Maximum Matching in the Online Batch-Arrival Model

Euiwoong Lee and Sahil Singla

Computer Science Department,
Carnegie Mellon University, USA
{euiwoonl,ssingla}@cs.cmu.com

Abstract. Consider a *two-stage matching* problem, where edges of an input graph are revealed in two stages (batches) and in each stage we have to immediately and irrevocably extend our matching using the edges from that stage. The natural greedy algorithm is half competitive. Even though there is a huge literature on online matching in adversarial *vertex arrival model*, no positive results were previously known in adversarial *edge arrival model*.
For two-stage bipartite matching problem, we show that the optimal competitive ratio is exactly $2/3$ in both the fractional and the randomized-integral models. Furthermore, our algorithm for fractional bipartite matching is *instance optimal*—achieves the best competitive ratio for any *given* first stage graph. We also study natural extensions of this problem to general graphs and to s stages, and present randomized-integral algorithms with competitive ratio $\frac{1}{2} + 2^{-O(s)}$.
Our algorithms use a novel $\mathbf{LP}$ and combine graph decomposition techniques with online primal-dual analysis.

Keywords: Online Algorithms; Matching; Primal-Dual Analysis; Edmonds-Gallai Decomposition; Competitive Ratio; Semi-Streaming

1 Introduction

The field of online algorithms has had tremendous success in modeling optimization problems under uncertain input (see books [3,11,26]). The framework involves an underlying optimization problem (e.g., max matching), where the input arrives in stages (e.g., edges or vertices of a graph) and we have to make immediate and irrevocable decisions at the end of each stage (e.g., whether to match the vertex or edge). The goal is to design online algorithms with high *competitive ratio*—for the worst possible input instance the expected ratio of the online algorithm to the best algorithm in hindsight.

Most prior works on competitive analysis have only considered "single" element arrival in each stage. Since the amount of information revealed in each stage is small, the interesting regime is when the number of stages are *large* (linear in input size). Although powerful, this model often becomes too pessimistic and algorithms with good competitive ratios cannot be obtained. Here we consider the alternate *online batch arrival model*, where a "large" portion of the input (batch) arrives in each stage and the algorithm makes an irrevocable decision at the end of each stage. For a single stage this model captures the offline optimization problem and for a linear number of stages it captures the standard online model. Can we obtain competitive ratios better than the standard online model for a "small" number of stages, say even two stages?

The motivation to study online batch arrival model is based on the fact that in many scenarios that involve decision making under uncertainty, it is conceivable that instead of making an irrevocable decision for each arrival, the decision-maker prefers to gather some information for a certain amount of time and make a collective decision based on it. Indeed, multistage and especially two-stage robust/ stochastic optimization problems have been actively studied in both computer science and operations research (see recent paper of Golovin et al. [14] and references therein, or a survey of Swamy and Shmoys [30]).

In this paper we study the online matching problem in the batch arrival model, where edges of a graph arrive in s stages/ batches. For the basic online model where edges arrive one-by-one, its competitive ratio is not well understood. In particular, it is still open that the competitive ratio is strictly bigger than $1/2$ or not, which is achieved by a simple greedy algorithm. We prove that in our online batch arrival model, the competitive ratio is strictly bigger than $1/2$ for any fixed number of stages. In particular, when $s = 2$, the tight competitive ratio is exactly $2/3$. For $s = 2$, we also present a new LP relaxation that guarantees *instance optimality*, which means that our algorithm's decision in the first stage is optimal over the arbitrary choice of the second batch.

Our algorithms use classical tools from matching theory, such as Edmonds-Gallai decomposition and TDIness of the matching polytope, combined with carefully chosen parameters to prove large competitive ratios (some inequalities are computer-assisted). These positive results imply that our online batch arrival model allows interesting algorithmic ideas to work, and may not be as pessimistic as the original online model.

1.1 Our Model and Results

Online matching in the *vertex-arrival model* started with the seminal work of Karp et al. [19]. In this setting, vertices on one side of a bipartite graph are revealed one-by-one along with their edges to the other side, and the problem is to immediately and irrevocably match this vertex. Since this problem occupies a central position in online algorithms and has many applications in online advertisement, many of its variants have been studied in great depth (see survey [26]). This includes problems like AdWords [28,4,5], vertex-weighted [1,6], edge-weighted [22,16], stochastic matching [10,27,25,29], random vertex arrival [13,24,18], and vertex arrival on both sides [32,2].

Even though there is a long list of work in the vertex arrival model, no non-trivial algorithms are known in the equally natural *edge-arrival model*. Here edges of a (bipartite) graph are revealed one-by-one and the online problem is to immediately and irrevocably decide whether to *pick* the revealed edge into a matching. The best known algorithm is greedy, which picks an edge whenever possible and is half-competitive. Even when edges incident to a vertex are revealed together, it already captures online bipartite matching with vertex arrival on both sides, where nothing more than half is known [8,32].

The *two-stage (fractional) bipartite matching problem* is formally defined as follows. Edges of a bipartite graph $G = ((U_1, U_2), E)$ are revealed in two stages: Stage 1 reveals a subgraph $G^{(1)} = ((U_1, U_2), E^{(1)})$ of G and we need to immediately and irrevocably decide which of its edges to pick into a (fractional) matching $X^{(1)}$, without any knowledge of the remaining edges. Unmatched Stage 1 edges then disappear. Stage 2 reveals the remaining edges $E^{(2)} = E \setminus E^{(1)}$ of G as a subgraph $G^{(2)} = ((U_1, U_2), E^{(2)})$, and we (fractionally) pick a subset $X^{(2)}$ of them while ensuring that $X^{(1)} \cup X^{(2)}$ forms a (fractional) matching. This paper gives randomized algorithms that maximize the competitive ratio for this problem, i.e. ratio of the expected size of (fractional) matching obtained by the algorithm to the size of the maximum matching in G. In the integral version, the algorithm is allowed to use internal randomness, since otherwise the optimal competitive ratio is half. Note that the optimal competitive ratio for the fractional version is at least that of the integral version.

Our first main result is for the two-stage fractional bipartite matching problem. We say an algorithm is *instance optimal* if given the first stage graph $G^{(1)}$, it outputs a fractional matching $X^{(1)}$ that achieves the optimal competitive ratio over the adversarial choice of $G^{(2)}$.

Theorem 1. *There exists an instance optimal algorithm for the two-stage fractional bipartite matching problem.*

Although the above algorithm is instance optimal, it does not prove that for any $G^{(1)}$ the competitive ratio is more than half. Also, it does not work if one can only select an integral matching in $G^{(1)}$. We show that a $2/3$-competitive algorithm is always possible and that this ratio cannot be improved. Indeed, our result is on a generalization to multiple stages. In the *s-stage general matching problem*, edges of a graph G are revealed in s stages. At the end of each stage, we immediately and irrevocably decide which of that stage's edges to pick into a matching, while the other edges disappear. The greedy algorithm is still half competitive and a simple example shows that for $s \geq 3$ the optimal competitive ratio is strictly less than $2/3$ (see §A). We show that one can still beat half for a small number of stages.

Theorem 2. *There exists a $(\frac{1}{2} + \frac{1}{2^{s+1}-2})$-competitive algorithm for the s-stage integral bipartite matching problem. The competitive ratio $\frac{2}{3}$ for $s = 2$ is information-theoretically tight.*

We also prove similar results for general graphs.

Theorem 3. *There exists a $(\frac{1}{2} + \frac{1}{2^{O(s)}})$-competitive algorithm for the s-stage integral general matching problem. For the two-stage fractional general matching problem, there exists a 0.6-competitive algorithm.*

Our proofs for bipartite matching results extend to corresponding *multistage bipartite vertex cover* results[1]. We describe the problem and prove these corollaries in §B.

1.2 Our Techniques

Consider a simple example where Stage 1 reveals a single edge (u, v). Should our algorithm pick this edge irrevocably? Suppose it does not, then no edge might appear in

[1] For general graphs, approximating vertex cover more than half is UGC hard even when the entire graph is given [20].

Stage 2, which makes the competitive ratio 0. On the other hand, if it picks (u, v) then Stage 2 might reveal two edges (u', u) and (v, v'), where $u' \neq v'$. Now the maximum matching has size two, but the algorithm only picks a single edge, which is half-competitive. This example already shows that no deterministic algorithm can be more than half competitive and that no randomized algorithm can be more than $2/3$ competitive—picking (u, v) with probability $2/3$ is optimal.

A natural extension of the above algorithm is to pick a (carefully chosen) maximum matching with probability $2/3$. In §A, we give an illustrative example where picking a maximum matching with probability δ for any $\delta \in (0, 1)$ fails to achieve a $2/3$-competitive ratio, regardless of how the maximum matching is chosen. This establishes the fact that different parts of the graph must use "different local distributions" to sample a matching. On the other hand, another important consideration is to avoid matching one vertex too much, since otherwise the adversary can ensure that the optimal edge indeed appears in Stage 2. Intuitively, the vertices should somehow be "uniformly matched". Our algorithms balance the above two seemingly contradictory objects by exploiting graph decomposition techniques and carefully chosen probability distributions.

We construct an online primal-dual algorithm, which finds online both a random matching $X^{(1)}$ and a dual solution $Y^{(1)}$ (vertex-cover) such that for any $G^{(2)}$ we can pick $X^{(2)} \subseteq E^{(2)}$ and a dual solution $Y^{(2)}$ (where $X^{(1)} \cup X^{(2)}$ is a matching) that satisfy the following two properties:

(i) ℓ_1 norm of $Y := Y^{(1)} + Y^{(2)}$ is the same as the cardinality of set $X := X^{(1)} \cup X^{(2)}$.

(ii) In expectation the dual solution $Y := Y^{(1)} + Y^{(2)}$ approximately satisfies every dual constraint, i.e., covers every edge of G by at least $2/3$.

Relaxed complementary slackness conditions now imply that the algorithm is $2/3$ competitive [31].

We view our technical contribution in two categories. The instance optimal algorithm for two-stage fractional bipartite matching uses a novel $\mathbf{LP}$ that computes both a matching and a vertex-cover (dual) simultaneously in the primal linear program. The $\mathbf{LP}$ allows both the algorithm and the adversary to interpret their optimal strategies. On the other hand, to prove concrete competitive ratios for various models, our technical contribution lies in the design of algorithms themselves, which are based on graph decomposition techniques and carefully chosen probability distributions to ensure Properties (i) and (ii). We believe these ideas to be useful for future research on online matching. In the following, we give brief overview of our results in more details.

Fractional Bipartite Matching: Instance optimality using a new LP. We write a linear program on $G^{(1)}$ to solve two-stage fractional bipartite matching problem. Our contribution in proving Theorem 1 is a new technique that strengthens the linear program for an online problem by moving the dual constraints (approximate edge-coverage) to the primal linear program. We believe that this technique might be of independent interest and will have other applications. Since our solutions are fractional, we use lower case letters x and y instead of X and Y. We maximize the competitive ratio α such that there exists a fractional matching $x^{(1)}$ in $E^{(1)}$ and a fractional vertex cover dual $y^{(1)}$ that satisfies: (a) ℓ_1 norms of $x^{(1)}$ and $y^{(1)}$ are equal and (b) every edge in $E^{(1)}$ is α-approximately covered by $y^{(1)}$. It turns out that these constraints are necessary but not sufficient by themselves. This means that for the optimal ratio α^* of the linear program, one can provide a Stage 2 graph $G^{(2)}$ where the algorithm cannot be more than α^* competitive; however, there might be graphs where α^* is not achievable by the algorithm. To further strengthen this linear program and prove our theorem, we add new constraints that force the dual y to cover "highly-matched" vertices in Stage 1.

Integral Bipartite Matching: Using Bipartite Matching Skeleton of [12]. For two-stage integral bipartite matching problem, the natural approach of rounding the instance optimal fractional matching solution fails. This has been also observed in previous online matching results [32]. In §A we give an example where going from fractional to integral setting strictly decreases the competitive ratio. Our first observation is that in the special case where $E^{(1)}$ contains a perfect matching, the algorithm that selects a perfect matching in $E^{(1)}$ w.p. $2/3$, and no edge otherwise, is $2/3$ competitive. To prove this we construct $Y^{(1)}$ by giving every matched vertex a value of $1/2$ (in expectation $1/3$). With simple case analysis, we show that for any $G^{(2)}$ we can always find $X^{(2)}$ and $Y^{(2)}$ that satisfy Properties (i) and (ii).

To obtain a $2/3$-competitive algorithm for any bipartite graph, we use a decomposition into bipartite matching skeleton due to Goel et al. [12]. It partitions the vertices of $G^{(1)}$ into disjoint *expanding pairs* (S_j, T_j) that satisfy $\alpha_j \cdot |N(S) \cap T_j| \geq |S|$ for any $S \subseteq S_j$; here j is an integer, $0 < \alpha_j \leq 1$, and $N(S)$ denotes the set of neighbors of S in $G^{(1)}$ (see §2). For each expanding pair (S_j, T_j), our online primal-dual algorithm finds a probability δ_j with which it picks a random maximum matching in (S_j, T_j), with some correlation between different pairs, and no edge in (S_j, T_j) otherwise. Moreover, we find ϵ_j that tells us how to distribute the mass of any picked edge between its vertices in the dual solution $y^{(1)}$. Some careful case analysis allows us to show that for any $E^{(2)}$ one can obtain both $X^{(2)}$ and $y^{(2)}$ that satisfy Properties (i) and (ii).

Integral General Matching: Using a new General Matching Skeleton. To beat half for two-stage general graph matching, we rely on Edmonds-Gallai decomposition. It gives us a characterization of any maximum matching in $G^{(1)}$ by partitioning the vertices of $G^{(1)}$ into three sets C, A, and D (see §2.2), where vertices in A form a "bridge" between vertices in D and C (see Figure 1) and the subgraph $G^{(1)}(C)$ of $G^{(1)}$ induced on C contains a perfect matching. For $G^{(1)}(C)$, as above, our algorithm again picks a perfect matching in C w.p. $2/3$, and no edge otherwise, while distributing the duals equally to all the vertices in C. Most of our effort goes in designing an algorithm for the induced subgraph $G^{(1)}(A \cup D)$.

