^{-2}$ for both the unweighted and vertex-weighted variants with integral arrival rates.

(C_1)

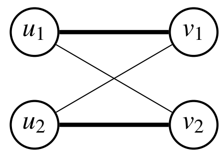

Figure 1: This cycle is the source of the negative result described by Jaillet and Lu [8]. Thick edges have $f_e = 2/3$ while thin edges have $f_e = 1/3$.

2.5 Overview of edge-weighted algorithm and contributions

A challenge that arises in applying the *power of two choices* to this setting is when the same edge (u, v) is included in both matchings M_1 and M_2. In this case, the copy of (u, v) in M_2 can offer no benefit and a second arrival of v is wasted. To use an example from related work, Haeupler *et al.* [7] choose two matchings in the following way. M_1 is attained by solving an LP with constraints 2.2, 2.3 and 2.4 and rounding to an integral solution. M_2 is constructed by finding a maximum weight matching and removing any edges which have already been included in M_1. A key element of their proof is showing that the probability of an edge being removed from M_2 is at most $1 - 1/e \approx 0.63$.

The approach in this paper is to construct two or three matchings together in a correlated manner to reduce the probability that some edge is included in all matchings. We will show a general technique to construct an ordered set of k matchings where k is an easily adjustable parameter. For $k = 2$, we show that the probability of an edge appearing in both M_1 and M_2 is at most $1 - 2/e \approx 0.26$.

For the algorithms presented, we first solve an LP on the input graph. We then round the LP solution vector to a sparse integral vector and use this vector to construct a randomly ordered set of matchings which will guide our algorithm during the online phase. We begin Section 3 with a simple warm-up algorithm which uses a set of two matchings as a guide to achieve a 0.688 competitive ratio, improving the best known result for this problem. We follow it up with a slight variation that improves the ratio to 0.7 and a more complex 0.705-competitive algorithm which relies on a convex combination of a 3-matching algorithm and a separate *pseudo-matching* algorithm.

2.6 Overview of edge-weighted algorithm for non-integral arrival rates and stochastic rewards

This algorithm is presented in Section 5. We believe the known I.I.D. model with stochastic rewards is an interesting new direction motivated by the work of [15] and [16] in the adversarial model. We introduce a new, more general LP specifically for this setting and show that a simple algorithm using the LP solution directly can achieve a competitive ratio of $1 - 1/e$. In [16], they show that no randomized algorithm can achieve a ratio better than $0.62 < 1 - 1/e$ in the adversarial model. Hence, achieving a $1 - 1/e$ for the i.i.d. model shows that this lower bound does not extend to this model.

2.7 LP rounding technique $\mathsf{DR}[\mathbf{f}, k]$

For the algorithms presented, we will first solve the benchmark LP in sub-section 2.1 for the input instance to get a fractional solution vector $\mathbf{f}$. We then round $\mathbf{f}$ to an integral solution $\mathbf{F}$ using a two step process we call $\mathsf{DR}[\mathbf{f}, k]$. The first step is to multiply $\mathbf{f}$ by k. The second step is to apply the powerful dependent rounding techniques of Gandhi, Khuller, Parthasarathy and Srinivasan [6] to this new vector. In this paper, we will always choose k to be 2 or 3. This will help us handle the fact that a vertex in V may appear more than once, but probably not more than two or three times.

While dependent rounding is typically applied to values between 0 and 1, the useful properties extend naturally to our case in which kf_e may be greater than 1 for some edge e. To understand this process, it is easiest to imagine splitting each kf_e into two edges with the integer value $f'_e = \lfloor kf_e \rfloor$ and fractional value $f''_e = kf_e - \lfloor kf_e \rfloor$. The former will remain unchanged by the dependent rounding since it is already an integer while the latter will be rounded to 1 with probability f''_e and 0 otherwise. Our final value F_e would be the sum of those two rounded values. The two properties of dependent rounding we will use are:

1) Marginal distribution: For every edge e, let $p_e = kf_e - \lfloor kf_e \rfloor$. Then, $\Pr[F_e = \lceil kf_e \rceil] = p_e$ and $\Pr[F_e = \lfloor kf_e \rfloor] = 1 - p_e$.

2) Degree-preservation: For any vertex $w \in U \cup V$, let its fractional degree kf_w be $\sum_{e \in \partial(w)} kf_e$ and integral degree be the random variable $F_w = \sum_{e \in \partial(w)} F_e$. Then $F_w \in \{\lfloor kf_w \rfloor, \lceil kf_w \rceil\}$.

3 Edge-weighted matching with integral arrival rates

In this section, we will consider the edge-weighted bipartite matching problem under the known IID model with integral arrivals rates. One set of vertices are fixed while the other set of vertices arrive in an online fashion and are drawn from a uniform distribution independently and identically. Formally, we are given a graph $G(U, V, E)$, where U is fixed and each $v \in V$ arrives online. Each edge $e \in E$ has a weight w_e. The edge set E and the corresponding weight W is known to the algorithm beforehand.

3.1 A simple 0.688-competitive algorithm

As a warm-up, we will describe a simple algorithm which achieves a competitive ratio of 0.688 and introduces key ideas in our approach. We begin by solving the LP in sub-section 2.1 to get a fractional solution vector $\mathbf{f}$ and applying $\mathsf{DR}[\mathbf{f}, 2]$ as described in Subsection 2.7 to get an integral vector $\mathbf{F}$. We construct a bipartite graph $G_{\mathbf{F}}$ with F_e copies of each edge e. Note that $G_{\mathbf{F}}$ will have max degree 2 since for all $w \in U \cup V$, $F_w \leq \lceil 2f_w \rceil \leq 2$ and therefore we can decompose it into two matchings using *Hall's Theorem*. Finally, we randomly permute the two matchings into an ordered pair of matchings, $[M_1, M_2]$. These matchings serve as a guide for the online phase of the algorithm, similar to [7]. The entire warm-up algorithm for the edge-weighted model, denoted by EW_0, is summarized in Algorithm 1.

Algorithm 1: [EW_0]

1 Construct and solve the benchmark LP in sub-section 2.1 for the input instance.
2 Let $\mathbf{f}$ be an optimal fraction solution vector. Invoke $\mathsf{DR}[\mathbf{f}, 2]$ to get an integral vector $\mathbf{F}$.
3 Create the graph $G_{\mathbf{F}}$ with F_e copies of each edge $e \in E$ and decompose it into two matchings.
4 Randomly permute the matchings to get a *random ordered* pair of matchings, say $[M_1, M_2]$.
5 When a vertex v arrives for the first time, try to assign v to some u_1 if $(u_1, v) \in M_1$; when v arrives for the second time, try to assign v to some u_2 if $(u_2, v) \in M_2$.
6 When a vertex v arrives for the third time or more, do nothing in that step.

3.1.1 Analysis of algorithm EW_0

We will show that EW_0 (Algorithm 1) achieves a competitive ratio of 0.688. Let $[M_1, M_2]$ be our randomly ordered pair of matchings. Note that there might exist some edge e which appears in both matchings if $f_e > 1/2$. Therefore, we consider three types of edges. We say an edge e is of type ψ_1, denoted by $e \in \psi_1$, iff e appears *only* in M_1. Similarly $e \in \psi_2$, iff e appears *only* in M_2 and $e \in \psi_b$, iff e appears in *both* M_1 and M_2. Let P_1, P_2, P_b be the probabilities of getting matched for $e \in \psi_1$, $e \in \psi_2$, and $e \in \psi_b$ respectively. According to the result in Haeupler *et al.* [7], the respective values are shown as follows.

Lemma 3.1. *Given M_1 and M_2, we have in the worst case (1) $P_1 = 0.5808$; (2) $P_2 = 0.14849$ and (3) $P_b = 0.632$.*

Proof. The detailed proof for Lemma 3.1 can be found in Section 3 of [7]. □

Let us now prove that EW_0 achieves a competitive ratio of at least 0.688.

Proof. Consider following two cases.

Case 1: $0 \leq f_e \leq 1/2$

By the marginal distribution property of dependent rounding, there can be at most one copy of e in $G_{\mathbf{F}}$ and the probability of including e in $G_{\mathbf{F}}$ is $2f_e$. Since an edge in $G_{\mathbf{F}}$ can appear in either M_1 or M_2 with equal probability 1/2, we have $\Pr[e \in \psi_1] = \Pr[e \in \psi_2] = f_e$. Thus, the final ratio is $(f_e P_1 + f_e P_2)/f_e = P_1 + P_2 = 0.729$.

Case 2: $1/2 \leq f_e \leq 1 - 1/e$

By the marginal distribution property, we know $\Pr[e \in \psi_b] = \Pr[F_e = \lceil 2f_e \rceil] = 2f_e - \lfloor 2f_e \rfloor = 2f_e - 1$. It follows that $\Pr[e \in \psi_1] = \Pr[e \in \psi_2] = (1/2)(1 - (2f_e - 1)) = 1 - f_e$. Thus, the final ratio is $((1 - f_e)(P_1 + P_2) + (2f_e - 1)P_b)/f_e \geq 0.688$, where the WS is for an edge e with $f_e = 1 - 1/e$.

□

3.2 A 0.7-competitive algorithm

In this section, we describe an improvement upon the previous warm-up algorithm to get a competitive ratio of 0.7. We start by making an observation about the performance of the warm-up algorithm. After solving the LP, let edges with $f_e > 1/2$ be called *large* and edges with $f_e \leq 1/2$ be called *small*. Let L and S, be the sets of large and small edges, respectively. Notice that in the previous analysis, small edges achieved a much higher competitive ratio of 0.729 versus 0.688 for large edges. This is primarily due to the fact that we may get two copies of a large edge in $G_{\mathbf{F}}$. In this case, the copy in M_1 has a better chance of being matched, since there is no edge which can block it, but the copy that is in M_2 has no chance of being matched.

To correct this imbalance, we make an additional modification to the f_e values *before* applying $\mathsf{DR}[\mathbf{f}, k]$. The rest of the algorithm is exactly the same. Let η be a parameter to be optimized later. For all large edges $\ell \in L$ such that $f_\ell > 1/2$, we set

$$f_\ell = f_\ell + \eta$$

For all small edges $s \in S$ which are adjacent to some large edge, let $\ell \in L$ be the largest edge adjacent to s such that $f_\ell > 1/2$. Note that it is possible for e to have two large neighbors, but we only care about the largest one. We set

$$f_s = f_s \left(\frac{1 - (f_\ell + \eta)}{1 - f_\ell} \right)$$

In other words, we increase the values of large edges while ensuring that for all $w \in U \cup V$, $f_w \leq 1$ by reducing the values of neighboring small edges proportional to their original values. Note that it is not possible for two large edges to be adjacent since they must both have $f_e > 1/2$. For all other small edges which are not adjacent to any large edges, we leave their values unchanged. We then apply $\mathsf{DR}[\mathbf{f}, 2]$ to this new vector, multiplying by 2 and applying dependent rounding as before.

3.2.1 Analysis

We can now prove Theorem 2.2.

Proof. As in the warm-up analysis, we'll consider large and small edges separately

- $0 \leq f_s \leq \frac{1}{2}$: Here we have two cases
  - Case 1: s is not adjacent to any large edges.

    In this case, the analysis is the same as the warm-up algorithm and we still get a 0.729 competitive ratio for these edges.
  - Case 2: s is adjacent to some large edge ℓ.

    For this case, let f_ℓ be the value of the largest neighboring edge in the original LP solution. Then s achieves a ratio of

    $$f_s \left(\frac{1 - (f_\ell + \eta)}{1 - f_\ell} \right) (0.1484 + 0.5803)/f_s = \left(\frac{1 - (f_\ell + \eta)}{1 - f_\ell} \right) (0.1484 + 0.5803)$$

    Note that for $f_\ell \in [0, 1)$ this is a decreasing function with respect to f_ℓ. So the worst case is $f_\ell = 1 - 1/e$ and we have a ratio of

    $$\left(\frac{1 - (1 - 1/e + \eta)}{1 - (1 - 1/e)} \right) (0.1484 + 0.5803) = \left(\frac{1/e - \eta}{1/e} \right) (0.1484 + 0.5803)$$

- $\frac{1}{2} < f_\ell \leq 1 - \frac{1}{e}$:

  Here, the ratio is $((1-(f_\ell+\eta))(P_1+P_2)+(2(f_\ell+\eta)-1)P_b)/f_\ell$, where the WS is for an edge e with $f_\ell = 1-1/e$ since this is a decreasing function with respect to f_ℓ.