The crucial difference between bipartite and general matching is that D is no longer independent and any maximum matching contains edges inside D. We choose $D' \subseteq D$ and apply our bipartite matching algorithm to the bipartite graph induced by $A \cup D'$ (ignoring edges inside A). Finally, we match edges inside D and construct duals to satisfy Properties (i) and (ii). Special care is taken for vertices in D' since they may be matched by both procedures. Analysis involves more technical work since the number of dual variables for general graph matching is exponential.

1.3 Further Related Works

Online matching has been also studied by the streaming community [9,12,17,21]. Instead of irrevocably choosing an edge into the matching, the algorithm is given $O(n)$ space[2]. It can use this space to store a subset of edges (the graph might have $O(n^2)$ edges) and in the end outputs a matching from the set of stored edges. A major open question in both online and streaming community is whether we can beat half for the *adversarial edge arrival model* [7,15,17,21]. Here the edges of an adversarial graph are revealed in an adversarial order. Whenever an edge is revealed, the algorithm has to immediately and irrevocably decide whether to pick/ store the edge. Adversarial edge arrival easily captures adversarial vertex arrival if we show all edges incident to a vertex together.

For edge arrival, the best known lower bound in the online setting is 0.57 [7] and in the streaming setting is $1 - \frac{1}{e}$ [17]. On the positive side, for the *random edge arrival model*, where edges of an adversarial graph are revealed in a uniformly random order, algorithms that beat half are known both in the streaming [21] and online settings [15]. The only known positive result for adversarial edge arrival is of Goel et al. [12]. They study two-stage matching problem in the streaming model, i.e., where the algorithm can store any $O(n)$ edges after first stage, and show tight matching upper and lower bounds of $2/3$. Our Theorem 2 can be seen as extending their result from streaming to the online setting; thereby showing that streaming setting does not give any additional power for the worst case graphs.

We note that since the decisions are irrevocable in the online setting, algorithmic results often become much harder going from streaming to online setting, while proving hardness results becomes easier. The $2/3$ lower bound of [12] is based on Ruzsa-Szemerédi graph and involves nontrivial technical work, whereas in §1 we saw a simple example that gives $2/3$ hardness for the online setting. On the other hand, we need several new ideas to design a $2/3$ competitive algorithm for the online setting.

1.4 Organization

§2 contains preliminaries and describes the graph decomposition results used in this work. §3 gives our instance optimal algorithm for two-stage fractional bipartite matching, proving Theorem 1. §4 proves Theorem 2 for bipartite matching. §4.1 first presents an algorithm that achieves $2/3$-competitive ratio for two-stage integral matching, and §4.2 generalizes this algorithm to show that achieves $\frac{1}{2} + \frac{1}{2^{s+1}-2}$-competitive ratio for s-stage integral matching. §5 proves Theorem 3 for general matching. §5.1 constructs our matching skeleton for general graphs. §5.2 presents an algorithm that achieves 0.6-competitive ratio for two-stage fractional matching, and §5.3 gives an algorithm that achieves $\frac{1}{2} + 2^{O(-s)}$-competitive ratio for s-stage integral matching. §B proves Corollary 1 and 2 for vertex cover.

[2] This setting is often called "semi-streaming" in the literature, while "streaming" means $O(polylog\ n)$ space. Since this is not our focus, we do not distinguish between the two settings.

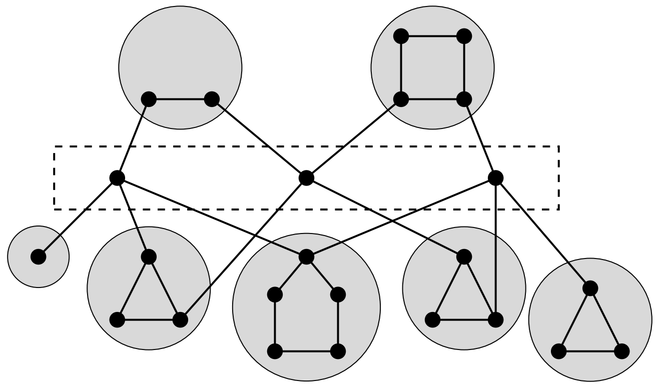


Fig. 1. Edmonds-Gallai decomposition

2 Preliminaries and Notation

In the s-stage matching problem, for each Stage i ($1 \leq i \leq s$), the graph $G^{(i)} = (V, E^{(i)})$ is given. Let $(G^{(1)} \cup \cdots \cup G^{(i)})$ denote the graph $(V, E^{(1)} \cup \cdots \cup E^{(i)})$.

For the integral matching problem, in stage i, the algorithm is supposed to return $X^{(i)} \subseteq E^{(i)}$ such that $X^{(1)} \cup \cdots \cup X^{(i)}$ is a matching in $G^{(1)} \cup \cdots \cup G^{(i)}$. The algorithm is allowed to use internal randomness. For the fractional matching problem, in stage i, the algorithm is supposed to return $x^{(i)} \in [0,1]^{E^{(i)}}$ such that $x^{(1)} + \cdots + x^{(i)}$ is in the matching polytope of $G^{(1)} \cup \cdots \cup G^{(i)}$. By definition, the competitive ratio for the integral matching problem is at most that of the fractional matching problem.

2.1 Notation

Let $G = (V, E)$ be an arbitrary graph. For $S \subseteq V$, let $G(S)$ be the subgraph induced by S and let $N(S) := \{v \in V \setminus S : (u, v) \in E \text{ for some } u \in S\}$. Let $G \setminus S := G(V \setminus S)$. Let $E(S)$ be the set of edges of $G(S)$. Let $o(G)$ be the number of odd components in G. Call an odd component $S \subseteq V$ *factor-critical* if for any $s \in S$, the induced subgraph $G(S \setminus \{s\})$ has a perfect matching. For $i \in \{1, 2\}$, we denote $3 - i$ by .

2.2 Graph Decompositions

We first state the classical Edmonds-Gallai decomposition of a general graph.

Lemma 1 (Theorem 3.2.1 **in [23]).** *Let $G = (V, E)$ be an undirected simple graph. Partition V into union of sets D, A, C, where $D = \{v \in V \mid \exists$ a max matching in G missing $v\}$, $A = N(D)$, $C = V \setminus (D \cup A)$. Then,*

1. *$C(G)$ consists of even components and has a perfect matching.*
2. *$D(G)$ consists of odd components and every component is factor-critical.*

3. *For every maximum matching of G, (1) every vertex in C is matched with another vertex in C, (2) every vertex in A is matched with another vertex in D, and (3) each odd component of $G(D)$ has at most one vertex matched to A.*

For bipartite graphs, one can obtain stronger properties from the decomposition due to Goel et al. [12].

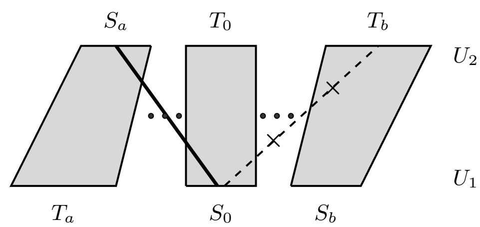


Fig. 2. Bipartite matching skeleton [12]. Solid edges can exist but not dashed edges.

Lemma 2 (Bipartite matching skeleton [12]). *Let $G = ((U_1, U_2), E)$ be a bipartite with no isolated vertex. There exists a partition of the vertices (see Figure 2) into pairs of subsets $\{(S_j, T_j)\}_j$, where j is an integer in the interval $[a, b]$ for integers $a \leq 0 \leq b$, such that*

1. *$S_j \subseteq U_1$ and $T_j \subseteq U_2$ for $j \geq 0$, and vice versa for $j < 0$.*
2. *$|T_j| = \frac{1}{\alpha_j}|S_j|$ and for any $P \subseteq S_j$, one has $|N(P) \cap T_j| \geq \frac{1}{\alpha_j}|P|$.*
3. *$\alpha_a < \alpha_{a+1} \cdots < \alpha_0 = 1 > \cdots > \alpha_{b-1} > \alpha_b$.*
4. *There exists a fractional matching between S_j and T_j such that vertices in S_j are perfectly matched and vertices in T_j are exactly α_j matched. Call (S_j, T_j) an α_j-*expanding pair.
5. *There is no edge of G between vertices in S_j and T_k for j, k where $\alpha_j > \alpha_k$.*
6. *There is no edge of G between vertices in T_j and T_k for any j, k.*

3 Instance Optimal Two-stage Fractional Bipartite Matching

In this section, we present a polynomial time instance optimal algorithm for the two-stage fractional bipartite matching, proving Theorem 1. Recall that an algorithm is *instance optimal* if for every $G^{(1)}$, it is guaranteed to achieve the optimal competitive ratio *given* $G^{(1)}$.

Theorem 4. *For any bipartite $G^{(1)}$, the following* **LP** *computes the optimal competitive ratio given $G^{(1)}$.*

$$\begin{array}{lll} \max & \alpha & \\ \text{s.t.} & f_u = \sum_{v \in N_1(u)} x^{(1)}_{u,v} & \forall u \in V \\ & f_u \leq 1, & \forall u \in V \end{array}$$

$$
\begin{aligned}
\textstyle\sum_{(u,v)\in E^{(1)}} x_{u,v}^{(1)} \geq \sum_u y_u^{(1)},& & \\
y_u^{(1)} + y_v^{(1)} \geq \alpha, & \qquad \forall (u,v) \in E^{(1)} & (1)\\
y_u^{(1)} \geq f_u - (1-\alpha), & \qquad \forall u \in V & (2)\\
x_{u,v}^{(1)}, y_u^{(1)} \geq 0, & \qquad \forall u, v \in V &
\end{aligned}
$$

We prove sufficiency and necessity directions in Lemma 3 and Lemma 4, respectively.

Lemma 3. $\mathbf{LP}$ *ensures that we can find a fractional matching x and fractional vertex cover certificate y, both of the same value, such that every edge appearing in first or second stage can be covered by at least α.*

Proof. Consider any 2nd stage extension LP for a given first stage solution $x^{(1)}$. Also, consider its dual.

$$
\begin{array}{ll|ll}
\max & \sum_{(u,v)\in E^{(2)}} x_{u,v}^{(2)} & \min & \sum_u y'_u(1-f_u) \\
\text{s.t.} & & \text{s.t.} & y'_u + y'_v \geq 1, \quad \forall (u,v) \in E^{(2)} \\
\sum_{v\in N_2(u)} x_{u,v}^{(2)} \leq 1 - f_u, & \forall u \in V & & \\
x_{u,v}^{(2)} \geq 0, & \forall u, v \in V & & y'_u \geq 0, \quad \forall u \in V
\end{array}
\qquad (3)
$$

For any vertex u, define second stage vertex cover $y_u^{(2)}$ to be $y'_u(1-f_u)$. Hence the fractional vertex cover y is defined as $y_u := y_u^{(1)} + y_u^{(2)} = y_u^{(1)} + y'_u(1-f_u)$. It can be easily verified that $\|y\|_1$ is the same as the obtained fraction matching. Eq. (1) tells that any first stage edge is α covered by y. We next show that any second stage edge (u, v) is also α covered by y. This is because $y_u + y_v$

$$
\begin{aligned}
&= y_u^{(1)} + y_v^{(1)} + y'_u(1-f_u) + y'_v(1-f_v) \\
&\geq y_u^{(1)} + y_v^{(1)} + (y'_u + y'_v)(1 - \max\{f_u, f_v\}) \\
&\geq -(1-\alpha) + 1 = \alpha \qquad \text{(using Eq. (3) and Eq. (2))}.
\end{aligned}
$$

Lemma 4. $\mathbf{LP}$ *is tight, i.e. we can produce a Stage 2 graph s.t. no algorithm can be better than α competitive.*

Proof. We prove by contradiction and consider any decision x^* at the end of the first stage (this also fixes f_u^*) by an optimal algorithm with competitive ratio $\beta > \alpha$. We note that the optimal value of the following LP is greater than $\sum x_{u,v}^*$ as otherwise we get a feasible solution to $\mathbf{LP}$ with value β, and this is a contradiction that α is the optimal value of $\mathbf{LP}$.

$$
\begin{aligned}
\min \quad & \sum_u y'_u \\
\text{s.t.} \quad & y'_u + y'_v \geq \alpha, & \forall (u,v) \in E^{(1)} \\
& y'_u \geq f_u^* - (1-\alpha), & \forall u \in V_1
\end{aligned}
$$

$$y'_u \geq 0, \qquad \forall u \in V$$

Also, consider its dual linear program.

$$\begin{aligned} \max \quad & \sum_{(u,v)\in E^{(1)}} \alpha\, Z_{u,v} + \sum_u (f_u^* - (1-\alpha))\, Y_u \\ \text{s.t.} \quad & \textstyle\sum_{v\in N_1(u)} Z_{u,v} + Y_u \leq 1, && \forall u \in V_1 \\ & Z_{u,v}, Y_u \geq 0, && \forall u, v \in V_1 \end{aligned} \tag{4}$$

Proposition 1. *The above dual linear program has an optimal integral solution.*

Proof. Given an instance of the dual LP, consider the instance of the maximum weight matching where each edge $(u, v) \in E^{(1)}$ has weight α and each for each vertex $u \in V$, we create a new vertex u' and an edge (u, u') with weight $f_u^* - (1-\alpha)$. There is one-to-one correspondence between feasible solutions of the two problems. Since maximum weight matching always has an optimal integral solution, so does the above dual LP.

Given the solution to the above dual LP, the second stage graph consists $\sum Y_u$ disjoint edges, each adjacent to exactly one vertex with $Y_u = 1$. Note that due to Eq. (4), edges with $Z_{u,v} = 1$ can never be adjacent to a vertex with $Y_u = 1$. Hence the optimum matching for this two stage graph is at least $\sum_u Y_u + \sum_{(u,v)\in E^{(1)}} Z_{u,v}$. On the other hand, the two stage algorithm's value is $\sum_u x_{u,v}^* + \sum_u Y_u\,(1 - f_u^*)$. Combining,

$$\begin{aligned} \alpha\,\mathsf{OPT} - \mathsf{ALG} &\geq \alpha \left(\textstyle\sum_u Y_u + \sum_{(u,v)\in E^{(1)}} Z_{u,v}\right) - \left(\textstyle\sum_{u,v} x_{u,v}^* + \sum_u Y_u\,(1 - f_u^*)\right) \\ &= \left(\textstyle\sum_{(u,v)\in E^{(1)}} \alpha\, Z_{u,v} + \sum_u (f_u^* - (1-\alpha))\, Y_u\right) - \textstyle\sum_{u,v} x_{u,v}^* \quad > 0. \end{aligned}$$

4 Bipartite Matching

In this section, we prove Theorem 2 for bipartite matching. §4.1 first presents an algorithm that achieves $2/3$-competitive ratio for two-stage integral matching, and §4.2 generalizes this algorithm to show that achieves $\frac{1}{2} + \frac{1}{2^{s+1}-2}$-competitive ratio for s-stage integral matching.