Choosing the optimal value of $\eta = 0.0142$, yields an overall competitive ratio of 0.7 for this new algorithm.

□

3.3 A 0.705-competitive algorithm

In this section, we will describe an algorithm EW (Algorithm 2), that achieves a competitive ratio of 0.705. The algorithm first solves the benchmark LP in sub-section 2.1 and obtains a fractional optimal solution $\mathbf{f}$. By invoking DR[$\mathbf{f}$, 3], it obtains a random integral solution $\mathbf{F}$. Notice that from LP constraint 2.4 we see $f_e \leq 1-1/e \leq 2/3$. Therefore after DR[$\mathbf{f}$, 3], each $F_e \in \{0,1,2\}$. Consider the graph $G_{\mathbf{F}}$ where each edge e is associated with the value of F_e. We say an edge e is *large* if $F_e = 2$ and *small* if $F_e = 1$ (note that this differs from the definition of large and small in the previous sub-section).

We design two non-adaptive algorithms, denoted by EW_1 and EW_2, which take the sparse graph $G_{\mathbf{F}}$ as input. The difference between the two algorithms EW_1 and EW_2 is that EW_1 favors the small edges while EW_2 favors the large edges. The final algorithm is to take a convex combination of EW_1 and EW_2 i.e. run EW_1 with probability q and EW_2 with probability $1-q$.

Algorithm 2: EW[q]

1. Construct and solve the benchmark LP in sub-section 2.1 for the input instance. Let $\mathbf{f}$ be the optimal solution vector.
2. Invoke DR[$\mathbf{f}$, 3] to obtain the vector $\mathbf{F}$.
3. Independently run EW_1 and EW_2 with probabilities q and $1-q$ respectively on $G_{\mathbf{F}}$.

3.3.1 Algorithm EW_1

In this section, we describe the randomized algorithm EW_1 (Algorithm 3). Suppose we view the graph of $G_{\mathbf{F}}$ in another way where each edge has F_e copies. Let PM[$\mathbf{F}$, 3] refer to the process of constructing the graph $G_{\mathbf{F}}$ with F_e copies of each edge, decomposing it into three matchings, and randomly permuting the matchings. EW_1 first invokes PM[$\mathbf{F}$, 3] to obtain a *random ordered* triple of matchings, say $[M_1, M_2, M_3]$. Notice that from the LP constraint 2.4 and the properties of DR[$\mathbf{f}$, 3] and PM[$\mathbf{F}$, 3], an edge will appear in at most two of the three matchings. For a small edge $e = (u, v)$ in $G_{\mathbf{F}}$, we say e is of type Γ_1 if u has two other neighbors v_1 and v_2 in $G_{\mathbf{F}}$ with $F_{(u,v_1)} = F_{(u,v_2)}$=1. We say e is of type Γ_2 if u has exactly one other neighbor v_1 with $F_{(u,v_1)} = 2$. WLOG we can assume that for every u, $F_u = \sum_{e \in \partial(u)} F_e = 3$; otherwise, we can add a dummy node v' to the neighborhood of u.

Note, we use the terminology, assign v to u to denote that edge (u, v) is matched by the algorithm if u is not matched until that step.

Here, h is a parameter we will fix at the end of analysis. Let R[EW_1, 1/3] and R[EW_1, 2/3] be the competitive ratio for a small edge and large edge respectively.

Algorithm 3: $\mathsf{EW}_1[h]$

1 Invoke $\mathsf{PM}[\mathbf{F},3]$ to obtain a *random ordered* triple matchings, say $[M_1,M_2,M_3]$.
2 When a vertex v comes for the first time, assign v to some u_1 with $(u_1,v) \in M_1$.
3 When v comes for the second time, assign v to some u_2 with $(u_2,v) \in M_2$.
4 When v comes for the third time, if e is either a large edge or a small edge of type Γ_1 then assign v to some u_3 with $e = (u_3,v) \in M_3$. However, if e is a small edge of type Γ_2 then *with probability* h, assign v to some u_3 with $e = (u_3,v) \in M_3$; otherwise, do nothing.
5 When v comes for the fourth or more time, do nothing in that step.

Lemma 3.2. *For* $h = 0.537815$*,* EW_1 *achieves a competitive ratio* $\mathsf{R}[\mathsf{EW}_1,2/3] = 0.679417$*,* $\mathsf{R}[\mathsf{EW}_1,2/3] = 0.751066$ *for a large and small edge respectively.*

Proof. In case of the large edge e, we divide the analysis into three cases where each case corresponds to e being in one of the three matchings. And we combine these conditional probabilities using Bayes' theorem to get the final competitive ratio for e. For each of the two types of small edges, we similarly condition them based on the matching they can appear in, and combine them using Bayes' theorem. Complete proof can be found in section 8.1 in Appendix. □

3.3.2 Algorithm EW_2

EW_2 (Algorithm 5) is a non-adaptive algorithm which takes $G_{\mathbf{F}}$ as input and performs well on the *large edges*. Recall that the previous algorithm, EW_1, first invokes $\mathsf{PM}[\mathbf{F},3]$ to obtain a *random ordered* triple of matchings. In contrast, EW_2 will invoke a routine, denoted by $\mathsf{PM}^*[\mathbf{F},2]$ (Algorithm 4), to generate a (*random ordered*) pair of pseudo-matchings from $\mathbf{F}$. Recall that $\mathbf{F}$ is an integral solution vector where $\forall\, e \;\; F_e \in \{0,1,2\}$. WLOG, we can assume that $F_v = 1$ for every v in $G_{\mathbf{F}}$.

Algorithm 4: $\mathsf{PM}^*[\mathbf{F},2][y_1,y_2]$

1 Suppose v has two neighbors in $G_{\mathbf{F}}$, say u_1,u_2, with $e_1 = (u_1,v)$ being a large edge while $e_2 = (u_2,v)$ being a small edge. Add e_1 to the primary matching M_1 and e_2 to the secondary matching M_2.
2 Suppose v has three neighbors in $G_{\mathbf{F}}$ and the incident edges are $\partial(v) = (e_1,e_2,e_3)$. Take a random permutation of $\partial(v)$, say $(\pi_1,\pi_2,\pi_3) \in \Pi(\partial(v))$. Add π_1 to M_1 with probability y_1 and π_2 to M_2 with probability y_2.

Here $0 \le y_1, y_2 \le 1$ are parameters which will be fixed after the analysis. Algorithm 5 describes EW_2.

Algorithm 5: $[\mathsf{EW}_2][y_1,y_2]$

1 Invoke $\mathsf{PM}^*[\mathbf{F},2][y_1,y_2]$ to generate a *random ordered* pair of pseudo-matchings, say $[M_1,M_2]$.
2 When a vertex v comes for the first time, assign v to some u_1 if $(u_1,v) \in M_1$; When v comes for the second time, try to assign v to some u_2 if $(u_2,v) \in M_2$.
3 When a vertex v comes for the third or more time, do nothing in that step.

Let $\mathsf{R}[\mathsf{EW}_2,1/3]$ and $\mathsf{R}[\mathsf{EW}_2,2/3]$ be the competitive ratios for small edges and large edges, respectively.

Lemma 3.3. *For* $y_1 = 0.687$ *and* $y_2 = 1$, $\mathsf{EW}_2[y_1, y_2]$ *achieves a competitive ratio of* $\mathsf{R}[\mathsf{EW}_2, 2/3] = 0.8539$ *and* $\mathsf{R}[\mathsf{EW}_2, 1/3] = 0.4455$ *for a large and small edge respectively.*

Proof. We analyze this on a case-by-case basis by considering the local neighborhood of the edge. A large edge can have two possible cases in its neighborhood, while a small edge can have eight possible cases. Choosing the worst case among the two for large edge and the worst case among the eight for the small edge, we prove the claim. Complete details of the proof can be found in section 8.1 in Appendix. □

3.3.3 Convex Combination of EW_1 and EW_2

In this section, we will prove theorem 2.3.

Proof. Let (a_1, b_1) be the competitive ratios achieved by EW_1 for large and small edges, respectively. Similarly, let (a_2, b_2) denote the same for EW_2.

We will have the following two cases.

- $0 \leq f_e \leq \frac{1}{3}$: By marginal distribution property of $\mathsf{DR}[\mathbf{f}, 3]$, we know that $\Pr[F_e = 1] = 3f_e$. Thus, the final ratio is
$$3f_e(qb_1/3 + (1-q)b_2/3)/f_e = qb_1 + (1-q)b_2$$
- $1/3 \leq f_e \leq 1 - 1/e$: By the same properties of $\mathsf{DR}[\mathbf{f}, 3]$, we know that $\Pr[F_e = 2] = 3f_e - 1$ and $\Pr[F_e = 1] = 2 - 3f_e$. Thus, the final ratio is
$$\Big((3f_e - 1)(2qa_1/3 + 2(1-q)a_2/3) + (2 - 3f_e)(qb_1/3 + (1-q)b_2/3)\Big)/f_e$$

The competitive ratio of the convex combination is maximized at $q = 0.149251$ with a value of 0.70546.

□

4 Vertex-weighted stochastic I.I.D. matching with integral arrival rates

In this section, we will consider vertex-weighted online stochastic matching on a bipartite graph G under known $I.I.D.$ model with integral arrival rates. We will present an algorithm in which each u has a competitive ratio of at least 0.72998. Recall that after invoking $\mathsf{DR}[\mathbf{f}, 3]$, we can obtain a (*random*) integral vector $\mathbf{F}$ with $F_e \in \{0, 1, 2\}$. Define $\mathbf{H} = \mathbf{F}/3$ and let $G_{\mathbf{H}}$ be the graph induced by $\mathbf{H}$ and each edge takes the value $H_e \in \{0, 1/3, 2/3\}$. In this section, we focus on the sparse graph $G_{\mathbf{H}}$. The main steps of the algorithm are as follows.

1. Solve the vertex-weighted version of benchmark LP in sub-section 2.1. Let $\mathbf{f}$ be an optimal solution vector.
2. Invoke $\mathsf{DR}[\mathbf{f}, 3]$ to obtain an integral vector $\mathbf{F}$ and a fractional vector $\mathbf{H}$ with $\mathbf{H} = \mathbf{F}/3$.
3. Apply a series of modifications to $\mathbf{H}$ and transform it to another solution $\mathbf{H}'$. See sub-section 4.2.
4. Run the randomized list algorithm (RLA) [8] induced by $\mathbf{H}'$ on the graph $G_{\mathbf{H}}$.

The WS for vertex-weighted case in [8] is shown in Figure 2, which arrived at node u with a competitive ratio of 0.725. From their analysis, we find node u_1 has a competitive ratio of at least 0.736. Hence, we *boost* the performance of u at the cost of u_1. In other words, we increase the value of $H_{(u,v1)}$ and decrease the value $H_{(u_1,v_1)}$. Case (10) and (11) in Figure 7 illustrates this. After this modification, the new WS for vertex-weighted is now the C_1 cycle shown in Figure 1. In fact, this is the WS for the unweighted case in [8]. However, Lemma 4.1 and the cycle breaking algorithm, implies that C_1 cycle can be avoided with probability at least $3/e - 1$. This helps us improve the ratio even for the unweighted case in [8].

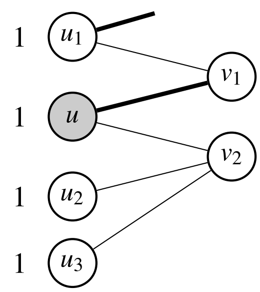

Figure 2: The worst case identified by Jaillet and Lu [8] for their vertex-weighted case. Thin edges have $H_e = 1/3$ and thick edges have $H_e = 2/3$.

Consider the graph $G_{\mathbf{H}}$ obtained after DR[$\mathbf{f}$, 3]. Notice that for some vertex u to appear in a C_1 cycle, it must have a neighboring edge with $H_e = 2/3$. Hence, if we bound the probability of this event, we can bound the probability of C_1 occuring after DR[$\mathbf{f}$, 3].