4.1 Two-stage Integral Bipartite Matching

We show that the optimal competitive ratio for two-stage integral bipartite matching is exactly $2/3$, proving Theorem 2 for $s = 2$. We already know from §1 that no algorithm can be better than $2/3$-competitive for two-stage integral bipartite matching problem. To prove the other direction, the idea is to find matching $X^{(1)}$ in a way that we have a corresponding fractional dual solution $Y^{(1)}$ such that for any Stage 2 graph $G^{(2)}$, we can find a matching $X^{(2)}$ and dual $Y^{(2)}$ where in expectation $Y^{(1)} + Y^{(2)}$ covers every edge in G by $2/3$.

Warmup: $G^{(1)}$ Contains a Perfect Matching Consider a simple algorithm that picks the perfect matching into $X^{(1)}$ w.p. $2/3$, and no edge otherwise. In Stage 2, the algorithm picks the maximum possible matching $X^{(2)} \in E^{(2)}$ such that $X^{(1)} \cup X^{(2)}$ is a matching. This is equivalent to finding a maximum matching in $G^{(2)}$ after ignoring the vertices matched in $X^{(1)}$.

To prove that this algorithm is $2/3$ competitive, for any vertex u we set $Y_u^{(1)} = 1/2$ whenever it's matched. For Stage 2, since bipartite maximum matching is equivalent to bipartite vertex cover, it also gives us integral vertex dual $Y^{(2)}$ such that $|Y^{(2)}| = |X^{(2)}|$ and every edge in $G^{(2)}$, with none of its vertices matched in $X^{(1)}$, is covered by $Y^{(2)}$. To prove that the algorithm is $2/3$ competitive, we show that for any edge $(u, v) \in E^{(1)} \cup E^{(2)}$,

$$\mathsf{E}[Y_u] + \mathsf{E}[Y_v] = \mathsf{E}[Y_u^{(1)}] + \mathsf{E}[Y_v^{(1)}] + \mathsf{E}[Y_u^{(2)}] + \mathsf{E}[Y_v^{(2)}] \geq 2/3.$$

For $(u, v) \in E^{(1)}$, the above equation is simply true because $\mathsf{E}[Y_u^{(1)}] + \mathsf{E}[Y_v^{(1)}] = \Pr[\text{Perfect matching picked}] \cdot (\frac{1}{2} + \frac{1}{2}) = \frac{2}{3}(\frac{1}{2} + \frac{1}{2}) = \frac{2}{3}$.

Now, consider an edge $(u, v) \in E^{(2)}$. Consider first the case where both u and v have an edge incident to them in Stage 1. Then, similar to above, we have $\mathsf{E}[Y_u^{(1)}] + \mathsf{E}[Y_v^{(1)}] = \frac{2}{3}(\frac{1}{2} + \frac{1}{2}) = \frac{2}{3}$. So WLOG assume v has no edge incident to it in $E^{(1)}$. Now,

$$\begin{aligned}
\mathsf{E}[Y_u] + \mathsf{E}[Y_v] = & \Pr[u \text{ is matched in } X^{(1)}] \cdot \mathsf{E}[Y_u^{(1)} + Y_v^{(1)} \mid u \text{ is matched in } X^{(1)}\ ] + \\
& \Pr[u \text{ is not matched in } X^{(1)}] \cdot \mathsf{E}[Y_u^{(2)} + Y_v^{(2)} \mid u \text{ is not matched in } X^{(1)}\ ] \\
= &\ \Pr[u \text{ is matched in } X^{(1)}] \cdot \frac{1}{2} + \Pr[u \text{ is not matched in } X^{(1)}] \cdot 1 \\
= &\ \frac{2}{3} \qquad (\text{because } \Pr[u \text{ is matched in } X^{(1)}] = 2/3).
\end{aligned}$$

Any Bipartite Graph $G^{(1)}$

Algorithm and construction of duals. The algorithm starts by constructing a matching skeleton for $G^{(1)}$ as described in §2.2. For $j \in \{a, \ldots, -1, 0, 1, \ldots, b\}$, let (S_j, T_j) denote the obtained expanding pair with expansion α_j between 0 and 1. We define

$$\delta_j := \frac{3 - \alpha_j}{3} \quad \text{and} \quad \epsilon_j := \frac{2 - \alpha_j}{3 - \alpha_j}.$$

The algorithm chooses a uniformly random r between $[0, 1]$ and picks a random maximum matching between all (S_j, T_j) with $\alpha_j < r$. Note that picking a matching in (S_j, T_j) implies every vertex in S_j is matched w.p. 1 and each vertex in T_j is matched w.p. probability α_j (not independently). The dual variables $Y^{(1)}$ are given values in a natural way: for any edge (u, v) in (S_j, T_j) picked into matching $X^{(1)}$, we assign $Y_u^{(1)} = \epsilon_j$ and $Y_v^{(1)} = 1 - \epsilon_j$. This clearly satisfies $\|Y^{(1)}\|_1 = |X^{(1)}|$.

In Stage 2, the algorithm picks the maximum possible matching $X^{(2)} \in E^{(2)}$ such that $X^{(1)} \cup X^{(2)}$ is a matching. This is equivalent to finding a maximum matching in $G^{(2)}$ after ignoring the vertices matched in $X^{(1)}$. Since bipartite maximum matching is equivalent to bipartite vertex cover, this step also gives us integral vertex dual $Y^{(2)}$ with

Algorithm 1 Two-stage Integeral Bipartite Matching Algorithm

Stage 1
1: Construct a matching skeleton with expanding pairs (S_j, T_j) and expansion α_j.
2: Define $\delta_j := \frac{3-\alpha_j}{3}$ and generate a uniformly random real $r \in [0, 1]$.
3: **for** each j **do**
4: **if** $\delta_j < r$ **then**
5: Pick a random maximum matching between (S_j, T_j) that matches all vertices of S_j w.p. 1 and each vertex of T_j w.p. exactly α_j.
6: **end if**
7: **end for**

8: **Stage** 2 Pick the optimal matching extension in $G^{(2)}$.

$|Y^{(2)}| = |X^{(2)}|$ and every edge in $G^{(2)}$, with none of its vertices matched in $X^{(1)}$, is covered by $Y^{(2)}$. In fact, the next section shows that for any edge $(u, v) \in E^{(1)} \cup E^{(2)}$, we have

$$\mathsf{E}[Y_u] + \mathsf{E}[Y_v] = \mathsf{E}[Y_u^{(1)}] + \mathsf{E}[Y_v^{(1)}] + \mathsf{E}[Y_u^{(2)}] + \mathsf{E}[Y_v^{(2)}] \geq 2/3.$$

Analysis of the online primal-dual algorithm. We show $\mathsf{E}[Y]$ covers every edge in $E^{(1)} \cup E^{(2)}$ by $2/3$. First, consider any Stage 1 edge (u, v). The only possible cases due to Lemma 2 are: (a) $u \in S_j$ and $v \in T_k$ for $\alpha_j \leq \alpha_k$, and (b) $u \in S_j$ and $v \in S_k$.

(a) $u \in S_j$ and $v \in T_k$ for $\alpha_j \leq \alpha_k$: Using linearity of expectation and noting that v is matched w.p. $\delta_k \alpha_k$,
$$\mathsf{E}[Y_u^{(1)}] + \mathsf{E}[Y_v^{(1)}] = \delta_j \epsilon_j + \delta_k \alpha_k (1 - \epsilon_k) \geq \delta_j \epsilon_j + \delta_j \alpha_j (1 - \epsilon_j) = 2/3,$$
where the inequality is because $\delta\alpha(1 - \epsilon) = \frac{\alpha}{3}$ decreases with decrease in α.

(b) $u \in S_j$ and $v \in S_k$: Using linearity of expectation,
$$\mathsf{E}[Y_u^{(1)}] + \mathsf{E}[Y_v^{(1)}] = \delta_j \epsilon_j + \delta_k \epsilon_k \geq 2\,\delta_0 \epsilon_0 = 2/3, \text{ because } \delta\epsilon \text{ is minimum for } \alpha = 1.$$

Next, consider any Stage 2 edge (u, v). Since Lemma 2 does not apply to Stage 2 edges, we need to consider all the following cases: (a) $u \in S_j$ and $v \in T_k$, (b) $u \in S_j$ and $v \in S_k$, (c) $u \in T_j$ and $v \in T_k$, and (d) $u \in S_j$ and v new, (e) u new and $v \in T_k$. We only discuss Case (c) here, and refer to §4.2 for others.

Case (c) *($u \in T_j$ and $v \in T_k$)*: WLOG assume $\alpha_j \geq \alpha_k$. We note that in Stage 1 vertex u is matched w.p. $\delta_j \alpha_j$ and vertex v w.p. $\delta_k \alpha_k$. Using linearity of expectation,
$$\mathsf{E}[Y_u^{(1)}] + \mathsf{E}[Y_v^{(1)}] = \delta_j \alpha_j (1 - \epsilon_j) + \delta_k \alpha_k (1 - \epsilon_k) = \alpha_j/3 + \alpha_k/3. \tag{5}$$
Also, Stage 2 dual gets value 1 if $r > \delta_k$, or v is not matched for $r \in [\delta_j, \delta_k]$, or both u, v not matched for $r < \delta_j$.
$$\mathsf{E}[Y_u^{(2)}] + \mathsf{E}[Y_v^{(2)}] = (1 - \delta_k) + (\delta_k - \delta_j)(1 - \alpha_k) + \delta_j (1 - \alpha_j)(1 - \alpha_k). \tag{6}$$
Summing (5) and (6), and substituting for δ_j and δ_k gives,
$$\begin{aligned} \mathsf{E}[Y_u] + \mathsf{E}[Y_v] &= \mathsf{E}[Y_u^{(1)} + Y_u^{(2)}] + \mathsf{E}[Y_v^{(1)} + Y_v^{(2)}] \\ &= \frac{1}{3}\left(2 + (1 - \alpha_j - \alpha_k)^2 + \alpha_j \alpha_k (1 - \alpha_j)\right) \quad \geq \quad 2/3. \end{aligned}$$

Algorithm 2 s-stage Integeral Bipartite Matching Algorithm

for each stage s **do**
    Construct a matching skeleton with expanding pairs (S_j, T_j) and expansion α_j.
    Define $\delta_j := \frac{p-\alpha_j(p-q)}{p+\alpha_j(1-p)}$ and generate a uniformly random real $r \in [0,1]$.
    for each j **do**
        if $\delta_j < r$ **then**
            Pick a random maximum matching between (S_j, T_j) that matches all vertices of S_j w.p. 1 and each vertex of T_j w.p. exactly α_j.
        end if
    end for
end for

4.2 s-stage Fractional and Integral Bipartite Matching

We generalize the algorithm from the previous section to prove Theorem 2 by induction on the number of stages, which implies the theorem is true for fractional bipartite matching. The idea is to match vertices in a way such that we have a corresponding fractional dual solution that for any next stage graph can be extended to violate each dual constraint by a small amount. The base case is trivially true for $s = 1$ since we can just pick the maximum matching. Suppose our algorithm can guarantee in expectation p competitive ratio for $s-1$ stages. We show that for s stages the algorithm can guarantee $q := \frac{2p}{2p+1}$ competitive ratio, which proves the induction step. Note that both p and q are functions of s.

Algorithm The algorithm starts by constructing a matching skeleton for Stage 1 bipartite graph $G^{(1)}$ as described in §2.2. For $j \in \{\ldots, -2, -1, 0, 1, 2, \ldots\}$, let (S_j, T_j) denote the obtained expanding pair with expansion $\alpha_j \leq 1$. We define

$$\delta_j := \frac{p - \alpha_j(p-q)}{p + \alpha_j(1-p)}.$$

We choose a uniformly random r between $[0,1]$ and pick a random maximum matching between all (S_j, T_j) with $\alpha_j < r$. Note that picking a matching in (S_j, T_j) implies every vertex in S_j is matched w.p. 1 and each vertex in T_j is matched w.p. probability α_j (not independently). The dual variables $Y^{(1)}$ are given values in a natural way: for any edge (u, v) in (S_j, T_j) picked into matching $X^{(1)}$, we assign $Y_u^{(1)} = \epsilon_j$ and $Y_v^{(1)} = 1 - \epsilon_j$. This clearly satisfies $\|Y^{(1)}\|_1 = |X^{(1)}|$.

For the remaining $s - 1$ stages, the algorithm operates recursively on the obtained graph.

Analysis of the Induction Step Let x denote the indicator variable for the matching obtained by the above algorithm. We construct a dual solution y such that $\sum_u y_u = \sum_e x_e$. Moreover, for every edge (u, v) in the graph,

$$\mathsf{E}[y_u] + \mathsf{E}[y_v] \geq q, \tag{7}$$

where the expectation is over random choices of the algorithm. To define how the dual variables behave, for every expanding pair (S_j, T_j) we define

$$\epsilon_j := \frac{pq - \alpha_j(p-q)}{p - \alpha_j(p-q)}$$

The dual variables are given values in a natural way: for any edge (u, v) picked in (S_j, T_j), we assign $y_u = \epsilon_j$ and $y_v = 1 - \epsilon_j$. We show Eq. (7) holds for any stage edge $(u, v) \in E$ using the following simple fact.

Fact 5 *For any j, if α_j decreases then the following expressions increase:*

$$-\alpha_j, \quad \delta_j, \quad \epsilon_j, \quad \epsilon_j\delta_j, \quad \delta_j(1-\epsilon_j), \quad -\alpha_j(1-\epsilon_j), \quad -\alpha_j\delta_j(1-\epsilon_j).$$

Fact 6 *For all j*

$$\begin{aligned}\delta_j\epsilon_j + \delta_j\alpha_j(1-\epsilon_j) &= q\\ \delta_j\alpha_j(1-\epsilon_j) &= (1-\delta_j)p\\ \delta_j\epsilon_j &= q - (1-\delta_j)p\end{aligned}$$

Proof. Simple algebra using the definitions.

Fact 7 *For all j*

$$1 - \delta_k - \alpha_k(\delta_k - q) \geq 0$$

Proof. To simplify notation we remove subscripts.

$$\begin{aligned}1 - \delta - \alpha(\delta - q) &= \alpha\left(\frac{(1-q) - (p - \alpha(p-q) - q(p + \alpha(1-p)))}{p + \alpha(1-p)}\right)\\ &= \alpha\left(\frac{(1-q)(1-p) + \alpha p(1-q)}{p + \alpha(1-p)}\right) \geq 0.\end{aligned}$$

Fact 8 *For all j, k*

$$\begin{aligned}&3 - \delta_j(1+\alpha_j) - \delta_k(1+\alpha_k) + q\alpha_j\alpha_k \geq 2(1-q)\\ \iff &1 + 2q - \delta_j(1+\alpha_j) - \delta_k(1+\alpha_k) + q\alpha_j\alpha_k \geq 0\end{aligned}$$

Proof. On simplifying, it gives denominator of $(1+2p)(-\alpha_j - p + \alpha_j p)(-\alpha_k - p + \alpha_k p) \geq 0$ and numerator

$$(-1+\alpha_k)^2p^2(-1+2p) + \alpha_j^2 p(2\alpha_k^2(-1+p)^2 + p(-1+2p) + \alpha_k(-1+5p-4p^2)) + \\ \alpha_j(2(1-2p)p^2 - \alpha_k^2 p(1-5p+4p^2) + \alpha_k(1-7p^2+8p^3))$$

This is verified as Case 0 of §C.

Stage 1 edges We first argue for any Stage 1 edge (u, v). The only possible cases due to Lemma 2 are the following.

(a) $u \in S_j$ and $v \in T_k$ for $\alpha_j \leq \alpha_k$: The expected dual is

$$\begin{aligned}&\delta_j\epsilon_j + \delta_k\alpha_k(1-\epsilon_k)\\ \geq\ &\delta_j\epsilon_j + \delta_j\alpha_j(1-\epsilon_j) = q. \qquad \text{(using Fact 5 and Fact 6)}\end{aligned}$$

(b) $u \in S_j$ and $v \in S_k$: The expected dual is

$$
\begin{aligned}
& \delta_j \epsilon_j + \delta_k \epsilon_k \\
& \geq 2\,\delta_0 \epsilon_0 && (\text{Fact 5 gives } \delta\epsilon \text{ is min for } \alpha = 1) \\
& = 2\,(pq - (p-q)) \geq q. && (\text{by definition})
\end{aligned}
$$

Stage 2 edges Next, consider any edge (u, v) that appears in Stage 2. Since Lemma 2 does not apply to Stage 2 edges, we consider all the following cases.

(a) $u \in S_j$ and $v \in T_k$: The expected dual for the case $\alpha_j \leq \alpha_k$ is

$$
\begin{aligned}
& \delta_k\,(\epsilon_j + (1-\epsilon_k)\alpha_k) + (1-\delta_j) \cdot p + (\delta_j - \delta_k)\epsilon_j \\
& \geq \delta_k\,(\epsilon_k + (1-\epsilon_k)\alpha_k) && (\epsilon_j \geq \epsilon_k \text{ using Fact 5}) \\
& = q && (\text{using Fact 6})
\end{aligned}
$$

The expected dual for the other case $\alpha_j > \alpha_k$ is

$$
\begin{aligned}
& \delta_j\,(\epsilon_j + (1-\epsilon_k)\alpha_k) + (1-\delta_k) \cdot p + (\delta_k - \delta_j)\,((1-\alpha_k) \cdot p + \alpha_k(1-\epsilon_k)) \\
& = \ (\delta_k\epsilon_k + \delta_k(1-\epsilon_k)\alpha_k) + (1-\delta_k) \cdot p + (\delta_k - \delta_j)(1-\alpha_k) \cdot p - (\delta_k\epsilon_k - \delta_j\epsilon_j) \\
& = q + p(1 - \delta_k\alpha_k - \delta_j + \delta_j\alpha_k) - (\delta_k\epsilon_k - \delta_j\epsilon_j) && (\text{using Fact 6}) \\
& = q + p(1 - \delta_k - \alpha_k(\delta_k - \delta_j)) && (\text{using Fact 6}) \\
& \geq q + p(1 - \delta_k - \alpha_k(\delta_k - q)) && (\delta_j \geq q) \\
& \geq q && (\text{using Fact 7})
\end{aligned}
$$

(b) $u \in S_j$ and $v \in S_k$: WLOG assume $\alpha_j \geq \alpha_k$. The expected dual is

$$
\begin{aligned}
& \delta_j\,(\epsilon_j + \epsilon_k) + (1-\delta_k) \cdot p + (\delta_k - \delta_j)\epsilon_k \\
& \geq 2\,\delta_0\epsilon_0 && (\text{Fact 5 gives } \delta\epsilon \text{ is min for } \alpha = 1) \\
& = 2\,(pq - (p-q)) \geq q. && (\text{by definition})
\end{aligned}
$$

(c) $u \in T_j$ and $v \in T_k$: WLOG assume $\alpha_j \geq \alpha_k$. The expected dual is

$$
\begin{aligned}
& \delta_j\,(\alpha_j(1-\epsilon_j) + \alpha_k(1-\epsilon_k) + (1-\alpha_j)(1-\alpha_k) \cdot p) + \\
& \qquad (1-\delta_k) \cdot p + (\delta_k - \delta_j)\,(\alpha_k(1-\epsilon_k) + (1-\alpha_k) \cdot p) \\
& = q + p(1 - \delta_k - \alpha_k(\delta_k - \delta_j)) + \delta_j(\alpha_j(1-\epsilon_j) + (1-\alpha_j)(1-\alpha_k) \cdot p - \epsilon_j) \\
& && (\text{using Case (a)}) \\
& = p\,(3 - \delta_j(1+\alpha_j) - \delta_k(1+\alpha_k) + \delta_j\alpha_j\alpha_k) \\
& \geq p\,(3 - \delta_j(1+\alpha_j) - \delta_k(1+\alpha_k) + q\alpha_j\alpha_k) && (\text{using } \delta_j \geq q) \\
& \geq p(2(1-q)) = q. && (\text{using Fact 8})
\end{aligned}
$$

(d) $u \in S_j$ and v new, or u new and $v \in T_k$: The expected dual values are at least as much as previous cases.

Stage $\geq$ 3 edges Here we are already done because induction hypothesis gives that these edges are covered by $p \geq q$.

5 General Matching

In this section, we prove Theorem 3 for general matching. §5.1 constructs our matching skeleton for general graphs. §5.2 presents an algorithm that achieves 0.6-competitive ratio for two-stage fractional matching, and §5.3 gives an algorithm that achieves $\frac{1}{2} + 2^{O(-s)}$-competitive ratio for s-stage integral matching.

5.1 General Matching Skeleton

We introduce our general matching skeleton for general graphs. It is based on the Edmonds-Gallai decomposition (Lemma 1) and the bipartite matching skeleton due to Goel et al. (Lemma 2).

Given $G^{(1)} = (V, E^{(1)})$, let $V^{(1)} \subseteq V$ be the vertices incident to at least one edge in $E^{(1)}$. First apply the Edmonds-Gallai decomposition (Lemma 1) to partition $V^{(1)}$ into three subsets D, A, C. Further partition D to D_s and D_l such that $G^{(1)}(D_s)$ is the set of isolated vertices, and $G^{(1)}(D_l)$ is the union of odd components whose size is at least three. Let $\hat{D}_l$ be the set of such odd components of size at least three (so that $\bigcup_{U \in \hat{D}_l} U = D_l$).

Fix a maximum matching $M \subseteq E^{(1)}$. For each odd component $U \in \hat{D}_l$, at most one vertex is matched to A. Let $r(U)$ be the vertex in U matched to A in M. If no vertex is matched to A, let $r(U) := *$ (assume $* \notin V$). Let $D' = D_s \cup \{r(U) : r(U) \neq *\}_{U \in \hat{D}_l}$.

Consider the bipartite graph $G^{(1)}(A, D')$, where the set of vertices is $A \cup D'$ and the set of edges is $\{(u, v) \in E : u \in A, v \in D'\}$. By definition of D', M induces a matching in $G^{(1)}(A, D')$ that matches every vertex in A. (See Figure 3 for an example.) We use the bipartite matching skeleton of $G^{(1)}(A, D')$ to have $S_0, \ldots, S_a \subseteq A$ and $T_0, \ldots, T_a \subseteq D'$ with $1 \geq \alpha_0 > \cdots > \alpha_b$. Since there exists a matching in G' that matches every vertex in A, all S_i's are contained in A and all T_i's are contained in D'. See [12, Lemma 3.2] for details.

5.2 Two-stage Fractional General Matching

We present a two-stage fractional bipartite matching algorithm that achieves a 0.6-competitive ratio. Given the first graph $G^{(1)} = (V, E^{(1)})$, Algorithm 3 describes how to construct a fractional matching $\{x^{(1)}_{u,v}\}_{u,v}$. Given Edmonds-Gallai decomposition (C, A, D) for $G^{(1)}$, the algorithm selects 0.6 times a perfect matching within C. Our previous algorithm for bipartite graphs is run on the bipartite graph between A and D' (ignoring edges inside A), with slightly different parameters. Finally, for large odd components $U \subseteq D_l$, we choose a fractional matching within U. Some additional care must be taken when $r(U) \in T_j$ for some T_j, since it may be already fractionally matched.

An important property of Algorithm 3 is that every vertex is fractionally matched with at least (and close to) 0.6 unless it forms a singleton component in $G^{(1)}(D)$. Dual variables $\{y^{(1)}_u\}_u$ are constructed in the following way: $y^{(1)}_u \leftarrow \alpha_j - 0.4\alpha_j^2$ if u forms a singleton component in $G(D)$ and $u \in T_j$. Let $y^{(1)}_v \leftarrow 0.3$ for all other $v \in V^{(1)}$. The following proposition shows that $\|x^{(1)}\|_1 = \|y^{(1)}\|_1$.

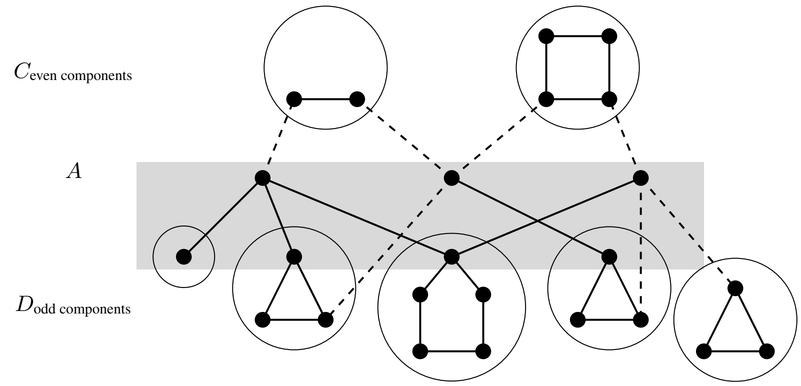


Fig. 3. Edmonds-Gallai decomposition for general matching. The shaded region represents $G^{(1)}(A, D')$ where we apply our bipartite matching algorithm. The dashed edges are never picked by our algorithm.

Proposition 2. *After Algorithm 3 terminates,* $\|x^{(1)}\|_1 := \sum_{(u,v)\in E^{(1)}} x^{(1)}_{u,v} = \|y^{(1)}\|_1 := \sum_{v\in V} y^{(1)}_v$.

Proof. We check each of 5 locations that we change $x^{(1)}$ and $y^{(1)}$.

(a) Line 7: $\|x^{(1)}\|_1$ is increased by $\frac{0.3(\alpha_j+1)}{\alpha_j}|S_j|$. $\|y^{(1)}\|_1$ is increased by $0.3|S_j|$ from S_j, and $0.3|T_j| = 0.3\frac{|S_j|}{\alpha_j}$ from T_j. Therefore, $\|y^{(1)}\|_1$ is also increased by $\frac{0.3(\alpha_j+1)}{\alpha_j}|S_j|$.
(b) Line 9: $\|x^{(1)}\|_1$ is increased by $(1-0.4\alpha)|S_j|$. $\|y^{(1)}\|_1$ is increased by $(0.6-0.4\alpha)|S_j|$ from S_j, and $0.4\alpha|T_j| = 0.4|S_j|$ from T_j. Therefore, $\|y^{(1)}\|_1$ is also increased by $(1-0.4\alpha)|S_j|$.
(c) Line 15: $\|x^{(1)}\|_1$ is increased by $\frac{0.6}{|U|-1}\cdot\frac{|U|(|U|-1)}{2} = 0.3|U|$. $\|y^{(1)}\|_1$ is also increased by $0.3|U|$.
(d) Line 19: $\|x^{(1)}\|_1$ is increased by $0.8\alpha\frac{|U|-1}{2} + \frac{0.6-0.8\alpha}{|U|-1}\cdot\frac{|U|(|U|-1)}{2} = 0.3|U| - 0.4\alpha$. $y^{(1)}_{r(U)}$ is increased by $0.3-0.4\alpha$, and $y^{(1)}_u$ for $u \in U \setminus \{r(U)\}$ is increased by 0.3. Therefore, $\|y^{(1)}\|_1$ is also increased by $0.3|U| - 0.4\alpha$.
(e) Line 21: $\|x\|_1$ is increased by $0.6\cdot\frac{(|U|-1)}{2} = 0.3(|U|-1)$. $\|y\|_1$ is also increased by $0.3(|U|-1)$.

It remains to verify whether $x^{(1)}$ is in the matching polytope. It will be proved later when we discuss Stage 2 edges.

For $u \in V$, let $f_u = \sum_{(u,v)\in E^{(1)}} x^{(1)}_{u,v}$ and $f_U = \sum_{(u,v)\in E^{(1)}(U)} x^{(1)}_{u,v}$. Given the second graph $G^{(2)} = (V, E^{(2)})$, the maximum fractional matching is computed by the following LP.

$$\max \qquad \sum_{(u,v)\in E^{(2)}} x^{(2)}_{u,v}$$

Algorithm 3 Two-stage Fractional Matching Algorithm

Stage 1

Compute the Edmonds-Gallai decomposition (C, A, D) of $G^{(1)}$.

Let M_C be a perfect matching for $G^{(1)}(C)$. Let $x_e^{(1)} \leftarrow 0.6$ for all $e \in M_C$, and let $y_u^{(1)} \leftarrow 0.3$ for all $u \in C$.