Lemma 4.1. *For any $u \in U$, the probability that u has a neighboring edge e in the graph $G_{\boldsymbol{H}}$ with $H_e = 2/3$ after* DR[$\boldsymbol{f}$, 3] *is at most $2 - 3/e$.*

Proof. It is easy to see that for some $e \in \partial(u)$ with $f_e \leq 1/3$, $F_e \leq 1$ after DR[$\mathbf{f}$, 3], and hence $H_e = F_e/3 \leq 1/3$. Thus only those edges $e \in \partial(u)$ with $f_e > 1/3$ will possibly be rounded to $H_e = 2/3$. Note that, there can be at most two such edges in $\partial(u)$, since $\sum_{e \in \partial(u)} f_e \leq 1$. Hence, we have the following two cases.

Case 1: $\partial(u)$ contains only one edge e with $f_e > 1/3$.
Let $q_1 = \Pr[H_e = 1/3]$ and $q_2 = \Pr[H_e = 2/3]$ after DR[$\mathbf{f}$, 3]. By DR[$\mathbf{f}$, 3], we know that

$$\mathbb{E}[H_e] = \mathbb{E}[F_e]/3 = q_2(2/3) + q_1(1/3) = f_e$$

Notice that $q_1 + q_2 = 1$ and hence $q_2 = 3f_e - 1$. Since this is an increasing function of f_e and $f_e \leq 1 - 1/e$ from LP constraint 2.4, we have $q_2 \leq 3(1 - 1/e) - 1 = 2 - 3/e$.

Case 2: $\partial(u)$ contains two edges e_1 and e_2 with $f_{e_1} > 1/3$ and $f_{e_2} > 1/3$.

Let q_2 be the probability that after DR[$\mathbf{f}$, 3], either $H_{e_1} = 2/3$ or $H_{e_2} = 2/3$. Note that, these two events are mutually exclusive since $H_u \leq 1$. Using the analysis from case 1, it follows that

$$q_2 = (3f_{e_1} - 1) + (3f_{e_2} - 1) = 3(f_{e_1} + f_{e_2}) - 2$$

From LP constraint 2.5, we know that $f_{e_1} + f_{e_2} \leq 1 - 1/e^2$, and hence $q_2 \leq 3(1 - 1/e^2) - 2 < 2 - 3/e$.

□

4.1 RLA Algorithm

Let $\mathbf{H}$ and $G_{\mathbf{H}}$ be the (random)fractional vector and corresponding induced graph respectively, obtained after invoking $\mathsf{DR}[\mathbf{f}, 3]$. Now we describe how the randomized list algorithm induced by $\mathbf{H}$, denoted by $\mathsf{RLA}[\mathbf{H}]$, works on the graph $G_{\mathbf{H}}$. The author is encouraged to refer [8] for more details.

Our goal is to generate a distribution over $\Pi(\partial(v))$, which denotes the set of all permutations of nodes incident to u. Here we refer to a permutation of all nodes incident to u as a random list $\mathcal{R}_v$. Let $H_u = \sum_{v \sim u} H_{(u,v)}$ and $H_v = \sum_{u \sim v} H_{(u,v)}$ for each u and v. WLOG assume for each v, $H_v = 1$; otherwise, we can add a dummy node u' to v with $H_{(u',v)} = 1 - H_v$ where u' is assumed to be matched at the beginning. The distribution $\mathcal{D}_v[\mathbf{H}]$ based on $\mathbf{H}$ is generated as follows:

- Suppose v has only two neighbors in $G_{\mathbf{H}}$, say u_1 and u_2. Then $\Pr[\mathcal{R}_v = (u_1, u_2)] = H_{(u_1,v)}$, and $\Pr[\mathcal{R}_v = (u_2, u_1)] = H_{(u_2,v)}$.
- Suppose v has three neighbors in $G_{\mathbf{H}}$. Then take a random permutation $(u_i, u_j, u_k) \in \Pi(\partial(v))$,

$$\Pr[\mathcal{R}_v = (u_i, u_j, u_k)] = H_{(u_i,v)} \frac{H_{(u_j,v)}}{H_{(u_j,v)} + H_{(u_k,v)}}$$

The resultant Random List Algorithm $\mathsf{RLA}[\mathbf{H}]$, is shown in Algorithm 6.

Algorithm 6: $\mathsf{RLA}[\mathbf{H}]$ (Random List Algorithm induced by $\mathbf{H}$)

```
1 When a vertex v comes, choose a random list R_v according to distribution D_v.
2 If all u in the list are matched, then drop the vertex v, otherwise, assign v to the first unmatched node
  u in the list.
```

4.2 Two kinds of Modifications to H

In this section, we discuss two kinds of modifications made to $\mathbf{H}$ before applying RLA. Similar steps of modifications was done by Jaillet and Lu [8], but here we change them to suit our algorithm.

4.2.1 The first modification: breaking cycles

There are three possible cycles of length 4 in the graph $G_{\mathbf{H}}$, as shown in Figure 3. Here we denote the first, second, and third type as C_1, C_2, and C_3 respectively. In [8], they give an efficient way to break C_2 and C_3, as shown in Figure 3. Cycle C_1 cannot be modified further and hence, is the bottleneck for their unweighted case.

Notice that, while breaking the cycles of C_2 and C_3, new cycles of C_1 can be created in the graph. Since our randomized construction of solution $\mathbf{H}$ gives us control on the probability of cycles C_1 occurring, we would like to break C_2 and C_3 in a controlled way, so as to not create any new C_1 cycles. This procedure is summarized in Algorithm 7.

Hence, we have the following properties after applying Algorithm 7, which is stated in Lemma 4.2. The proof of Lemma 4.2 can be found in section 8.2.2 of the Appendix.

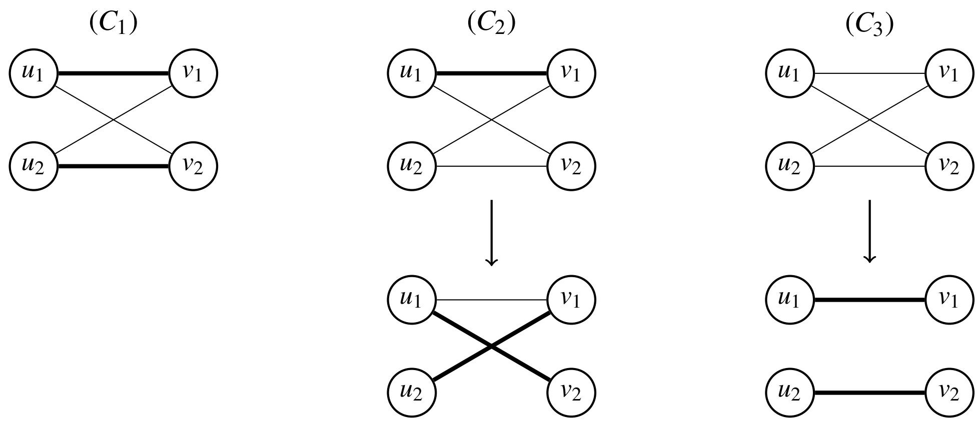

Figure 3: The three possible types of cycles of length 4 after applying $\mathsf{DR}[\mathbf{f}, 3]$. Thin edges have $H_e = 1/3$ and thick edges have $H_e = 2/3$.

Algorithm 7: [Cycle breaking algorithm] Offline Phase

1. While there is some cycle of type C_2 or C_3, Do:
2. Break all cycles of type C_2.
3. Break one cycle of type C_3 and return to the first step.

Lemma 4.2. *After applying Algorithm 7 to $G_{\boldsymbol{H}}$, we have (1) the value H_w is preserved for each $w \in U \cup V$; (2) no cycle of type C_2 or C_3 exists; (3) no new cycle of type C_1 is added.*

4.2.2 The second modification

Informally, this second modification decreases the rates of lists associated with those nodes u with $H_u = 1/3$ or $H_u = 2/3$ and increases the rates of lists associated with nodes u with $H_u = 1$. We will illustrate this with the following example.

Figure 4: An example of the need for the second modification. For the left: competitive analysis shows that in this case, u_1 and u_2 can achieve a high competitive ratio at the expense of u. For the right: an example of balancing strategy by making v_1 and v_2 slightly more likely to pick u when it comes.

Consider the graph G in Figure 4. Let thin and thick edges represent $H_e = 1/3$ and $H_e = 2/3$ respectively. We will now calculate the competitive ratio after applying RLA on G. Let $\mathsf{P_u}$ denote the probability that u gets matched after the algorithm. Let B_u denote the event that among the n random lists, there exists a list starting with u and G_u^v denote the event that among the n lists, there exists successive lists such that (1) Each of those lists starts with a $u' \neq u$ and $u' \in \partial(v)$ and (2) The lists arrive in an order which ensures u will be matched by the algorithm. From lemma 4 and Corollary 1 in [8], the following lemma follows:

Lemma 4.3. *Suppose u is not a part of any cycle of length 4. We have*

$$\mathsf{P}_u = 1 - (1 - \Pr[B_u]) \prod_{v \sim u} (1 - \Pr[G_u^v]) + o(1/n)$$

For the node u, we have $\Pr[B_u] = 1 - e^{-1}$. From definition, $G_u^{v_1}$ is the event that among the n lists, the random list $\mathcal{R}_{v_1} = (u_1, u)$ comes at least twice. Notice that the list $\mathcal{R}_{v_1} = (u_1, u)$ comes with probability $\frac{1}{3n}$. Thus we have $\Pr[G_u^{v_1}] = \Pr[X \geq 2] = 1 - e^{-1/3}(1 + 1/3)$, where $X \sim \text{Pois}(1/3)$. Similarly, we can get $\Pr[G_u^{v_2}] = 1 - e^{-2/3}(1 + 2/3)$ and the resultant $\mathsf{P}_u = 1 - \frac{20}{9e^2} \sim 0.699$. Observe that $\mathsf{P}_{u_1} \geq \Pr[B_{u_1}] = 1 - e^{-1/3}$ and $\mathsf{P}_{u_2} \geq \Pr[B_{u_2}] = 1 - e^{-2/3}$. Let $\mathsf{R}[\mathsf{RLA}, 1]$, $\mathsf{R}[\mathsf{RLA}, 1/3]$ and $\mathsf{R}[\mathsf{RLA}, 2/3]$ be the competitive ratio achieved by RLA for u, u_1 and u_2 respectively. Hence, we have $\mathsf{R}[\mathsf{RLA}, 1] \sim 0.699$ while $\mathsf{R}[\mathsf{RLA}, 1/3] \geq 3(1 - e^{-1/3}) \sim 0.8504$ and $\mathsf{R}[\mathsf{RLA}, 2/3] \geq 0.729$.

Intuitively, one can improve the worst case ratio by increasing the arrival rate for $\mathcal{R}_{v_1} = (u, u_1)$ while reducing that for $\mathcal{R}_{v_1} = (u_1, u)$. Suppose one modifies $H_{(u_1,v_1)}$ and $H_{(u,v_1)}$ to $H'_{(u_1,v_1)} = 0.1$ and $H'_{(u,v_1)} = 0.9$, the arrival rate for $\mathcal{R}_{v_1} = (u, u_1)$ and $\mathcal{R}_{v_1} = (u_1, u)$ gets modified to $0.1/n$ and $0.9/n$ respectively. The resulting changes are $\Pr[B_u] = 1 - e^{-0.9-1/3}$, $\Pr[G_u^{v_1}] = 1 - e^{-0.1}(1 + 0.1)$, $\mathsf{R}[\mathsf{RLA}, 1] = 0.751$, $\Pr[B_{u_1}] = 1 - e^{-1/3}$, $\Pr[G_{u_1}^{v_1}] \sim 0.227$ and $\mathsf{R}[\mathsf{RLA}, 1/3] \geq 0.8$. Hence, the performance on WS instance improves. Notice that after the modifications, $H'_u = H'_{(u,v_1)} + H_{(u,v_2)} = 0.9 + 1/3$.

To summarize, we have the following modifications. Firstly, our cycle breaking algorithm is more constrained, because we want to carefully decompose the cycles without introducing cycles C_1. Secondly, we also modify $\mathbf{H}$ in many cases, as described in Figure 7. These additional changes further help the WS described in Figure 1 and 2. Note that, we encounter two additional cases α_6 and α_7, since $\mathbf{H}$ is obtained differently. The complete details of these modifications can be found in section 8.2 of the Appendix.

Let $\mathbf{H}'$ be the solution vector obtained by applying two kinds of modifications to $\mathbf{H}$. The algorithm for the vertex-weighted case, denoted by VW, is summarized below.

Algorithm 8: VW [Vertex Weighted]

1. Construct and solve the LP in sub-section 2.1 for the input instance.
2. Invoke $\mathsf{DR}[\mathbf{f}, 3]$ to output $\mathbf{F}$ and $\mathbf{H}$. Apply the two kinds of modifications to morph $\mathbf{H}$ to $\mathbf{H}'$.
3. Run $\mathsf{RLA}[\mathbf{H}']$ on the graph $G_{\mathbf{H}}$.

4.3 Analysis of Algorithm VW

The algorithm VW consists of two different random processes: sub-routine $\mathsf{DR}[\mathbf{f}, 3]$ in the offline phase and RLA in the online phase. Consequently, the analysis consists of two parts. First, for a given graph $G_{\mathbf{H}}$, we analyze the ratio of $\mathsf{RLA}[\mathbf{H}']$ for each node u with $H_u = 1/3, H_u = 2/3$ and $H_u = 1$. The analysis is similar to [8]. Second, we analyze the probability that $\mathsf{DR}[\mathbf{f}, 3]$ transforms each u, with fractional f_u values, into the three discrete cases seen in the first part. By combining the results from these two parts we get the final ratio.

4.3.1 Competitive ratio analysis for $\mathsf{RLA}[\mathbf{H}']$

For a given $\mathbf{H}$ and $G_{\mathbf{H}}$, let P_u be the probability that u gets matched in $\mathsf{RLA}[\mathbf{H}']$. Notice that the value P_u is determined not just by the algorithm RLA itself, but also the modifications applied to $\mathbf{H}$. We define the competitive ratio of a vertex u achieved by RLA as P_u/H_u, after modifications. Lemma 4.4 gives the respective ratio values. The proof of Lemma 4.4 can be found in section 8.2.3 of the Appendix.

Lemma 4.4. *Consider a given $\boldsymbol{H}$ and a vertex u in $G_{\boldsymbol{H}}$. The respective ratios achieved by* RLA *after the modifications are as described below.*

- *If $H_u = 1$, then the competitive ratio* $\mathsf{R}[\mathsf{RLA}, 1] = 1 - 2e^{-2} \sim 0.72933$ *if u is in the first cycle C_1 and* $\mathsf{R}[\mathsf{RLA}, 1] \geq 0.735622$ *otherwise.*
- *If $H_u = 2/3$, then the competitive ratio* $\mathsf{R}[\mathsf{RLA}, 2/3] \geq 0.7847$.
- *If $H_u = 1/3$, then competitive ratio* $\mathsf{R}[\mathsf{RLA}, 1/3] \geq 0.7622$.

Now we have all essentials to prove Theorem 2.1.

Proof. From Lemmas 4.1 and 4.2, we know that any u is present in cycle C_1 with probability at most $(2 - 3/e)$.

Consider a node u with $2/3 \leq f_u \leq 1$ and let q_1, q_2, q_3 be the probability that after $\mathsf{DR}[\mathbf{f}, 3]$ and the first modification, $H_u = 1$ and u is in the first cycle C_1, $H_u = 1$ and u is not in C_1, $H_u = 2/3$ respectively. From Lemma 4.4, we get that the final ratio for u should be at least

$$(0.72933q_1 + 0.735622q_2 + (2/3) * 0.7847q_3)/(q_1 + q_2 + (2/3)q_3)$$

Minimizing the above expression subject to (1) $q_1 + q_2 + q_3 = 1$; (2) $0 \leq q_i, 1 \leq i \leq 3$; (3) $q_1 \leq 2 - 3/e$, we get a minimum value of 0.729982 for $q_1 = 2 - 3/e$ and $q_2 = 3/e - 1$.

For any node u with $0 \leq u \leq 2/3$, we know that the ratio is at least $\min(\mathsf{R}[\mathsf{RLA}, 2/3], \mathsf{R}[\mathsf{RLA}, 1/3]) = 0.7622$.

This completes the proof of Theorem 2.1.

□

5 Non-integral arrival rates with stochastic rewards

In this section, we will consider the edge-weighted stochastic matching problem under the known $I.I.D$ model. The setting here is strictly generalized over the previous sections in the following ways. Firstly, it allows an arbitrary arrival rate (say r_v) for each stochastic vertex v. Notice that, $\sum_v r_v = n$ where n is the total number of rounds. Secondly, each $e = (v, u) \in E$ is associated with a value p_e, which indicates the probability that edge $e = (u, v)$ is present when we assign v to u. We assume this process is independent of the stochastic of each v. We will show that the simple non-adaptive algorithm introduced in [7] can be easily extended to this general case. The extension achieves a competitive ratio of $(1 - \frac{1}{e})$. Note that Manshadi *et al.* [13] show that no non-adaptive algorithm can possibly achieve a ratio better than $(1-1/e)$ for the non-integral arrival rates even for the case of all $p_e = 1$. Thus, our algorithm is an optimal non-adaptive algorithm for this model.

We use a similar LP as [8] for the case of non-integral arrival rates. For each $e \in E$, let f_e be the probability that e gets matched in the offline optimal algorithm. Thus we have

$$\text{maximize} \sum_{e \in E} w_e f_e \tag{5.1}$$

$$\text{subject to} \sum_{e \in \partial(u)} f_e p_e \leq 1 \quad \forall u \in U \tag{5.2}$$

$$\sum_{e \in \partial(v)} f_e \leq r_v \quad \forall v \in V \tag{5.3}$$

Our algorithm is summarized in Algorithm 9.

Algorithm 9: SM

1 Construct and solve the LP (5.1). WLOG assume $\{f_e | e \in E\}$ is an optimal solution.

2 When a vertex v arrives, assign v to each of its neighbor u with a probability $\frac{f_{(u,v)}}{r_v}$.

Notice that constraint (5.3) ensures that the step 2 in the algorithm is valid. Let us now prove theorem 2.4.

Proof. Let $B(u,t)$ be the event that u is safe at beginning of round t and $A(u,t)$ to be the event that vertex u is matched during the round t conditioned on $B(u,t)$. From the algorithm, we know

$$\Pr[A(u,t)] \leq \sum_{v \sim u} \frac{r_v}{n} \frac{f_{u,v}}{r_v} p_e \leq \frac{1}{n}, \quad Pr[B(u,t)] = \Pr\left[\bigwedge_{i=1}^{t-1} (\neg A(u,i))\right] \geq \left(1 - \frac{1}{n}\right)^{t-1}$$

Consider an edge $e = (u,v)$ in the graph. Let us lower bound the probability that e gets matched in SM.

$$\begin{aligned}
\Pr[e \text{ is matched}] &= \sum_{t=1}^{n} \Pr[\text{v arrives at } t \text{ and } B(u,t)\ ] * \frac{f_e p_e}{r_v} \\
&\geq \sum_{t=1}^{n} \left(1 - \frac{1}{n}\right)^{t-1} \frac{r_v}{n} \frac{f_e p_e}{r_v} \\
&\geq \left(1 - \frac{1}{e}\right) f_e p_e
\end{aligned}$$

This completes the proof of the claim. □

6 Conclusion and Future Directions

In this paper, we gave improved algorithms for the Edge-Weighted and Vertex-Weighted models. We showed that using dependent rounding, one gets better control over the polytope solutions. Previously, there was gap between the best unweighted algorithm with a ratio of $1 - 2e^{-2}$ due to [8] and the negative result of $1 - e^{-2}$ due to [13]. We took a step towards closing that gap by showing that an algorithm can achieve $0.7299 > 1 - 2e^{-2}$ for both the unweighted and vertex-weighted variants with integral arrival rates. For the variant with edge weights, non-integral arrival rates, and stochastic rewards, we presented a $(1 - 1/e)$-competitive algorithm. This showed that the $0.62 < 1 - 1/e$ bound given in [16] for the adversarial model with stochastic rewards does not extend to the known I.I.D. model.

A natural next question in the edge-weighted setting is if one can do better using an *adaptive* strategy. In the stochastic rewards model with non-intgeral arrival rates setting, the next natural direction is to either improve upon the $\left(1 - \frac{1}{e}\right)$ ratio, or consider a simpler model with integral arrival rates and improve the ratio for this restricted model. Lastly, another direction to pursue is the use of dependent rounding in other models to improve performance.

7 Acknowledgements

The authors would like to thank Aranyak Mehta for his valuable comments, which have significantly helped improve the presentation of this paper.

References

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

[2] Bahman Bahmani and Michael Kapralov. Improved bounds for online stochastic matching. In *Algorithms–ESA 2010*, pages 170–181. Springer, 2010.

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

[4] Jon Feldman, Nitish Korula, Vahab Mirrokni, S Muthukrishnan, and Martin Pál. Online ad assignment with free disposal. In *Internet and network economics*, pages 374–385. Springer, 2009.

[5] Jon Feldman, Aranyak Mehta, Vahab Mirrokni, and S Muthukrishnan. Online stochastic matching: Beating 1-1/e. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 117–126. IEEE, 2009.

[6] Rajiv Gandhi, Samir Khuller, Srinivasan Parthasarathy, and Aravind Srinivasan. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM (JACM)*, 53(3):324–360, 2006.

[7] Bernhard Haeupler, Vahab S. Mirrokni, and Morteza Zadimoghaddam. Online stochastic weighted matching: Improved approximation algorithms. In *Internet and Network Economics*, volume 7090 of *Lecture Notes in Computer Science*, pages 170–181. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-25509-0. doi: 10.1007/978-3-642-25510-6_15. URL http://dx.doi.org/10.1007/978-3-642-25510-6_15.

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

[9] Richard M Karp, Umesh V Vazirani, and Vijay V Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 352–358. ACM, 1990.

[10] Thomas Kesselheim, Klaus Radke, Andreas Tönnis, and Berthold Vöcking. An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions. In *Algorithms–ESA 2013*, pages 589–600. Springer, 2013.

[11] Nitish Korula and Martin Pál. Algorithms for secretary problems on graphs and hypergraphs. In *Automata, Languages and Programming*, pages 508–520. Springer, 2009.

[12] Mohammad Mahdian and Qiqi Yan. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 597–606. ACM, 2011.

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

[14] Aranyak Mehta. Online matching and ad allocation. *Foundations and Trends in Theoretical Computer Science*, 8(4):265–368, 2012. ISSN 1551-305X. doi: 10.1561/0400000057. URL http://dx.doi.org/10.1561/0400000057.

[15] Aranyak Mehta and Debmalya Panigrahi. Online matching with stochastic rewards. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 728–737. IEEE, 2012.

[16] Aranyak Mehta, Bo Waggoner, and Morteza Zadimoghaddam. Online stochastic matching with unequal probabilities. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, ISBN 978-1-61197-374-7*. SIAM, 2015.

8 Appendix

8.1 Proofs in Section 3

8.1.1 Proof of Lemma 3.2

We will prove Lemma 3.2 using the following three Claims.

Claim 8.1. *For a large edge e,* $\mathsf{EW}_1[h]$ (3) *with parameter h achieves a competitive ratio of* $\mathsf{R}[\mathsf{EW}_1, 2/3] = 0.67529 + (1-h) * 0.00446$.

Claim 8.2. *For a small edge e of type Γ_1,* $\mathsf{EW}_1[h]$ (3) *achieves a competitive ratio of* $\mathsf{R}[\mathsf{EW}_1, 1/3] = 0.751066$*, regardless of the value h.*

Claim 8.3. *For a small edge e of type Γ_2,* $\mathsf{EW}_1[h]$ (3) *achieves a competitive ratio of* $\mathsf{R}[\mathsf{EW}_1, 1/3] = 0.72933 + h * 0.040415$.