Construct the matching skeleton with expanding pairs $\{(S_j, T_j)\}$ and expansion α_j for G' defined above.

for each j **do**

  Pick a fractional matching z between (S_j, T_j) that matches all vertices of S_j with value 1 and each vertex of T_j with value exactly α_j.

  if $\alpha_j \geq \frac{3}{4}$ **then**

    $x^{(1)} \leftarrow x^{(1)} + \frac{0.3(\alpha_j+1)}{\alpha_j} z$ for $(u, v) \in E^{(1)}(S_j, T_j)$. $y_u^{(1)} \leftarrow 0.3$ for all $u \in S_j \cup T_j$.

  else

    $x^{(1)} \leftarrow x^{(1)} + (1 - 0.4\alpha_j) z_{u,v}$. $y_u^{(1)} \leftarrow 0.6 - 0.4\alpha$ for $u \in S_j$. $y_u^{(1)} \leftarrow 0.4\alpha$ for $u \in T_j$.

  end if

end for

for each odd component $U \in \hat{D}_l$ **do**

  For $v \in U$, let z^v be a perfect matching in $G(U \setminus v)$.

  if $r(U) = *$ **then**

    $x^{(1)} \leftarrow x^{(1)} + \frac{0.6}{|U|-1} \sum_{v \in U} z^v$. $y_u^{(1)} \leftarrow 0.3$ for all $u \in U \setminus \{r(U)\}$.

  else

    Let j be such that $r(U) \in T_j$ for some j.

    if $\alpha_j < \frac{3}{4}$ **then**

      $x^{(1)} \leftarrow x^{(1)} + 0.8\alpha z^{r(U)} + \frac{0.6-0.8\alpha}{|U|-1} \sum_v z^v$. $y_u^{(1)} \leftarrow 0.3$ for all $u \in U$.

    else

      $x^{(1)} \leftarrow x^{(1)} + 0.6 z^{r(U)}$. $y_u^{(1)} \leftarrow 0.3$ for all $u \in U \setminus \{r(U)\}$.

    end if

  end if

end for

Stage 2 Pick the optimal fractional matching extension in $G^{(2)}$.

$$
\begin{array}{lll}
\text{s.t.} & \sum_{v \in N(u)} x_{u,v}^{(2)} \leq 1 - f_u & \forall u \in V \\
& \sum_{(u,v) \in E^{(2)}(U)} x_{u,v}^{(2)} \leq \frac{|U|-1}{2} - f_U & \forall |U| \text{ odd} \\
& x_{u,v}^{(1)} \geq 0 & \forall u, v \in V
\end{array}
$$

$$
\begin{array}{lll}
\min & \sum_u y'_u (1 - f_u) + \sum_{|U| \text{ odd}} y'_U \left(\frac{|U|-1}{2} - f_U \right) & \\
\text{s.t.} & y'_u + y'_v \geq 1 & \forall (u, v) \in E \\
& y'_u \geq 0 & \forall u \in V \\
& y'_U \geq 0 & \forall U \subseteq V, |U| \text{odd}.
\end{array}
$$

Let $x^{(2)}$ and y' be the optimal primal and dual solutions, respectively. Since the matching polytope is Totally Dual Integral (TDI), we assume that y' is integral. Let y be the final dual solution defined by $y_u = y_u^{(1)} + y'_u(1 - f_u)$ for $u \in V$ and $y_U = y'_U \left(1 - \frac{f_U}{(|U|-1)/2}\right)$ for an odd set $U \subseteq V$. We have

$$\begin{aligned}\textstyle\sum_v y_v + \sum_U y_U \left(\frac{|U|-1}{2}\right) &= \textstyle\sum_v y_v^{(1)} + \sum_v y'_v(1 - f_v) + \sum_U y'_U \left(\frac{|U|-1}{2} - f_U\right) \\ &= \sum_{(u,v)\in E^{(1)}} x_{u,v}^{(1)} + \sum_{(u,v)\in E^{(2)}} x_{u,v}^{(2)},\end{aligned}$$

which is the total amount of the fractional matching. Our goal is to show that for every edge $(u, v) \in E^{(1)} \cup E^{(2)}$,

$$y_u + y_v + \sum_{U:\{u,v\}\subseteq U, |U| \text{ odd}} y_U \geq 0.6, \tag{8}$$

which would imply that the fractional matching $x^{(1)} + x^{(2)}$ is at least 0.6 times the offline maximum matching.

Stage* 1 *edges. We prove Eq. (8) for edges $e = (u, v) \in E^{(1)}$. Indeed, we prove that $y_u^{(1)} + y_v^{(1)} \geq 0.6$, where $\{y_u^{(1)}\}$ are the dual variables computed in the first stage. By construction of $y^{(1)}$, $y_v^{(1)} = 0.3$ unless v is an isolated vertex of $G^{(1)}(D)$. Therefore, if none of u and v is an isolated vertex of $G^{(1)}(D)$, $y_u^{(1)} + y_v^{(1)} = 0.6$. The only remaining case is when $u \in A$ and v is an isolated vertex of $G^{(1)}(D)$. In other words, $u \in S_i$ and $v \in T_j$ for some i and j. Recall that $i \geq j$ and $\alpha_i < \alpha_j$ by Lemma 2. Then $y_u^{(1)} + y_v^{(1)} = (0.6 - 0.4\alpha_i) + 0.4\alpha_j \geq 0.6$.

Stage* 2 *edges. Assume towards contradiction that there is an edge $e = (u, v) \in E^{(2)}$ such that Eq. (8) is violated. Since y' is $\{0, 1\}$-valued,

$$\begin{aligned}& y_u^{(1)} + y_v^{(1)} + y'_u(1 - f_u) + y'_v(1 - f_v) + \sum_U y'_U \left(\tfrac{|U|-1}{2} - f_U\right) < 0.6 \\ \Longrightarrow \quad & y_u^{(1)} + y_v^{(1)} + \min_U(1 - f_u, 1 - f_v, (|U| - 1)/2 - f_U) < 0.6.\end{aligned}$$

Therefore, either $y_u^{(1)} + y_v^{(1)} + (1 - f_u) < 0.6$, $y_u^{(1)} + y_v^{(1)} + (1 - f_v) < 0.6$, or there exists U such that $y_u^{(1)} + y_v^{(1)} + (\frac{|U|-1}{2} - f_U) < 0.6$. In particular, if $x^{(1)}$ is not in a matching polytope (i.e., $f_u > 1$ for some u or $f_U > \frac{|U|-1}{2}$ for some odd set U), since $y_u^{(1)} \leq 0.3$ for all $u \in V$, at least one such case must happen. Ruling out these cases proves both the competitive ratio and the fact that $x^{(1)}$ is in a matching polytope.

For the first case, suppose that $y_u^{(1)} + y_v^{(1)} + (1 - f_u) < 0.6$ (the case $y_u^{(1)} + y_v^{(1)} + (1 - f_v) < 0.6$ is handled similarly). We show that $y_u^{(1)} + (1 - f_u) \geq 0.6$ for every $u \in V$, so that this case cannot occur.

Lemma 5. *For every $v \in V$, $y_u^{(1)} + (1 - f_u) \geq 0.6 \Longleftrightarrow f_u - y_u^{(1)} \leq 0.4$.*

Proof. 1. $v \in C$: $y_v^{(1)} = 0.3$, $f_u = 0.6$.
2. $v \subseteq A$: $v \in S_j$ for some j. If $\alpha_j \geq \frac{3}{4}$, $y_v^{(1)} = 0.3$, $f_v = \frac{0.3(\alpha_j+1)}{\alpha_j} \leq 0.7$. Otherwise, $y_v^{(1)} = 0.6 - 0.4\alpha$ and $f_v = 1 - 0.4\alpha$.

3. v is an isolated vertex in $D(G^{(1)})$: $v \in T_j$ for some j. If $\alpha_j \geq \frac{3}{4}$, $y_v^{(1)} = 0.3$, $f_v = 0.3(\alpha_j+1) \leq 0.6$. Otherwise, $y_v^{(1)} = 0.4\alpha$ and $f_v = \alpha - 0.4\alpha^2$, so $f_v - y_v^{(1)} = 0.6\alpha - 0.4\alpha^2 \leq 0.6$.
4. $v \in U$ for an odd component $U \in \hat{D}_l$, but not $v \neq r(U)$: $y_v^{(1)} = 0.3$ and $f_v = 0.6$.
5. $v = r(U)$ for an odd component $U \in \hat{D}_l$: $v \in T_j$ for some j. If $\alpha_j \geq \frac{3}{4}$, $y_v^{(1)} = 0.3$, $f_v = 0.3(\alpha_j+1) \leq 0.6$. Otherwise, $y_v^{(1)} = 0.3$ and $f_v = \alpha - 0.4\alpha^2 + 0.6 - 0.8\alpha = -0.4\alpha^2 + 0.2\alpha + 0.6 \leq 0.625$.

Finally, suppose that there exist two vertices $u, v \in V$ and U such that $\{u, v\} \subseteq U$, $|U|$ is odd, and $y_u^{(1)} + y_v^{(1)} + (\frac{|U|-1}{2} - f_U) < 0.6$. Among all such triples (u, v, U), take one triple that minimizes $|U|$. The following two lemmas show that indeed such U must be contained in one expanding pair (S_j, T_j).

Lemma 6. *U cannot be partitioned into $U_1, U_2, \{u\}$ or $U_1, U_2, \{v\}$ such that $|U_1|, |U_2|$ are odd and there is no edge $(a, b) \in U_1 \times U_2$ with $x_{a,b}^{(1)} > 0$.*

Proof. Assume towards contradiction that U is partitioned into $U_1, U_2, \{u\}$ such that every edge $(a, b) \in U_1 \times U_2$ has $x_{a,b}^{(1)} = 0$. This implies that

$$f_U = f_{U_1} + f_{U_2} + f_u \leq \frac{|U_1|-1}{2} + \frac{|U_2|-1}{2} + f_u = \frac{|U|-1}{2} + f_u - 1.$$

Therefore,

$$\begin{aligned} & y_u^{(1)} + y_v^{(1)} + (\frac{|U|-1}{2} - f_U) < 0.6 \\ \Longrightarrow \quad & y_u^{(1)} + \frac{|U|-1}{2} - f_U < 0.6 \\ \Longrightarrow \quad & y_u^{(1)} + (1 - f_u) < 0.6, \end{aligned}$$

which contradicts Lemma 5. The case that U is partitioned into $U_1, U_2, \{v\}$ follows from the same proof.

Lemma 7. *There exists j such that $U \subseteq S_j \cup T_j$.*

Proof. Let $E_x \subseteq E^{(1)}$ be such that $(a, b) \in E_x$ if and only if $x_{a,b}^{(1)} > 0$, and let $G_x = (V, E_x)$.

Proposition 3. *$G_x(U)$ is connected.*

Proof. First, assume there exists a connected component $U_1 \subsetneq U$ that contains neither u nor v. Since U_1 is disconnected from the rest of U, $f_U = f_{U_1} + f_{U_2}$. Consider $U_2 = U \setminus U_1$. If $|U_1|$ is even and $|U_2|$ is odd,

$$\begin{aligned} y_u^{(1)} + y_v^{(1)} + \frac{|U_2|-1}{2} - f_{U_2} &\leq y_u^{(1)} + y_v^{(1)} + \frac{|U_2|-1}{2} - f_{U_2} + \frac{|U_1|}{2} - f_{U_1} \\ &= y_u^{(1)} + y_v^{(1)} + \frac{|U|-1}{2} - f_U < 0.6, \end{aligned}$$

contradicting the minimality of $|U|$. If $|U_1|$ is odd and $|U_2|$ is even,

$$y_u^{(1)} + y_v^{(1)} + \frac{|U_2|}{2} - f_{U_2} \leq y_u^{(1)} + y_v^{(1)} + \frac{|U_2|}{2} - f_{U_2} + \frac{|U_1|-1}{2} - f_{U_1}$$

$$= y_u^{(1)} + y_v^{(1)} + \frac{|U|-1}{2} - f_U < 0.6.$$

If $|U_2| = \{u, v\}$, this implies $y_u^{(1)} + 1 - f_u < 0.6$, contradicting Lemma 5. If $|U_2| \geq 4$, let $U_3 \subseteq U_2$ such that $\{u, v\} \subseteq U_3$ and $|U_3| = |U_2| - 1$. Since $f_{U_2} \leq f_{U_3} + 1$,

$$y_u^{(1)} + y_v^{(1)} + \frac{|U_3|-1}{2} - f_{U_3} \leq y_u^{(1)} + y_v^{(1)} + \frac{|U_2|}{2} - f_{U_2} < 0.6,$$

contradicting the minimality of $|U|$.

Finally, assume that $G_x(U)$ has two connected components U_1 and U_2 such that $u \in U_1$ and $v \in U_2$. Without loss of generality, suppose $|U_1|$ is even. Then U is partitioned into $U_1 \setminus \{u\}, U_2, \{u\}$ such that $|U_1 \setminus \{u\}|, |U_2|$ are odd and there is no edge $(a, b) \in U_1 \times U_2$ such that $x_{a,b}^{(1)} > 0$. This contradicts Lemma 6.

By the design of Algorithm 3, the only connected components of G_x, besides isolated vertices, are

- Isolated edges (u, v) where $u, v \in C$.
- Odd component $W \in \hat{D}_l$ with $r(W) = *$.
- For each $j = 1$, $S_j \cup T_j \cup (\cup_{W \in \hat{D}_l : r(S) \in T_j} W)$.

If u, v are in C or in $W \in \hat{D}_l$, we have $y_u^{(1)} = y_v^{(1)} = 0.3$, so $y_u^{(1)} + y_v^{(1)} \geq 0.6$. Therefore, the first two cases cannot occur and there must exist j such that $\{u, v\} \subseteq U \subseteq S_j \cup T_j \cup (\cup_{W \in \hat{D}_l : r(S) \in T_j} W)$. The design of Algorithm 3 also ensures that $y_u^{(1)} = y_v^{(1)} = 0.3$ when $\alpha_j \geq \frac{3}{4}$, so we can assume that $\alpha_j < \frac{3}{4}$. Furthermore, at least one of u and v is in D_s, as $y_w^{(1)} = 0.3$ for all $w \in D_l$. Without loss of generality, suppose $u \in D_s$.

Suppose there exists $W \in \hat{D}_l$ such that $r(W) \in T_j$ and $U_W := U \cap (W \setminus \{r(W)\}) \neq \emptyset$. If $r(W) \notin U$, U_W and $U \setminus U_W$ are disconnected ($u \in U \setminus U_W$ so both sets are nonempty), contradicting Proposition 3. If $v \in U_W$, let $v \leftarrow r(W)$. This can be done without loss of generality, since $y_v^{(1)} = y_{R(W)}^{(1)} = 0.3$. In particular, (u, v, U) still satisfies $y_u^{(1)} + y_v^{(1)} + (\frac{|U|-1}{2} - f_U) < 0.6$. We analyze the following two cases.

- Suppose $v = r(W)$. If $|U_W|$ is odd, U is partitioned into $U_W, U \setminus (\{v\} \cup U_W), \{v\}$ such that both $|U_W|, |U \setminus (\{v\} \cup U_W)|$ are odd and there is no edge $(a, b) \in U_W \times \big(U \setminus (\{v\} \cup U_W)\big)$ with $x_{a,b}^{(1)} > 0$. This contradicts Lemma 6. If $|U_W|$ is even, consider $U \setminus U_W$. Since $f_U = f_{U \setminus U_W} + f_{U_W \cup \{v\}}$ and $f_{U_W \cup \{v\}} \leq \frac{|U_W|}{2}$, we have
$$y_u^{(1)} + y_v^{(1)} + \frac{|U \setminus U_W| - 1}{2} - f_{U \setminus U_W} \leq y_u^{(1)} + y_v^{(1)} + \frac{|U|-1}{2} - f_U < 0.6,$$
contradicting the minimality of $|U|$.
- $v \notin U_W \cup \{r(W)\}$. If $|U_W|$ is even, $U \setminus U_W$ contradicts the minimality of $|U|$ as above. If $|U_W|$ is odd, consider $U \setminus (U_W \cup \{r(W)\}$. Since $f_U \leq f_{U \setminus (U_W \cup \{v\})} + f_{U_W} + 1$ and $f_{U_W} \leq \frac{|U_W|-1}{2}$,
$$\begin{aligned} y_u^{(1)} + y_v^{(1)} + \frac{|U \setminus (U_W \cup \{r(W)\})| - 1}{2} - f_{U \setminus (U_W \cup \{r(W)\})} \\ \leq \quad y_u^{(1)} + y_v^{(1)} + \frac{|U|-1}{2} - f_U \quad < \quad 0.6, \end{aligned}$$

contradicting the minimality of $|U|$.

Therefore, every odd component $W \in \hat{D}_l$ intersects with U only in T_j.

Let $s := |U \cap S_j|$ and $t := |U \cap T_j|$. As above we can assume $u \in D_s \cap T_j$ (otherwise $y_u^{(1)} + y_v^{(1)} \geq 0.6$), so that $t \geq 1$. Even if $t = 1$, $v \in S_j$ and $y_u^{(1)} + y_v^{(1)} = 0.4\alpha + (0.6 - 0.4\alpha) = 0.6$. Therefore, $t \geq 2$ and $u, v \in T_j$. Since $f_U \leq \min(s(1 - 0.4\alpha), t(\alpha - 0.4\alpha^2))$,

$$
\begin{aligned}
& y_u^{(1)} + y_v^{(1)} + ((|U-1|)/2 - f_S) < 0.6 \\
\implies \quad & 0.4\alpha + 0.4\alpha + \big((s+t-1)/2 - \min(s(1-0.4\alpha), t(\alpha - 0.4\alpha^2))\big) < 0.6.
\end{aligned}
$$

The following lemma shows that the final inequality cannot be satisfied for every integer $s \geq 1, t \geq 2$, and $\alpha \in (0, \frac{3}{4})$.

Lemma 8. *For any integers $s \geq 1, t \geq 2$ and $\alpha \in (0, \frac{3}{4})$,*

$$0.4\alpha + 0.4\alpha + \left(\frac{s+t-1}{2} - \min(s(1-0.4\alpha), t(\alpha - 0.4\alpha^2))\right) \geq 0.6.$$

Proof. Let Q be the LHS of the inequality. Note that $s(1 - 0.4\alpha) \geq t(\alpha - 0.4\alpha^2)$ if and only if $s \geq t\alpha$. We consider the following three small cases.

1. $s = 1, t = 2, \alpha \leq 0.5$: $Q = 0.4\alpha + 0.4\alpha + (1 - 2(\alpha - 0.4\alpha^2)) = 1 - 1.2\alpha + 0.8\alpha^2 \geq 0.6$.
2. $s = 1, t = 2, \alpha > 0.5$: $Q = 0.4\alpha + 0.4\alpha + (1 - (1 - 0.4\alpha)) = 1.2\alpha > 0.6$.
3. $s = 2, t = 2$: For all α, $s(1 - 0.4\alpha) \geq t(\alpha - 0.4\alpha^2)$, and
$$Q = 0.4\alpha + 0.4\alpha + (1.5 - 2(\alpha - 0.4\alpha^2)) = 1.5 - 1.2\alpha + 0.8\alpha^2 > 1.$$

For general s and t, if $s \geq t\alpha$,

$$Q = 0.4\alpha + 0.4\alpha + \left(\frac{t\alpha + t - 1}{2} - t(\alpha - 0.4\alpha^2)\right) = 0.8\alpha + \frac{t}{2}(1 - \alpha + 0.8\alpha^2) - 0.5.$$

When $t = 2$, $Q \geq 0.6$ by Case 1. above. If $t \geq 3$, for any $\alpha \in [0, 1]$, $\frac{t}{2}(1 - \alpha + 0.8\alpha^2) \geq 1.1$.

If $s < t\alpha$,

$$Q = 0.4\alpha + 0.4\alpha + \left(\frac{s + \frac{s}{\alpha} - 1}{2} - s(1 - 0.4\alpha)\right) = 0.8\alpha + \frac{s}{2}\left(\frac{1}{\alpha} - 1 + 0.8\alpha\right) - 0.5.$$

When $s = 1, 2$, we have $Q \geq 0.6$ by Cases 2 and 3 above. If $s \geq 3$, for any $\alpha \in [0, 1]$, $\frac{s}{2}\left(\frac{1}{\alpha} - 1 + 0.8\alpha\right) \geq 1.1$.

Therefore, there is no $e \in E^{(2)}$ such that Eq. (8) is violated and our algorithm guarantees 0.6 competitive ratio.

5.3 s-stage Integral General Matching

In this section, we prove Theorem 3 for s-stage. Let $p_1 := 1$ and $p_s := \frac{1}{2} + \frac{(p_{s-1} - \frac{1}{2})}{20} = \frac{1}{2} + \frac{1}{2 \cdot 20^{s-1}}$. Algorithm 4 presents our algorithm for s-stage integral matching for general graphs with the competitive ratio p_s.

Algorithm 4 s-stage Integral Matching Algorithm

Compute the Edmonds-Gallai decomposition (C, A, D). Let $X \leftarrow \emptyset$.
Let M_C be a perfect matching for $G^{(1)}(C)$. Let $X \leftarrow X \cup M_C$ with probability q. Let $y_u \leftarrow \frac{q}{2}$ for all $u \in C$
Construct the matching skeleton with expanding pairs $\{(S_j, T_j)\}$ and expansion α_j for G' defined above
Let $p \leftarrow p_{s-1}$, and $q \leftarrow p_s$.

for each j **do**
  Let $\mathcal{D}_j$ be a distribution over matchings between (S_j, T_j) that matches all vertices of S_j with probability 1 and each vertex of T_j with probability α_j
  Independently generate a uniformly random real $c_j \in [0, 1]$
  if $\alpha_j \geq \frac{pq}{2p-q-pq}$ **then**
    if $c_j \leq \frac{q(\alpha_j+1)}{2\alpha_j}$ **then**
      Independently sample a random matching M_j from $\mathcal{D}_j$.
      $X \leftarrow X \cup M_j$.
    end if
    $y_u \leftarrow \frac{q}{2}$ for all $u \in S_j \cup T_j$
  else
    Let $\delta_j := \frac{p-\alpha_j(p-q)}{p+\alpha_j(1-p)}$, $\epsilon_j = \frac{pq-\alpha_j(p-q)}{p-\alpha_j(p-q)}$.
    if $c_j \leq \delta_j$ **then**
      Independently sample a random matching M_j from $\mathcal{D}_j$.
      $X \leftarrow X \cup M_j$.
    end if
    $y_u \leftarrow \delta_j \epsilon_j$ for all $u \in S_j$
    $y_u \leftarrow \delta_j(1-\epsilon_j)\alpha_j$ for all $u \in T_j$
  end if
end for

for each odd component $U \in \hat{D}_l$ **do**
  For $v \in U$, let M_v be a perfect matching in $G(U \setminus v)$
  Independently generate a uniformly random real $c_U \in [0, 1]$
  if $r(U) = *$ **then**
    if $c_U \leq \frac{q|U|}{2(|U|-1)}$ **then** Sample a random $v \in U$ and $X \leftarrow X \cup M_v$ **end if**
  else
    Let j be such that $r(U) \in T_j$ for some j.
    if $\alpha_j \geq \frac{pq}{2p-q-pq}$ **then**
      if $c_U \leq \frac{q|U|}{2(|U|-1)}$ **then** $X \leftarrow X \cup M_{r(U)}$ **end if**
    else
      if $r(U)$ is matched **then** $X \leftarrow X \cup M_{r(U)}$.
      else
        if $c_U \in [0, \frac{q-2\delta_j\alpha_j(1-\epsilon_j)}{1-\delta_j\alpha_j}]$ **then** Sample a random $v \in U \setminus r(U)$ and $X \leftarrow X \cup M_v$ **end if**
        if $c_U \in [\frac{q-2\delta_j\alpha_j(1-\epsilon_j)}{1-\delta_j\alpha_j}, \frac{q-\delta_j\alpha_j+\frac{q-2\delta_j\alpha_j(1-\epsilon_j)}{|U|-1}}{1-\delta_j\alpha_j}]$ **then** $X \leftarrow X \cup M_{r(U)}$ **end if**
      end if
    end if
  end if
  $y_u \leftarrow \frac{q}{2}$ for all $u \in U$.
end for

Our algorithm is only stated for the first stage. In stage t, we view it as the first stage of the $(s-t+1)$-stage problem. As for bipartite matching, we use induction to prove our competitive ratios. Consider $s \geq 2$ and suppose that the competitive ratio for $(s-1)$-stage matching is at least $p := p_{s-1} > 0.5$. Given a s-stage graph $G = (V, E^{(1)} \cup \cdots \cup E^{(s)})$, let $X^{(1)}$ be the random matching we choose in the first stage and X' be the random matching we choose in the later $(s-1)$ stages. In addition to the matching $X^{(1)}$, the algorithm also constructs dual solution $\{y_v^{(1)}\}_{v\in V}$ such that $\mathsf{E}[|X^{(1)}|] = \sum_v y_v^{(1)}$. Let $E' := E^{(2)} \cup \cdots \cup E^{(s)}$. Applying the competitive ratio of p to the last $(s-1)$-stage graph $G' := (V, E') \setminus X^{(1)}$, there exist an integral dual solution $\{y'_v\}_{v\in V} \cup \{y'_S\}_{|S| \text{ odd}}$ such that $\sum_v y'_v + \sum_S y'_S \frac{|S|-1}{2} = |X'|$ and $y'_u + y'_v + \sum_{S:\{u,v\}\subseteq S} y'_S \geq p$ for every edge $(u,v) \in G'$. As in the bipartite case, we construct the final dual solution y as $y := y^{(1)} + \mathsf{E}[y']$.

$$\sum_v y_v + \sum_S y_S \cdot \frac{|S|-1}{2} = \sum_v y_v^{(1)} + \sum_v \mathsf{E}[y'_v] + \sum_S \mathsf{E}[y'_S \frac{|S|-1}{2}] = \mathsf{E}[|X^{(1)}| + |X'|].$$

Let $q := p_s$. The following lemma finishes the proof that the competitive ratio for s-stage integral matching is at least q.

Lemma 9. *For every edge $(u,v) \in E$, $y_u + y_v + \sum_{S:\{u,v\}\subseteq S} y_S \geq q$.*

Proof. For any $(u,v) \in E^{(1)}$, unless one of them is D_s, $y_u = y_v = \frac{q}{2}$. Therefore, one of them (say u) must be in D_s. Let j be such that $u \in D_s \cap T_j$. Since $(u,v) \in E^{(1)}$, $v \in S_i$ for some $i \geq j$ and

$$\begin{aligned} y_u + y_v &= \delta_i \epsilon_i + \delta_j (1-\epsilon_j)\alpha_j \\ &= \frac{pq - \alpha_i(p-q)}{p + \alpha_i(1-p)} + \frac{(p-pq)\alpha_j}{p+\alpha_j(1-p)} \geq \frac{pq - \alpha_j(p-q)}{p+\alpha_j(1-p)} + \frac{(p-pq)\alpha_j}{p+\alpha_j(1-p)} = q. \end{aligned}$$

Consider $(u,v) \in E'$. First, assume that both u and v are in $V^{(1)}$. By construction of $y^{(1)}$, at least one of them (say u) must be in $D_s \cap T_j$ for some j with $\alpha_j < \frac{pq}{2p-q-pq}$. We consider the following cases.

– $v \in C$ or $v \in U$ for some odd component $U \in \hat{D}_l$ with $r(U) \notin S_j$: In this case, $y_v = \frac{q}{2}$, and the event v is matched by $X^{(1)}$ is independent from whether u is matched or not. For $v \in C$ or $v \neq r(U)$, $\Pr[v \text{ is matched}] = q$. Therefore,

$$\begin{aligned} & y_u + y_v + \sum_S y_S \\ \geq\ & y_u + y_v + p \cdot \Pr[\text{neither } u \text{ nor } v \text{ is matched}] \\ =\ & \frac{q}{2} + \frac{(p-pq)\alpha_j}{p+\alpha_j(1-p)} + p(1-q)\left(1 - \frac{\alpha_j p - \alpha_j^2(p-q)}{p+\alpha_j(1-p)}\right) \\ \geq\ & q. \end{aligned}$$

The last inequality is verified as the Case 1 of §C.
If $v = r(U) \in S_i$ for some $i \neq j$,

$$\begin{aligned} \Pr[v \text{ is matched}] &= q - 2\delta_i\alpha_i(1-\epsilon_i) + \delta_i\alpha_i = q - \delta_i(\alpha_i)(1-2\epsilon_i) \\ &= q - \frac{\alpha_i(p + \alpha_i(p-q) - 2pq)}{p+\alpha_i(1-p)}. \end{aligned}$$

Therefore,

$$
\begin{aligned}
& y_u + y_v + \sum_S y_S \\
& \geq y_u + y_v + p \cdot \Pr[\text{neither } u \text{ nor } v \text{ is matched}] \\
& = \frac{q}{2} + \frac{(p-pq)\alpha_j}{p+\alpha_j(1-p)} + p\left(1 - q + \frac{\alpha_i(p+\alpha_i(p-q)-2pq)}{p+\alpha_i(1-p)}\right)\left(1 - \frac{\alpha_j p - \alpha_j^2(p-q)}{p+\alpha_j(1-p)}\right) \\
& \geq q.
\end{aligned}
$$

The last inequality is verified as Case 2 in §C.