By setting $h = 0.537815$, the two types of small edges have the same ratio and we get that $\mathsf{EW}_1[h]$ achieves $(\mathsf{R}[\mathsf{EW}_1, 2/3], \mathsf{R}[\mathsf{EW}_1, 1/3]) = (0.679417, 0.751066)$. Thus, this proves Lemma 3.2.

Proof of Claim 8.1

Proof. Consider a large edge $e = (u, v_1)$ in the graph $G_{\mathbf{F}}$. Let $e' = (u, v_2)$ be the other small edge incident to u. Edges e and e' can appear in $[M_1, M_2, M_3]$ in the following three ways.

- α_1: $e \in M_1, e' \in M_2, e \in M_3$.
- α_2: $e' \in M_1, e \in M_2, e \in M_3$.
- α_3: $e \in M_1, e \in M_2, e' \in M_3$.

Notice that the random triple of matchings $[M_1, M_2, M_3]$ is generated by invoking $\mathsf{PM}[\mathbf{F}, 3]$. From the property of $\mathsf{PM}[\mathbf{F}, 3]$, we know that α_i will occur with probability $1/3$ for $1 \le i \le 3$. For α_1 and α_2, we can ignore the second copy of e in M_3 and from Lemma 3.1 we have

$$\Pr[e \text{ is matched} \mid \alpha_1] \ge 0.580831 \text{ and } \Pr[e \text{ is matched} \mid \alpha_2] \ge 0.148499$$

For α_3, we have

$$\begin{aligned}
\Pr[e \text{ is matched} \mid \alpha_3] &= \sum_{t=1}^{n} \frac{1}{n}\left(1-\frac{2}{n}\right)^{t-1} + \sum_{t=1}^{n} \frac{1}{n}\left(\frac{t-1}{n}\right)\left(1-\frac{2}{n}\right)^{t-2} + \sum_{t=1}^{n} \frac{1}{n}\left(\frac{(t-1)(t-2)}{2n^2}\right)\left(1-\frac{2}{n}\right)^{t-3} \\
&\quad + (1-h)\sum_{t=1}^{n} \frac{1}{n}\left(\frac{1}{n^3}\right)\binom{t-1}{3}\left(1-\frac{2}{n}\right)^{t-4} \\
&\ge 0.621246 + (1-h) * 0.00892978
\end{aligned}$$

Hence, we have

$$\Pr[e \text{ is matched}] = \frac{1}{3}\sum_{i=1}^{3} \Pr[e \text{ is matched} \mid \alpha_i] \ge \frac{2}{3}\mathsf{R}[\mathsf{EW}_1, 2/3]$$

where $\mathsf{R}[\mathsf{EW}_1, 2/3] = 0.67529 + (1-h) * 0.00446489$.

□

Proof of Claims 8.2 and 8.3

Proof. Consider a small edge $e = (u, v)$ of type Γ_1. Let e_1 and e_2 be the two other small edges incident to u. For a given triple of matchings $[M_1, M_2, M_3]$, we say e is of type ψ_1 if e appears in M_1 while the other two in the remaining two matchings. Similarly, we define the type ψ_2 and ψ_3 for the case where e appears in M_2 and M_3 respectively. Notice that the probability that e is of type ψ_i, $1 \le i \le 3$ is $1/3$.

Similar to the calculations in the proof of Claim 8.1, we have

$$\Pr[e \text{ is matched} \mid \psi_1] \ge 0.571861, \quad \Pr[e \text{ is matched} \mid \psi_2] \ge 0.144776, \quad \Pr[e \text{ is matched} \mid \psi_3] \ge 0.0344288$$

Therefore we have

$$\Pr[e \text{ is matched}] = \frac{1}{3}\sum_{i=1}^{3}\Pr[e \text{ is matched} \mid \psi_i] \geq \frac{1}{3}\mathsf{R}[\mathsf{EW}_1, 1/3]$$

where $\mathsf{R}[\mathsf{EW}_1, 1/3] = 0.751066$.

Consider a small edge $e = (u, v)$ of type Γ_2, we define type $\beta_i, 1 \leq i \leq 3$, if e appears in M_i while the large edge e' incident to u appears in the remaining two matchings. Similarly, we have

$\Pr[e \text{ is matched} \mid \psi_1] \geq 0.580831$, $\Pr[e \text{ is matched} \mid \psi_2] \geq 0.148499$, $\Pr[e \text{ is matched} \mid \psi_3] \geq h*0.0404154$

Hence, the ratio for a small edge of type Γ_2 is $\mathsf{R}[\mathsf{EW}_1, 1/3] = 0.72933 + h * 0.0404154$.

□

8.1.2 Proof of Lemma 3.3

We will prove Lemma 3.3 using the following two Claims.

Claim 8.4. *For a large edge e, $\mathsf{EW}_2[y_1, y_2]$ (5) achieves a competitive ratio of*

$$\mathsf{R}[\mathsf{EW}_2, 2/3] = \min\left(0.948183 - 0.099895y_1 - 0.025646y_2, 0.871245\right)$$

Claim 8.5. *For a small edge e, $\mathsf{EW}_2[y_1, y_2]$ (5) achieves a competitive ratio of* $\mathsf{R}[\mathsf{EW}_2, 1/3] = 0.4455$*, when* $y_1 = 0.687, y_2 = 1$.

Therefore, by setting $y_1 = 0.687, y_2 = 1$ we get that $\mathsf{R}[\mathsf{EW}_2, 2/3] = 0.8539$ and $\mathsf{R}[\mathsf{EW}_2, 1/3] = 0.4455$, which proves Lemma 3.3.

Proof of Claim 8.4

Proof. Figure 5 shows the two possible configurations for a large edge.

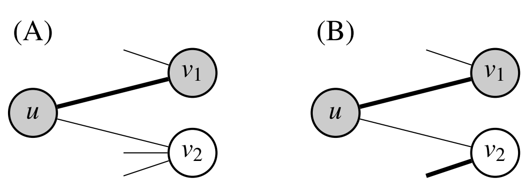

Figure 5: Diagram of configurations for a large edge $e = (u, v_1)$. Thin and Thick lines represent small and large edges respectively.

Consider a large edge $e = (u, v_1)$ with the configuration (A). From $\mathsf{PM}^*[\mathbf{F}, 2][y_1, y_2]$, we know that e will always be in M_1 while $e' = (u, v_2)$ will be in M_1 and M_2 with probability $y_1/3$ and $y_2/3$ respectively.

We now have the following cases

- α_1: $e \in M_1$ and $e' \in M_1$. This happens with probability $y_1/3$ and $\Pr[e \text{ is matched} \mid \alpha_1] \geq 0.432332$.
- α_2: $e \in M_1$ and $e' \in M_2$. This happens with probability $y_2/3$ and $\Pr[e \text{ is matched} \mid \alpha_2] \geq 0.580831$.
- α_3: $e \in M_1$ and $e' \notin M_1, e' \notin M_2$. This happens with probability $(1-y_1/3-y_2/3)$ and $\Pr[e \text{ is matched} \mid \alpha_1] \geq 0.632121$.

Therefore we have

$$\begin{aligned}\Pr[e \text{ is matched}] &= \frac{y_1}{3}\Pr[e \text{ is matched} \mid \alpha_1] + \frac{y_2}{3}\Pr[e \text{ is matched} \mid \alpha_2] + (1 - \frac{y_1}{3} - \frac{y_2}{3})\Pr[e \text{ is matched} \mid \alpha_3] \\ &\geq \frac{2}{3}(0.948183 - 0.099895y_1 - 0.025646y_2)\end{aligned}$$

Consider the configuration (B). From $\mathsf{PM}^*[\mathbf{F}, 2][y_1, y_2]$, we know that e will always be in M_1 and $e' = (u, v_2)$ will always be in M_2. Thus we have

$$\Pr[e \text{ is matched}] = \Pr[e \text{ is matched} \mid \alpha_2] = \frac{2}{3} * 0.871245$$

Hence, this completes the proof of Claim 8.4.

□

Proof of Claim 8.5

Proof. Figure 6 shows all possible configurations for a small edge.

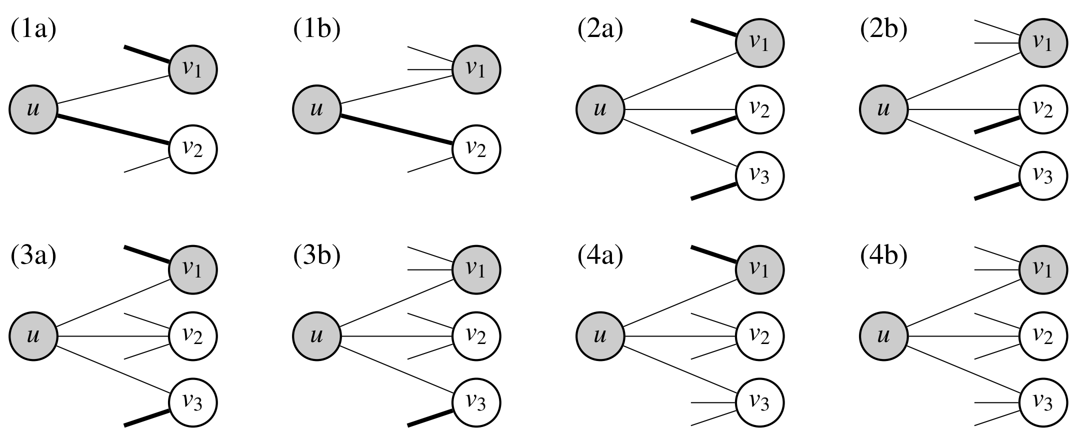

Figure 6: Diagram of configurations for a small edge $e = (u, v_1)$. Thin and Thick lines represent small and large edges respectively.

Similar to the proof of Claim 8.4, we will do a case-by-case analysis on the various configurations. Let $e_i = (u, v_i)$ for $1 \leq i \leq 3$ and $\mathcal{E}$ be the event that e_1 gets matched. For a given e_i, denote $e_i \in M_0$ if $e_i \notin M_1, e_i \notin M_2$.

- $(1a)$. Observe that $e_1 \in M_2$ and $e_2 \in M_1$. Thus we have $\Pr[\mathcal{E}] = \frac{1}{3} * 0.44550$.
- $(1b)$. Observe that we have two cases: $\{\alpha_1 : e_2 \in M_1, e_1 \in M_1\}$ and $\{\alpha_2 : e_2 \in M_1, e_1 \in M_2\}$. Case α_1 happens with probability $y_1/3$ and the conditional probability is $\Pr[\mathcal{E} \mid \alpha_1] = 0.432332$. Case α_2 happens with probability $y_2/3$ and the conditional is $\Pr[\mathcal{E} \mid \alpha_2] = 0.148499$. Thus we have

$$\Pr[\mathcal{E}] = y_1/3 * \Pr[\mathcal{E} \mid \alpha_1] + y_2/3 * \Pr[\mathcal{E} \mid \alpha_2] \geq \frac{1}{3}(0.432332 y_1 + 0.148499 y_2)$$

- $(2a)$. Observe that $e_1 \in M_2, e_2 \in M_2, e_3 \in M_2$. $\Pr[\mathcal{E}] = \frac{1}{3} * 0.601704$
- $(2b)$. Observe that we have two cases: $\{\alpha_1 : e_1 \in M_1, e_2 \in M_2, e_3 \in M_2\}$ and $\{\alpha_2 : e_1 \in M_2, e_2 \in M_2, e_3 \in M_2\}$. Case α_1 happens with probability $y_1/3$ and the conditional is $\Pr[\mathcal{E} \mid \alpha_1] = 0.537432$. Case α_2 happens with probability $y_2/3$ and conditional is $\Pr[\mathcal{E} \mid \alpha_2] = 0.200568$. Thus we have

$$\Pr[\mathcal{E}] = y_1/3 * \Pr[\mathcal{E} \mid \alpha_1] + y_2/3 * \Pr[\mathcal{E} \mid \alpha_2] \geq \frac{1}{3}(0.537432 y_1 + 0.200568 y_2)$$

- $(3a)$. Observe that we have three cases: $\{\alpha_1 : e_1 \in M_2, e_2 \in M_1, e_3 \in M_2\}$, $\{\alpha_2 : e_1 \in M_2, e_2 \in M_2, e_3 \in M_2\}$ and $\{\alpha_2 : e_1 \in M_2, e_2 \in M_0, e_3 \in M_2\}$. Case α_1 happens with probability $y_1/3$ and conditional is $\Pr[\mathcal{E} \mid \alpha_1] = 0.13171$. Case α_2 happens with probability $y_2/3$ and conditional is $\Pr[\mathcal{E} \mid \alpha_2] = 0.200568$. Case α_3 happens with probability $(1 - y_1/3 - y_2/3)$ and conditional is $\Pr[\mathcal{E} \mid \alpha_3] = 0.22933$.

  Similarly, we have

$$\begin{aligned}\Pr[\mathcal{E}] &= y_1/3 * \Pr[\mathcal{E} \mid \alpha_1] + y_2/3 * \Pr[\mathcal{E} \mid \alpha_2] + (1 - y_1/3 - y_2/3) * \Pr[\mathcal{E} \mid \alpha_3] \\ &\geq \frac{1}{3}(0.13171 y_1 + 0.200568 y_2 + (3 - y_1 - y_2)0.22933)\end{aligned}$$

- $(3b)$. Observe that we have six cases.
  - $\alpha_1 : e_1 \in M_1, e_2 \in M_1, e_3 \in M_2$. $\Pr[\alpha_1] = y_1^2/9$ and $\Pr[\mathcal{E} \mid \alpha_1] = 0.4057$.
  - $\alpha_2 : e_1 \in M_1, e_2 \in M_2, e_3 \in M_2$. $\Pr[\alpha_2] = y_1 y_2/9$ and $\Pr[\mathcal{E} \mid \alpha_2] = 0.5374$.
  - $\alpha_3 : e_1 \in M_1, e_2 \in M_0, e_3 \in M_2$. $\Pr[\alpha_3] = y_1/3(1 - y_1/3 - y_2/3)$ and $\Pr[\mathcal{E} \mid \alpha_3] = 0.58083$.
  - $\alpha_4 : e_1 \in M_2, e_2 \in M_1, e_3 \in M_2$. $\Pr[\alpha_4] = y_1 y_2/9, \Pr[\mathcal{E} \mid \alpha_4] = 0.1317$.
  - $\alpha_5 : e_1 \in M_2, e_2 \in M_2, e_3 \in M_2$. $\Pr[\alpha_5] = y_2^2/9, \Pr[\mathcal{E} \mid \alpha_5] = 0.2006$.
  - $\alpha_6 : e_1 \in M_2, e_2 \in M_0, e_3 \in M_2$. $\Pr[\alpha_6] = y_2/3(1 - y_1/3 - y_2/3)/3$ and $\Pr[\mathcal{E} \mid \alpha_6] = 0.22933$.

  Therefore we have

$$\Pr[\mathcal{E}] \geq \frac{1}{3}\Big(0.135241 y_1^2 + 0.223033 y_1 y_2 + 0.066856 y_2^2 + y_1(3 - y_1 - y_2)0.193610 + y_2(3 - y_1 - y_2)0.076443\Big)$$

- $(4a)$. Observe that we have following six cases.
  - $\alpha_1 : e_1 \in M_2, e_2 \in M_1, e_3 \in M_1$. $\Pr[\alpha_1] = y_1^2/9$ and $\Pr[\mathcal{E} \mid \alpha_1] = 0.08898$.

– $\alpha_2 : e_1 \in M_2, e_2 \in M_2, e_3 \in M_2$. $\Pr[\alpha_2] = y_2^2/9$ and $\Pr[\mathcal{E}\,|\alpha_2] = 0.2006$.

– $\alpha_3 : e_1 \in M_2, e_2 \in M_0, e_3 \in M_0$. $\Pr[\alpha_3] = (1 - y_1/3 - y_1/3)^2$, and $\Pr[\mathcal{E}\,|\alpha_3] = 0.2642$.

– α_4: $e_1 \in M_2$ while either $e_2 \in M_1, e_3 \in M_2$ or $e_2 \in M_2, e_3 \in M_1$. $\Pr[\alpha_2] = 2y_1y_2/9$ and $\Pr[\mathcal{E}\,|\alpha_4] = 0.1317$.

– α_5: $e_1 \in M_2$ while either $e_2 \in M_1, e_3 \in M_0$ or $e_2 \in M_0, e_3 \in M_1$. $\Pr[\alpha_5] = 2y_1/3(1-y_1/3-y_2/3)$ and $\Pr[\mathcal{E}\,|\alpha_5] = 0.14849$.

– α_6: $e_1 \in M_2$ while either $e_2 \in M_2, e_3 \in M_0$ or $e_2 \in M_0, e_3 \in M_2$. $\Pr[\alpha_5] = 2y_2/3(1-y_1/3-y_2/3)$ and $\Pr[\mathcal{E}\,|\alpha_6] = 0.22933$.

Therefore we have

$$\begin{aligned}\Pr[\mathcal{E}] \quad \geq \quad & \frac{1}{3}\Big(0.029661y_1^2 + 2 * 0.043903y_1y_2 + 0.066856y_2^2 \\ & +2y_1(3 - y_1 - y_2)0.0494997 + 2y_2(3 - y_1 - y_2)(0.076443) + (3 - y_1 - y_2)^2 0.0880803\Big)\end{aligned}$$

$(4b)$. Observe that in this configuration, we have additional six cases to the ones discussed in $(4a)$. Let α_i be the cases defined in $(4a)$ for each $1 \leq i \leq 6$. Notice that each $\Pr[\alpha_i]$ has a multiplicative factor of $y_2/3$. Now, consider the six new cases.

- $\beta_1 : e_1 \in M_1, e_2 \in M_1, e_3 \in M_1$. $\Pr[\alpha_1] = y_1^3/27$ and $\Pr[\mathcal{E}\,|\alpha_1] = 0.3167$.
- $\beta_2 : e_1 \in M_1, e_2 \in M_2, e_3 \in M_2$. $\Pr[\alpha_2] = y_1y_2^2/27$ and $\Pr[\mathcal{E}\,|\alpha_2] = 0.5374$.
- $\beta_3 : e_1 \in M_1, e_2 \in M_0, e_3 \in M_0$. $\Pr[\alpha_3] = y_1/3 * (1 - y_1/3 - y_2/3)^2$ and $\Pr[\mathcal{E}\,|\alpha_3] = 0.632$.
- β_4: $e_1 \in M_1$ and either $e_2 \in M_1, e_3 \in M_2$ or $e_2 \in M_2, e_3 \in M_1$. $\Pr[\alpha_2] = 2y_1^2y_2/27$ and $\Pr[\mathcal{E}\,|\alpha_4] = 0.4057$.
- β_5: $e_1 \in M_1$ and either $e_2 \in M_1, e_3 \in M_0$ or $e_2 \in M_0, e_3 \in M_1$. $\Pr[\alpha_5] = 2y_1^2/9 * (1 - y_1/3 - y_2/3)$ and $\Pr[\mathcal{E}\,|\alpha_5] = 0.4323$.
- β_6: $e_1 \in M_1$ and either $e_2 \in M_2, e_3 \in M_0$ or $e_2 \in M_0, e_3 \in M_2$. $\Pr[\alpha_5] = 2y_1y_2/9 * (1 - y_1/3 - y_2/3)$ and $\Pr[\mathcal{E}\,|\alpha_6] = 0.58083$.

Hence, we have

$$\begin{aligned}\Pr[\mathcal{E}] \quad \geq \quad & \frac{1}{3}\Big(0.632y_1 - 0.133133y_1^2 + 0.0093y_1^3 + 0.264241y_2 \\ & -0.11127y_1y_2 + 0.01170y_1^2y_2 - 0.0232746y_2^2 + 0.00488y_1y_2^2 + 0.00068y_2^3\Big)\end{aligned}$$

Setting $y_1 = 0.687$, $y_2 = 1$, we get that the competitive ratio for a small edge is 0.44550. The bottleneck cases are configurations $(1a)$ and $(1b)$. □

8.2 Proofs in section 4

8.2.1 The Second Modification to H

Figure 7 describes the various modifications applied to **H** vector. The values on top of the edge, denote the new values. Cases (11) and (12) help improve upon the WS described in Figure 2.

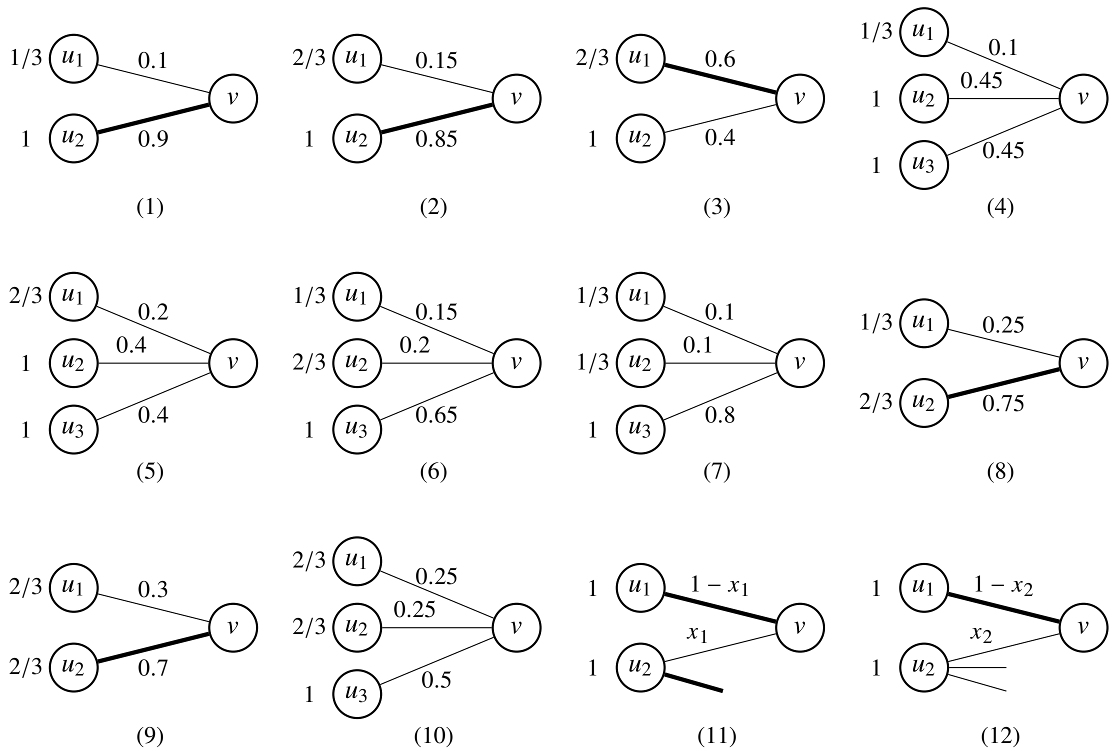

Figure 7: Illustration for second modification to **H**. The value assigned to each edge represents the value after the second modification. Here, $x_1 = 0.2744$ and $x_2 = 0.15877$.

8.2.2 Proof of Lemma 4.2

The proof of Lemma 4.2 follows from the following Claims:

Claim 8.6. *Breaking cycles will not change the value H_w for any $w \in U \cup V$.*

Claim 8.7. *After breaking a cycle of type C_2, the vertices u_1, u_2, v_1, and v_2 can never be part of any length four cycle.*

Claim 8.8. *When all length four cycles are of type C_1 or C_3, breaking exactly one cycle of type C_3 cannot create a new cycle of type C_1.*

Proof of Claim 8.6

Proof. As shown in Figure 3, we increase and decrease edge values f_e in such a way that their sums H_w at any vertex w will be preserved. □

Notice that C_2 cycles can be freely broken without creating new C_1 cycles. After removing all cycles of type C_2, removing a single cycle of type C_3 cannot create any cycles of type C_1. Hence, Algorithm 7 removes all C_2 and C_3 cycles without creating any new C_1 cycles.