- $v \in S_i$ for $i \neq j$: Note that the case $i \geq j$ is already covered, so we can assume $i < j$. Still v and u are matched independently. If $\alpha_i \geq \frac{pq}{2p-q-pq}$, $y_v = \frac{q}{2}$ and $\Pr[v \text{ is matched}] \leq \frac{2p-q}{2p}$ (maximized when $\alpha_j = \frac{pq}{2p-q-pq}$), so

$$
\begin{aligned}
& y_u + y_v + \sum_S y_S \\
& \geq y_u + y_v + p \cdot \Pr[\text{neither } u \text{ nor } v \text{ is matched}] \\
& = \frac{q}{2} + \frac{(p-pq)\alpha_j}{p+\alpha_j(1-p)} + p\left(1 - \frac{2p-q}{2p}\right)\left(1 - \frac{\alpha_j p - \alpha_j^2(p-q)}{p+\alpha_j(1-p)}\right) \\
& \geq q.
\end{aligned}
$$

The last inequality is verified as the Case 3 of §C. If $\alpha_i < \frac{pq}{2p-q-pq}$ $y_v = \frac{pq-\alpha_i(p-q)}{p+\alpha_i(1-p)}$ and $\Pr[v \text{ is matched}] = \delta_i = \frac{p-\alpha_i(p-q)}{p+\alpha_i(1-p)}$, so

$$
\begin{aligned}
& y_u + y_v + \sum_S y_S \\
& \geq y_u + y_v + p \cdot \Pr[\text{neither } u \text{ nor } v \text{ is matched}] \\
& = \frac{pq-\alpha_i(p-q)}{p+\alpha_i(1-p)} + \frac{(p-pq)\alpha_j}{p+\alpha_j(1-p)} + p(1 - \frac{p-\alpha_i(p-q)}{p+\alpha_i(1-p)})(1 - \frac{\alpha_j p - \alpha_j^2(p-q)}{p+\alpha_j(1-p)}) \\
& \geq q.
\end{aligned}
$$

The last inequality is verified as the Case 4 of §C.

- $v \in T_j \cap D_s$: $y_v = y_u = \delta_j(1-\epsilon_j)\alpha_j = \frac{(p-pq)\alpha_j}{p+\alpha_j(1-p)}$. If $c_j > \delta_j = \frac{p-\alpha_j(p-q)}{p+\alpha_j(1-p)}$, both u and v are unmatched. Given that $c_j \leq \frac{p-\alpha_j(p-q)}{p+\alpha_j(1-p)}$, $\Pr[v \text{ or } u \text{ is matched}] \leq 2\alpha_j$ by union bound. Therefore,

$$
\begin{aligned}
& y_u + y_v + \sum_S y_S \\
\geq\ & \frac{2(p-pq)\alpha_j}{p+\alpha_j(1-p)} + p\left(1 - \frac{p-\alpha_j(p-q)}{p+\alpha_j(1-p)}\right) + \left(\frac{p-\alpha_j(p-q)}{p+\alpha_j(1-p)}\right)\max\{0, 1-2\alpha_j\} \\
\geq\ & q.
\end{aligned}
$$

The last inequality is verified as the Case 5 of §C.

- $v \in U$ for some odd component $U \in \hat{D}_l$ with $r(U) \in S_j$: $y_v = \frac{q}{2}$. Let F_r, F_u be the event that $r(U), u$ is matched by (S_j, T_j) respectively. $\Pr[F_r] = \Pr[F_u] = \delta_j\alpha_j =$

$\frac{\alpha_j p-\alpha_j^2(p-q)}{p+\alpha_j(1-p)}$. The design of Algorithm 4 ensures that v is matched whenever $r(U)$ is matched. Given $\overline{F_r}$, v and u are matched independently, and

$$\Pr[F_u|\overline{F_r}] \leq \frac{\delta_j\alpha_j}{1-\delta_j\alpha_j} = \frac{\frac{\alpha_j p-\alpha_j^2(p-q)}{p+\alpha_j(1-p)}}{1-\frac{\alpha_j p-\alpha_j^2(p-q)}{p+\alpha_j(1-p)}} = \frac{\alpha_j p-\alpha_j^2(p-q)}{p+\alpha_j(1-2p)+\alpha_j^2(p-q)}.$$

If $v = r(U)$,

$$\Pr[v \text{ is matched}|\overline{F_r}] = \frac{q-2\delta_j(1-\epsilon_j)\alpha_j}{1-\delta_j\alpha_j} = \frac{q-2\frac{\alpha_j p-\alpha_j pq}{p+\alpha_j(1-p)}}{1-\frac{\alpha_j p-\alpha_j^2(p-q)}{p+\alpha_j(1-p)}}$$
$$= \frac{pq+\alpha_j(q-2p+pq)}{p+\alpha_j(1-2p)+\alpha_j^2(p-q)},$$

and

$$\begin{aligned}
& y_u + y_v + \sum_S y_S \\
& \geq y_u + y_v + p\cdot\Pr[\overline{F_r}](1-\Pr[F_u|\overline{F_r}])(1-\Pr[v \text{ is matched}|\overline{F_r}]) \\
& \geq \frac{q}{2} + \frac{(p-pq)\alpha_j}{p+\alpha_j(1-p)} + \\
& \quad p\cdot\left(1-\frac{\alpha_j p-\alpha_j^2(p-q)}{p+\alpha_j(1-p)}\right)\cdot\left(1-\frac{\alpha_j p-\alpha_j^2(p-q)}{p+\alpha_j(1-2p)+\alpha_j^2(p-q)}\right)\cdot \\
& \qquad \left(1-\frac{pq+\alpha_j(q-2p+pq)}{p+\alpha_j(1-2p)+\alpha_j^2(p-q)}\right) \\
& \geq q.
\end{aligned}$$

The last inequality is verified as the Case 6 of §C.

If $v \neq r(U), \Pr[v \text{ is matched}|\overline{F_r}] = \frac{q-\delta_j\alpha_j}{1-\delta_j\alpha_j} = \frac{q-\frac{\alpha_j p-\alpha_j^2(p-q)}{p+\alpha_j(1-p)}}{1-\frac{\alpha_j p-\alpha_j^2(p-q)}{p+\alpha_j(1-p)}} = \frac{pq+\alpha_j(q-p-pq)+\alpha_j^2(p-q)}{p+\alpha_j(1-2p)+\alpha_j^2(p-q)}$,

and

$$\begin{aligned}
& y_u + y_v + \sum_S y_S \\
& \geq y_u + y_v + p\cdot\Pr[\overline{F_r}](1-\Pr[F_u|\overline{F_r}])(1-\Pr[v \text{ is matched}|\overline{F_r}]) \\
& \geq \frac{q}{2} + \frac{(p-pq)\alpha_j}{p+\alpha_j(1-p)} + \\
& \quad p\cdot\left(1-\frac{\alpha_j p-\alpha_j^2(p-q)}{p+\alpha_j(1-p)}\right)\cdot\left(1-\frac{\alpha_j p-\alpha_j^2(p-q)}{p+\alpha_j(1-2p)+\alpha_j^2(p-q)}\right)\cdot \\
& \qquad \left(1-\frac{pq+\alpha_j(q-p-pq)+\alpha_j^2(p-q)}{p+\alpha_j(1-2p)+\alpha_j^2(p-q)}\right) \\
& \geq q.
\end{aligned}$$

The last inequality is verified as the Case 6 of §C.

If one of them (say u) is not incident on any edge of $E^{(1)}$ and thus not in $V^{(1)}$, it is a special case of one of Case 1, 2, 3, and 4 with $\alpha_j = 0$.

References

1. G. Aggarwal, G. Goel, C. Karande, and A. 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. A. Blum, T. Sandholm, and M. Zinkevich. Online algorithms for market clearing. *Journal of the ACM (JACM)*, 53(5):845–879, 2006.
3. A. Borodin and R. El-Yaniv. *Online computation and competitive analysis*. cambridge university press, 2005.
4. N. Buchbinder, K. Jain, and J. S. Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *European Symposium on Algorithms*, pages 253–264. Springer, 2007.
5. N. R. Devanur and T. 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.
6. N. R. Devanur, K. Jain, and R. D. Kleinberg. Randomized primal-dual analysis of ranking for online bipartite matching. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 101–107, 2013.
7. L. Epstein, A. Levin, J. Mestre, and D. Segev. Improved approximation guarantees for weighted matching in the semi-streaming model. *SIAM Journal on Discrete Mathematics*, 25(3):1251–1265, 2011.
8. L. Epstein, A. Levin, D. Segev, and O. Weimann. Improved bounds for online preemptive matching. In *30th International Symposium on Theoretical Aspects of Computer Science*, pages 389–399, 2013.
9. J. Feigenbaum, S. Kannan, A. McGregor, S. Suri, and J. Zhang. On graph problems in a semi-streaming model. *Elsevier Theoretical Computer Science*, 348(2):207–216, 2005.
10. J. Feldman, A. Mehta, V. 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.
11. A. Fiat. Online algorithms: The state of the art (lecture notes in computer science). 1998.
12. A. Goel, M. Kapralov, and S. Khanna. On the communication and streaming complexity of maximum bipartite matching. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 468–485. SIAM, 2012.
13. G. Goel and A. Mehta. Online budgeted matching in random input models with applications to adwords. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 982–991. Society for Industrial and Applied Mathematics, 2008.
14. D. Golovin, V. Goyal, V. Polishchuk, R. Ravi, and M. Sysikaski. Improved approximations for two-stage min-cut and shortest path problems under uncertainty. *Mathematical Programming*, 149(1-2):167–194, 2015.
15. G. P. Guruganesh and S. Singla. Online Matroid Intersection: Beating Half for Random Arrival. *arXiv preprint arXiv:1512.06271*, 2015.
16. B. Haeupler, V. S. Mirrokni, and M. Zadimoghaddam. Online stochastic weighted matching: Improved approximation algorithms. In *International Workshop on Internet and Network Economics*, pages 170–181. Springer, 2011.
17. M. Kapralov. Better bounds for matchings in the streaming model. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1679–1697. SIAM, 2013.

18. C. Karande, A. Mehta, and P. Tripathi. Online bipartite matching with unknown distributions. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, pages 587–596. ACM, 2011.
19. R. M. Karp, U. V. Vazirani, and V. 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, 1990.
20. S. Khot and O. Regev. Vertex cover might be hard to approximate to within 2- ε. *Journal of Computer and System Sciences*, 74(3):335–349, 2008.
21. C. Konrad, F. Magniez, and C. Mathieu. Maximum matching in semi-streaming with few passes. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 231–242. Springer, 2012.
22. N. Korula and M. Pál. Algorithms for secretary problems on graphs and hypergraphs. In *International Colloquium on Automata, Languages and Programming*, pages 508–520. Springer, 2009.
23. L. Lovász and M. D. Plummer. Matching Theory. *Ann. Discrete Math*, 29, 1986.
24. M. Mahdian and Q. 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, 2011.
25. V. H. Manshadi, S. O. Gharan, and A. Saberi. Online stochastic matching: Online actions based on offline statistics. *Mathematics of Operations Research*, 37(4):559–573, 2012.
26. A. Mehta. Online matching and ad allocation. *Theoretical Computer Science*, 8(4):265–368, 2012.
27. A. Mehta and D. Panigrahi. Online matching with stochastic rewards. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 728–737. IEEE, 2012.
28. A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. Adwords and generalized online matching. *Journal of the ACM (JACM)*, 54(5):22, 2007.
29. A. Mehta, B. Waggoner, and M. Zadimoghaddam. Online stochastic matching with unequal probabilities. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1388–1404. SIAM, 2015.
30. C. Swamy and D. B. Shmoys. Approximation algorithms for 2-stage stochastic optimization problems. *ACM SIGACT News*, 37(1):33–46, 2006.
31. V. V. Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.
32. Y. Wang and S. C.-w. Wong. Two-sided online bipartite matching and vertex cover: Beating the greedy algorithm. In *International Colloquium on Automata, Languages, and Programming*, pages 1070–1081. Springer, 2015.

A Illustrative Examples

Theorem 9. *For two-stage fractional (thus integral) bipartite matching, there exists a graph where picking a maximum matching with probability δ for any $\delta \in (0, 1)$ fails to achieve competitive ratio $\frac{2}{3}$ regardless how the maximum matching is chosen.*

Proof. We first consider the graph $G^{(1)} = K_{2,3}$, where $\{u_1, u_2\}$ form the left side and $\{v_1, v_2, v_3\}$ form the right side. We consider the following $\mathbf{LP}$ from Theorem 4 exactly captures the competitive ratio.

$$\max \qquad \alpha$$

$$
\begin{aligned}
\text{s.t.} \qquad & f_u = \sum_{v \in N_1(u)} x^{(1)}_{u,v} && \forall u \in V \\
& f_u \le 1, && \forall u \in V \\
& \sum_{(u,v) \in E^{(1)}} x^{(1)}_{u,v} \ge \sum_u y^{(1)}_u, && \\
& y^{(1)}_u + y^{(1)}_v \ge \alpha, && \forall (u,v) \in E^{(1)} \qquad (9) \\
& y^{(1)}_u \ge f_u - (1-\alpha), && \forall u \in V \qquad (10) \\
& x^{(1)}_{u,v}, y^{(1)}_u \ge 0, && \forall u, v \in V
\end{aligned}
$$

Suppose that in the first stage, the algorithm outputs a maximum matching with probability exactly $\frac{2}{3}$. Let p_1, p_2, p_3 be the probability that the algorithm outputs a maximum matching missing v_1, v_2, v_3 respectively. $p_1 + p_2 + p_3 = \frac{2}{3}$. Since every maximum matching involves exactly two edges matching u_1 and u_2, $\sum_{u,v \in E^{(1)}} x^{(1)}_{u,v} = \frac{4}{3}$, $f_{u_1} = f_{u_2} = \frac{2}{3}$, and $f_{v_1} + f_{v_2} + f_{v_3} = \frac{4}{3}$.