Proof of Claim 8.7

Proof. Consider the structure after breaking a cycle of type C_2. Note that the edge (u_2, v_2) has been permanently removed and hence, these four vertices together can never be part of a cycle of length four. The vertices u_1 and v_1 have $H_{u_1} = 1$ and $H_{v_1} = 1$ respectively. So they cannot have any other edges and therefore cannot appear in any length four cycle. The vertices u_2 and v_2 can each have one additional edge, but since the edge (u_2, v_2) has been removed, they can never be part of any cycle with length less than six.

□

Proof of Claim 8.8

Proof. First, we note that since no edges will be added during this process, we cannot create a new cycle of length four or join with a cycle of type C_1. Therefore, the only cycles which could be affected are of type C_3. However, every cycle c of type C_3 falls into one of two cases

Case 1: c is the cycle we are breaking.

In this case, c cannot become a cycle of type C_1 since we remove two of its edges and break the cycle.

Case 2: c is not the cycle we are breaking.

In this case, c can have at most one of its edges converted to a $2/3$ edge. Let c' be the length four cycle we are breaking. Note that c and c' will differ by at least one vertex. When we break c', the two edges which are converted to $2/3$ will cover all four vertices of c'. Therefore, at most one of these edges can be in c. □

Note that breaking one cycle of type C_3 could create cycles of type C_2, but these cycles are always broken in the next iteration, before breaking another cycle of type C_3.

8.2.3 Proof of Lemma 4.4

When $H_u = 1$ and u is in the cycle C_1, [8] show that the competitive ratio of u is $1 - 2e^{-2}$. Hence, for the remaining cases, we use the following Claims.

Claim 8.9. *If $H_u = 1$ and u is not in C_1, then we have* $\mathsf{R}[\mathsf{RLA}, 1] \geq 0.735622$.

Claim 8.10. $\mathsf{R}[\mathsf{RLA}, 2/3] \geq 0.7870$.

Claim 8.11. $\mathsf{R}[\mathsf{RLA}, 1/3] \geq 0.8107$.

Recall that B_u is the event that among the n random lists, there exists a list starting with u and G_u^v is the event that among the n lists, there exist successive lists such that (1) all start with some u' which are different from u but are neighbors of v; and (2) they ensure u will be matched.

Notice that P_u is the probability that u gets matched in $\mathsf{RLA}[\mathbf{H}']$. For each u, we compute $\Pr[B_u]$ and $\Pr[G_u^v]$ for all possibilities of $v \sim u$ and using Lemma 4.3 we get P_u. First, we discuss two different ways to calculate $\Pr[G_u^v]$. For some cases, we use a direct calculation, while for the rest we use the Markov-chain approximation method.

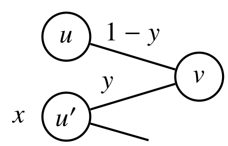

Figure 8: Case 1 in calculation of $\Pr[G_u^v]$

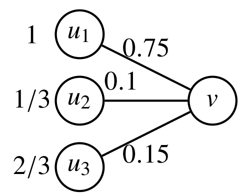

Figure 9: Case 2 in calculation of $\Pr[G_u^v]$

Two ways to compute the value $\Pr[G_u^v]$

1. **Case 1**: Consider the case when v has two neighbors as shown in Figure 8. Assume v has two neighbors u and u' and after modifications, $H'_{(u',v)} = y$, $H'_{(u,v)} = 1-y$ and $H'_{u'} = x$. Thus, the second certificate event G_u^v corresponds to the event (1) a list starting with u' comes at some time $1 \le i < n$; (2) the list $\mathcal{R}_v = (u', u)$ comes for a second time at some j with $i < j \le n$. Note that the arrival rate of a list starting with u' is $H'_{u'} = x/n$ and the rate of list $\mathcal{R}_v = (u', u)$ is y/n. Therefore we have

$$\begin{aligned} \Pr[G_u^v] &= \sum_{i=1}^{n-1} \left(x/n(1-x/n)^{(i-1)}(1-(1-y/n)^{(n-i)} \right) & (8.1) \\ &\sim \frac{x - e^{-y}x + (-1+e^{-x})y}{x-y} \quad (\text{if } x \neq y) & (8.2) \\ &\sim 1 - e^{-x}(1+x) \quad (\text{if } x = y) & (8.3) \end{aligned}$$

2. **Case 2**: Consider the case when v has three neighbors. The value $\Pr[G_u^v]$ is approximated using the Markov Chain method, similar to [8]. Let us use the following example to illustrate the method.

   Consider the following case as shown in Figure 9 (v has three neighbors u, u_1 and u_2 with $H_u = 1, H_{u_1} = 1/3$ and $H_{u_2} = 2/3$). Recall that after modifications, we have $H'_{(u_1,v)} = b = 0.1, H'_{(u_2,v)} = c = 0.15$ and $H'_{(u,v)} = d = 0.75$. We simulate the process of u getting matched resulting from several successive random lists starting from either u_1 or u_2 by an n-step Markov Chain as follows. We have 5 states: $s_1 = (0,0,0), s_2 = (0,1,0), s_3 = (0,0,1), s_4 = (0,1,1)$ and $s_5 = (1,*,*)$ and the three numbers in each triple correspond to u, u_1 and u_2 being matched(or not) respectively. State s_5 corresponds to u being matched; the matched status of u_1 and u_2 is irrelevant. The chain initially starts in s_1 and the probability of being in state s_5 after n steps gives an approximation to $\Pr[G_u^v]$. The one-step transition probability matrix M is shown as follows.

$$
\begin{aligned}
M_{1,2} &= \frac{b}{n}, M_{1,3} = \frac{c+1/3}{n}, M_{1,1} = 1 - M_{1,2} - M_{1,3} \\
M_{2,4} &= \frac{c+1/3}{n} + \frac{bc}{(c+d)n}, M_{2,5} = \frac{bd}{(c+d)n}, \\
M_{2,3} &= 1 - M_{2,4} - M_{2,5} \\
M_{3,4} &= \frac{b}{n} + \frac{cb}{(b+d)n}, M_{3,5} = \frac{cd}{(b+d)n} \\
M_{3,3} &= 1 - M_{3,4} - M_{3,5} \\
M_{4,5} &= \frac{b+c}{n}, M_{4,4} = 1 - M_{4,5} \\
M_{5,5} &= 1 \\
M_{i,j} &= 0 \text{ for all other i, j}
\end{aligned}
$$

Notice that $M_{1,3} = \frac{c+1/3}{n}$ and not $\frac{2}{3n}$ since after modifications, the arrival rate of a list starting with u_2 decreases correspondingly.

Let us now prove the three Claims 8.9, 8.10 and 8.11. Here we give the explicit analysis for the case when $H_u = 1$. For the remaining cases, similar methods can be applied. Hence, we omit the analysis and only present the related computational results which leads to the conclusion.

Proof of Claim 8.9

Proof. Notice that u is not in the cycle C_1 and thus Lemma 4.3 can be used. Figure 10 describes all possible cases when a node $u \in U$ has $H_u = 1$. (We ignore all those cases when $H_u < 1$, since they will not appear in the WS.)

Let v_1 and v_2 be the two neighbors of u with $H_{(u,v_1)} = 2/3$ and $H_{(u,v_2)} = 1/3$. In total, there are 4×10 combinations, where v_1 is chosen from some $\alpha_i, 1 \le i \le 4$ and v_2 is chosen from some $\beta_i, 1 \le i \le 9$. For $H_u = 1$, we need to find the worst combination among these such that the value P_u is minimized. We can find this WS using the Lemma 4.3.

For each type of α_i, β_j, we compute the values it will contribute to the term $(1 - B_u)\prod_{v\sim u}(1 - \Pr[G_u^v])$. For example, assume v_1 is of type α_1, denoted by $v_1(\alpha_1)$. It contributes $e^{-0.9}$ to the term $(1 - B_u)$ and $(1 - \Pr[G_u^{v_1}])$ to $\prod_{v\sim u}(1-\Pr[G_u^v])$, thus the total value it contributes is $\gamma(v_1,\alpha_1) = e^{-0.9}(1-\Pr[G_u^{v_1}])$. Similarly, we can compute all $\gamma(v_1,\alpha_i)$ and $\gamma(v_2,\beta_j)$. Let $i^* = \operatorname{argmax}_i \gamma(v_1,\alpha_i)$ and $j^* = \operatorname{argmax}_j \gamma(v_2,\beta_j)$. The WS is for the combination $\{v_1(\alpha_{i^*}), v_2(\beta_{j^*})\}$ and the resulting value of P_u and $\mathsf{R}[\mathsf{RLA}, 1]$ is as follows:

$$\mathsf{P}_u = 1 - \gamma(v_1,\alpha_{i^*})\gamma(v_2,\beta_{j^*})$$

$$\mathsf{R}[\mathsf{RLA}, 1] = \mathsf{P}_u / H_u = \mathsf{P}_u$$

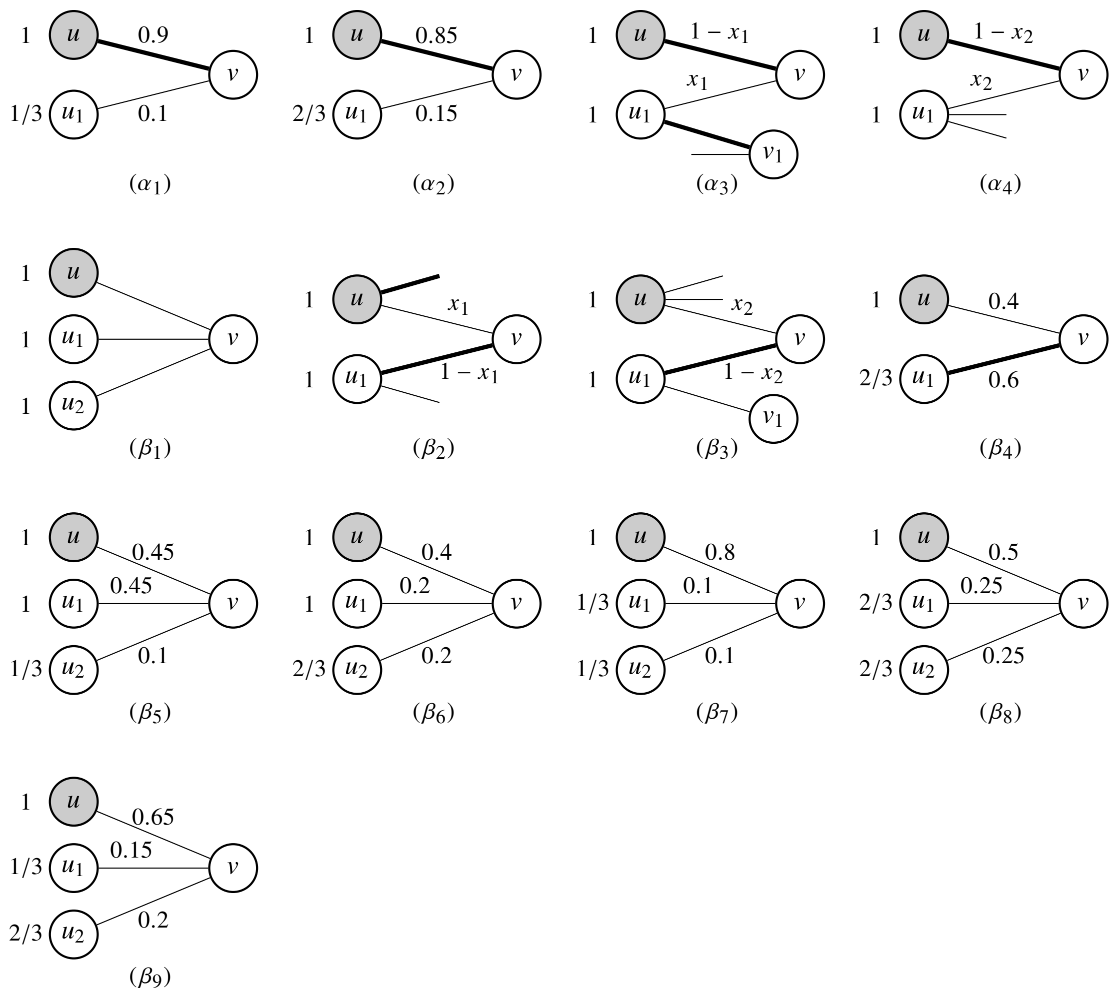

Figure 10: Vertex-weighted $H_u = 1$ cases. The value assigned to each edge represents the value after the second modification. No value indicates no modification. Here, $x_1 = 0.2744$ and $x_2 = 0.15877$.

Here is a list of $\gamma(v_1, \alpha_i)$ and $\gamma(v_2, \beta_j)$, for each $1 \le i \le 4$ and $1 \le j \le 9$.

- α_1: We have $\Pr[G_u^v] = 1 - e^{-0.1} * 1.1$ and $\gamma(v, \alpha_1) = e^{-0.1} * 1.1 * e^{-0.9} = 0.404667$.
- α_2: $\Pr[G_u^v] \ge 1 - e^{-0.15} * 1.15$ and $\gamma(v, \alpha_2) \le 0.423$.

  Notice that after modifications, $H'_{u_1} \ge 0.15$. Hence, we use this and Equation 8.1 to compute the lower bound of $\Pr[G_u^v]$.
- α_3: $\Pr[G_u^v] \ge 0.0916792$ and $\gamma(v, \alpha_3) \le 0.439667$.

  Notice that for any large edge e incident to a node u with $H_u = 1$ (before modification), we have after modification, $H'_e \ge 1 - 0.2744 = 0.7256$. Thus we have $H'_{(u_1, v_1)} \ge 0.7256$ and $H'_{u_1} \ge 1$. From Equation 8.1, we get $\Pr[G_u^v] \ge 0.0916792$.
- α_4: $\Pr[G_u^v] \ge 0.0307466$ and $\gamma(v, \alpha_4) \le 0.417923$.

Notice that for any small edge e incident to a node u with $H_u = 1$ (before modification), we have after modification, $H'_e \geq 0.15877$. Thus, we have $H'_{u_1} \geq 3 * 0.15877$.

- β_1: $\Pr[G_u^v] = 0.1608$ and $\gamma(v, \beta_1) = 0.601313$.
- β_2: $\Pr[G_u^v] \geq 0.208812$ and $\gamma(v, \beta_2) \leq 0.601313$.

  After modifications, we have $H'_{(u_1,v_1)} \geq 0.2744$ and thus we get $H'_{u_1} \geq 1$.
- β_3: $\Pr[G_u^v] \geq 0.251611$ and $\gamma(v, \beta_2) \leq 0.63852$.

  After modifications, we have $H'_{(u_1,v_1)} \geq 0.2744$ and thus we get $H'_{u_1} \geq 1 - 0.15877 + 0.2744$.
- β_4: $\Pr[G_u^v] = 0.121901$ and $\gamma(v, \beta_4) = 0.588607$.
- β_5: $\Pr[G_u^v] = 0.1346$ and $\gamma(v, \beta_5) = 0.551803$.
- β_6: $\Pr[G_u^v] \geq 0.1140$ and $\gamma(v, \beta_6) \leq 0.593904$.
- β_7: $\Pr[G_u^v] = 0.0084$ and $\gamma(v, \beta_7) = 0.4455$.
- β_8: $\Pr[G_u^v] \geq 0.0397$ and $\gamma(v, \beta_8) \leq 0.582451$.
- β_9: $\Pr[G_u^v] \geq 0.0230$ and $\gamma(v, \beta_9) \leq 0.510039$.

Using the computed values above, let us compute the ratio of a node u with $H_u = 1$.

- If u has three neighbors, then the WS configuration is when each of the three neighbors of u is of type β_3. This is because, the value of $\gamma(v, \beta_3)$ is the largest. The resultant ratio is 0.73967.
- If u has two neighbors, then the WS configuration is when one of the neighbor is of type β_1 (or β_2) and the other is of type α_3. The resultant ratio is 0.735622.

□

Proof of Claim 8.10

Proof. The proof is similar to that of Claim 8.9. The Figure 11 shows all possible configurations of a node u with $H_u = 2/3$. Note that the WS cannot have $F(v) < 1$ and hence we omit them here. For a neighbor v of u, if $H_{(u,v)} = 2/3$, then v is in one of $\alpha_i, 1 \leq i \leq 3$; if $H_{(u,v)} = 1/3$, then v is in one of $\beta_i, 1 \leq i \leq 8$. We now list the values $\gamma(v, \alpha_i)$ and $\gamma(v, \beta_j)$, for each $1 \leq i \leq 3$ and $1 \leq j \leq 8$.

- α_1: We have $\Pr[G_u^v] = 1 - e^{-0.25} * 1.25$ and $\gamma(v, \alpha_1) = e^{-0.25} * 1.25 * e^{-0.75} = 0.459849$.
- α_2: We have $\Pr[G_u^v] \geq 0.0528016$ and $\gamma(v, \alpha_1) \leq 0.470365$.
- α_3. We have $\Pr[G_u^v] \geq 0.13398$ and $\gamma(v, \alpha_3) \leq 0.475282$.
- β_1: We have $\Pr[G_u^v] = 1 - e^{-0.7} * 1.7$ and $\gamma(v, \beta_1) = 0.625395$.
- β_2: We have $\Pr[G_u^v] \geq 0.226356$ and $\gamma(v, \beta_2) \leq 0.665882$.
- β_3: We have $\Pr[G_u^v] \geq 0.1819$ and $\gamma(v, \beta_3) \leq 0.669804$.
- β_4: We have $\Pr[G_u^v] \geq 0.1130$ and $\gamma(v, \beta_4) \leq 0.635563$.

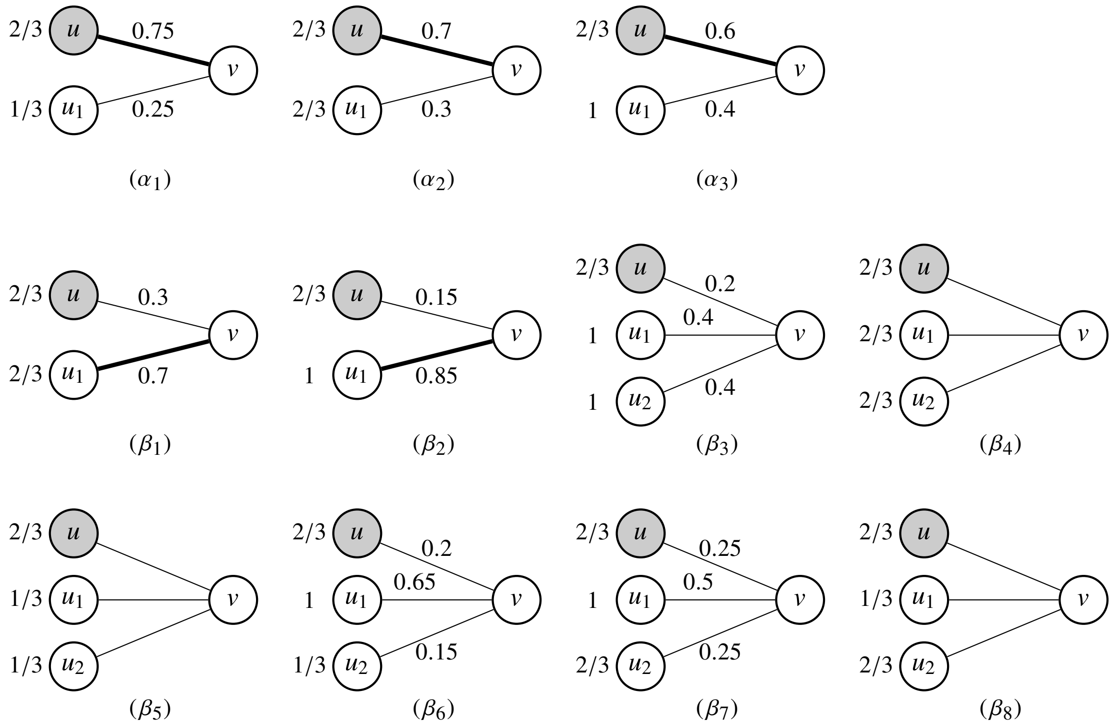

Figure 11: Vertex-weighted $H_u = 2/3$ cases. The value assigned to each edge represents the value after the second modification. No value indicates no modification.

- β_5: We have $\Pr[G_u^v] \geq 0.0587$ and $\gamma(v, \beta_5) \leq 0.674471$.
- β_6: We have $\Pr[G_u^v] \geq 0.1688$ and $\gamma(v, \beta_6) \leq 0.680529$.
- β_7: We have $\Pr[G_u^v] \geq 0.1318$ and $\gamma(v, \beta_7) \leq 0.676155$.
- β_8: We have $\Pr[G_u^v] \geq 0.0587$ and $\gamma(v, \beta_8) \leq 0.674471$.

Hence, the WS structure is when u is such that $H_u = 2/3$ and has one neighbor of type α_3. The resultant ratio is 0.7870.

□

Proof of Claim 8.11

Proof. The Figure 12 shows the possible configurations of a node u with $H_u = 1/3$. Again, we omit those cases where $H_v < 1$.

We now list the values $\gamma(v, \alpha_i)$, for each $1 \leq i \leq 8$.

- α_1: We have $\Pr[G_u^v] = 1 - e^{-0.75} * 1.75$ and $\gamma(v, \alpha_1) = 0.643789$.
- α_2: We have $\Pr[G_u^v] \geq 0.282256$ and $\gamma(v, \alpha_2) \leq 0.649443$.
- α_3: We have $\Pr[G_u^v] \geq 0.1935$ and $\gamma(v, \alpha_3) \leq 0.729751$.

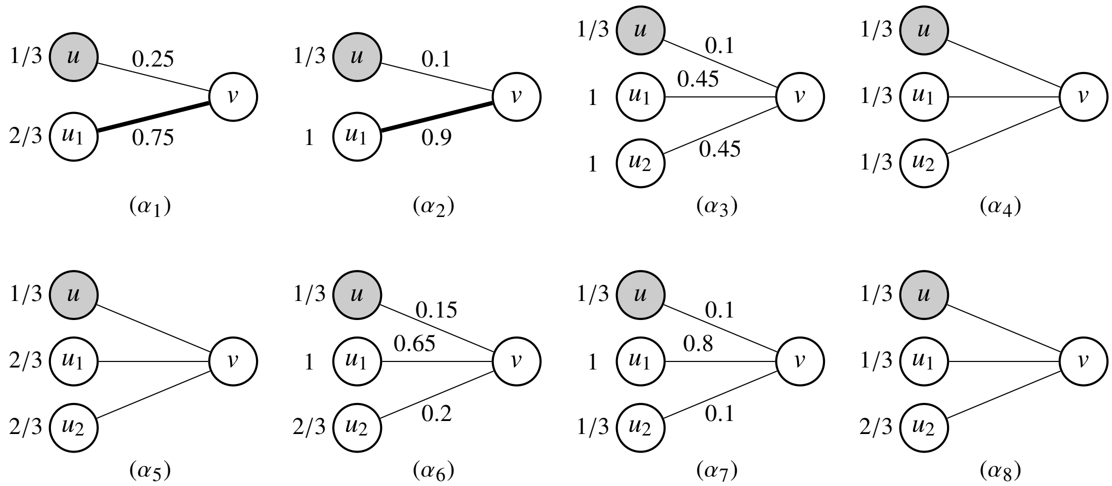

Figure 12: Vertex-weighted $H_u = 1/3$ cases. The value assigned to each edge represents the value after the second modification. No value indicates no modification.

- α_4: We have $\Pr[G_u^v] \geq 0.0587$ and $\gamma(v, \alpha_4) \leq 0.674471$.
- α_5: $\gamma(v, \alpha_5) \leq 0.674471$.
- α_6: We have $\Pr[G_u^v] \geq 0.1546$ and $\gamma(v, \alpha_6) \leq 0.727643$.
- α_7: We have $\Pr[G_u^v] \geq 0.1938$ and $\gamma(v, \alpha_7) \leq 0.72948$.
- α_8: $\gamma(v, \alpha_8) \leq 0.674471$.

Hence, the WS for node u with $H_u = 1/3$ is when u has one neighbor of type α_3. The resultant ratio is 0.8107.

$\square$