We now show that it is impossible to construct $y^{(1)}$ that satisfies (9) and (10). From (9), $y_{v_i} \ge \frac{2}{3} - \min(y_{u_1}, y_{u_2})$ for all i. Summing over i and using $\min(y_{u_1}, y_{u_2}) \le \frac{y_{u_1} + y_{u_2}}{2}$,

$$
\begin{aligned}
& \sum_i y_{v_i} \ge 2 - 3 \cdot \frac{y_{u_1} + y_{u_2}}{2} \\
\iff \quad & 3 \cdot \frac{y_{u_1} + y_{u_2}}{2} + \sum_i y_{v_i} \ge 2.
\end{aligned}
$$

On the other hand,

$$
\sum_u y_u + \sum_v y_v \le \sum_e x_e = \frac{4}{3}.
$$

This is only possible when $y_{u_1} + y_{u_2} = \frac{4}{3}$. and $y_{v_1} = y_{v_2} = y_{v_3} = 0$. However, since $f_{v_1} + f_{v_2} + f_{v_3} = \frac{4}{3}$, there exists i such that $f_{v_i} \ge \frac{4}{9}$. For that i, (10) is violated.

To finish the proof of the theorem, consider $G^{(1)}$ with disjoint $K_{2,3}$ and a single edge. For the single edge, the only possible strategy is to match with probability $\frac{2}{3}$, but it does not work for $K_{2,3}$ to achieve the competitive ratio of $\frac{2}{3}$.

Theorem 10. *Three-stage fractional (thus integral) matching has competitive ratio strictly less than* $2/3$*.*

Proof. The first stage graph $G^{(1)}$ consists of two disjoint edges $e_1 = (u_1, v_1)$ and $e_2 = (u_2, v_2)$. In order to achieve the competitive ratio $\frac{2}{3}$, the only strategy is to match both of them with value $\frac{2}{3}$. The second stage graph $G^{(2)}$ introduces a new vertex v_3 and two edges $e_3 = (u_1, v_3)$ and $e_4 = (u_2, v_3)$. Given the fixed strategy for the first stage and $G^{(2)}$, the following $\mathbf{LP}$ from Theorem 4 exactly captures the competitive ratio.

$$
\max \qquad \alpha
$$

$$
\begin{aligned}
\text{s.t.} \quad & f_u = \sum_{v \in N_2(u)} x_{u,v} && \forall u \in V \\
& f_u \le 1, && \forall u \in V \\
& x_{e_1} + x_{e_2} + x_{e_3} + x_{e_4} \ge \sum_u y_u, && && (11)\\
& y_u + y_v \ge \alpha, && \forall (u,v) \in E^{(2)} && (12)\\
& y_u \ge f_u - (1-\alpha), && \forall u \in V && (13)\\
& x_{e_1} = x_{e_2} = \frac{1}{3}, \\
& x_{u,v}, y_u \ge 0, && \forall u, v \in V
\end{aligned}
$$

Suppose (x, y) is a feasible solution that achieves $\alpha = \frac{2}{3}$. (13) implies that $y_{v_1} \ge \frac{1}{3}$ and $y_{v_2} \ge \frac{1}{3}$, and $y_{u_1} \ge \frac{1}{3} + x_{e_3}$, $y_{u_2} \ge \frac{1}{3} + x_{e_4}$. Since

$$y_{v_1} + y_{v_2} + y_{u_1} + y_{u_2} \ge \frac{2}{3} + x_{e_3} + x_{e_4} = x_{e_1} + x_{e_2} + x_{e_3} + x_{e_4} \ge \sum_u y_u,$$

(the last ineqaulity implies by (11)) every inequality above must hold as an equality. In particular, $y_{v_3} = 0$. for e_3 and e_4 implies that $y_{u_1}, y_{u_2} \ge \frac{2}{3}$, which implies that $x_{e_3}, x_{e_4} \ge \frac{1}{3}$. This implies that $f_{v_3} \ge \frac{2}{3}$. (13) is violated for v_3, contradicting that (x, y) is feasible for $\alpha = \frac{2}{3}$.

Theorem 11. *For two-stage integral matching in general graphs, no algorithm can be* $2/3$*-competitive.*

Proof. First, we consider a simpler case when $G^{(1)}$ consists two edges $e_1 = (u, v_1)$ and $e_2 = (u, v_2)$.

Lemma 10. *To achieve the competitive ratio* $\frac{2}{3}$*, the only possible strategy is to match both edges w.p. exactly* $\frac{1}{3}$.

Proof. Let x_i be the probability that e_i is matched. Without loss of generality, assume $x_1 \ge x_2$. If $x_1 + x_2 < \frac{2}{3}$, having $G^{(2)}$ as an empty graph makes the competitive ratio is less than $\frac{2}{3}$. If $x_1 + x_2 > \frac{2}{3}$, let $G^{(2)}$ have two edges $e_3 = (v_1, v_2)$ and $e_4 = (u, w)$ for a new vertex w. $G^{(1)} \cup G^{(2)}$ has a matching of size 2, but the expected value of our strategy is $x_1 + x_2 + 2 \times (1 - x_1 + x_2) = 2 - x_1 - x_2 < \frac{2}{3} \cdot 2$. Therefore $x_1 + x_2 = \frac{2}{3}$. Finally, suppose $x_1 > \frac{1}{3} > x_2$. let $G^{(2)}$ have one edge $e_3 = (v_1, w)$ for a new vertex w. $G^{(1)} \cup G^{(2)}$ has a matching of size 2, but the expected value of our strategy is $x_1 + x_2 + (1 - x_1) = 1 + x_2 < \frac{2}{3} \cdot 2$. Therefore, $x_1 = x_2 = \frac{1}{3}$.

To prove the theorem, $G^{(1)}$ consists of two disjoint copies of the same graph above. Formally it has 4 edges $(u_1, v_1), (u_1, v_2), (u_2, v_3), (u_2, v_4)$. From the above lemma, every edge must be matched with probability exactly $\frac{1}{3}$. No matter how we correlate these four edges, there always exist $i \in \{1, 2\}$ and $j \in \{3, 4\}$ such that $\delta := \Pr[e_i, e_j \text{ are both picked}] < \frac{1}{3}$. The second stage graph $G^{(2)}$ consists (v_i, v_j). $G^{(1)} \cup G^{(2)}$ has a matching of size 3. The probability that we can increase the matching size by 1 in the second stage is

$$\Pr[\text{None of } e_i, e_j \text{ is picked}] = 1 - \left(\frac{2}{3} - \delta\right) < \frac{2}{3}.$$

Therefore, the expected size of the matching is less than 2.

Theorem 12. *There is a gap between two-stage fractional bipartite matching and two-stage integral bipartite matching. That is not all two-stage bipartite graphs admit the same competitive ratio for both two-stage fractional and integral bipartite matching problem.*

Proof. Consider $G^{(1)}$ to be a bipartite graph on vertices $(\{u_1, u_2, u_3\}, \{v_1, v_2, v_3\})$. The edges are (u_3, v_1), (u_3, v_2), (v_3, u_1), and (v_3, u_2). Using our $\mathbf{LP}$ from Theorem 4, we get that for two-stage fractional matching, the optimal competitive ratio is $5/7 \approx 0.71$. On the other hand, for two-stage integral matching one can show that every algorithm is strictly less than $5/7$ competitive. This is because $G^{(2)}$ could be empty, or $G^{(2)}$ could consist of, along with edges (u_3, v_4) and (u_4, v_3), either edges $\{(u_1, v_1), (u_2, v_2)\}$ or edges $\{(u_1, v_2), (u_2, v_1)\}$. The integral algorithm has no way to differentiate between the two cases and has to guess randomly. Considering all possible cases shows that a randomized integral algorithm can never match the performance of fractional matching, and is always less than 0.7 competitive.

B Online Bipartite Vertex Cover

The two-stage online bipartite vertex cover problem reveals edges of a bipartite graph $G = ((U_1, U_2), E)$ in two stages $E^{(1)}$ and $E^{(2)}$. At the end of Stage 1, the algorithm has to irrevocably pick a (fractional) vertex cover. Then edges $E^{(2)}$ are revealed and the algorithm, which is not allowed to drop (decrease) picked vertices, has to pick a minimum vertex cover of $E^{(1)} \cup E^{(2)}$.

The proof of the following theorem is similar to that of Theorem 4 for two-stage bipartite matching.

Theorem 13. *The following linear program exactly captures two-stage fractional vertex cover.*

$$\begin{aligned}
\min \quad & \beta \\
\text{s.t.} \quad & \sum_{v \in N_1(u)} x_{u,v} \le y_u + \beta - 1, && \forall u \in V && (14)\\
& \sum_{(u,v) \in E^{(1)}} x_{u,v} \ge \sum_u y_u, && && (15)\\
& y_u + y_v \ge 1, && \forall (u,v) \in E^{(1)} && (16)\\
& x_{u,v}, y_u \ge 0, && \forall u, v \in V && (17)
\end{aligned}$$

The above theorem directly implies the following lemma.

Corollary 1. *There exists an instance optimal algorithm for the two-stage fractional bipartite vertex cover problem.*

We can also extend our s-stage bipartite matching result to s-stage bipartite vertex cover.

Corollary 2. *There exists a* $\frac{1}{2} + \frac{1}{2^{s+1}-2}$ *competitive algorithm for the* s*-stage bipartite vertex cover problem.*

Proof. We note that in the proof of Theorem 2 for s-stage bipartite matching, we constructed an online fractional vertex cover solution in parallel. Unlike matching, there is a rounding scheme that converts any given algorithm for online fractional vertex cover to an algorithm for online vertex cover in bipartite graphs [32].

C Missing Inequalities

We verify 8 inequalities using Mathematica :

Case 0 is for Section 4 and Cases 1 – 7 are for Section 5. They are verified using Reduce [] function. We can easily check from the output that each expression is nonnegative in the desired region and becomes 0 only at the boundaries.

Case 0 (for Section 4)

```
c0 = (-1 + b) ^2 * p^2 * (-1 + 2 p) +
  a^2 * p * (2 b^2 (-1 + p) ^2 + p (-1 + 2 p) + b (-1 + 5 p - 4 p^2)) +
  a (2 (1 - 2 p) p^2 - b^2 * p * (1 - 5 p + 4 p^2) + b (1 - 7 p^2 + 8 p^3))
```

$$(-1+b)^2 p^2 (-1+2p) + a^2 p \left(2 b^2 (-1+p)^2 + p(-1+2p) + b\left(-1+5p-4p^2\right)\right) + a\left(2(1-2p)p^2 - b^2 p\left(1-5p+4p^2\right) + b\left(1-7p^2+8p^3\right)\right)$$

```
Reduce[c0 ≤ 0 && a ≥ 0 && a ≤ 1 && b ≥ 0 && b ≤ 1 && p ≥ 1 / 2 && p ≤ 1, {a, b, p}, Reals]
```

$$\left(a == 0 \,\&\&\, \left(\left(0 \le b < 1 \,\&\&\, p == \frac{1}{2}\right) \,||\, \left(b == 1 \,\&\&\, \frac{1}{2} \le p \le 1\right)\right)\right) \,||\, \left(0 < a < 1 \,\&\&\, \left(\left(b == 0 \,\&\&\, p == \frac{1}{2}\right) \,||\, (b == 1 - a \,\&\&\, p == 1)\right)\right) \,||\, \left(a == 1 \,\&\&\, b == 0 \,\&\&\, \frac{1}{2} \le p \le 1\right)$$

Cases 1 – 7 (for Section 5)

```
q = 1 / 2 + (p - (1 / 2)) / 20
```

$$\frac{1}{2} + \frac{1}{20}\left(-\frac{1}{2} + p\right)$$

Case 1

```
c1 = q / 2 + ((p - p * q) * a) / (p + a * (1 - p)) +
  p * (1 - q) * (1 - (a * p - a^2 * (p - q)) / (p + a * (1 - p)))
```

$$\frac{1}{2}\left(\frac{1}{2} + \frac{1}{20}\left(-\frac{1}{2} + p\right)\right) + \frac{a\left(p - \left(\frac{1}{2} + \frac{1}{20}\left(-\frac{1}{2} + p\right)\right) p\right)}{a(1-p) + p} + \left(\frac{1}{2} + \frac{1}{20}\left(\frac{1}{2} - p\right)\right) p \left(1 - \frac{a p - a^2\left(-\frac{1}{2} + \frac{1}{20}\left(\frac{1}{2} - p\right) + p\right)}{a(1-p) + p}\right)$$

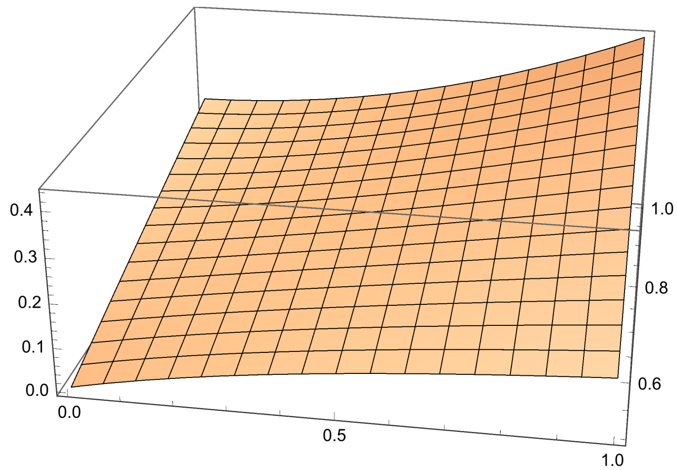
```
Simplify[c1 - q]
```

$$\left(19 a^2 p \left(21 - 44 p + 4 p^2\right) + 20 p \left(19 - 40 p + 4 p^2\right) - 20 a \left(-19 + 101 p - 90 p^2 + 8 p^3\right)\right) / (1600 (a (-1 + p) - p))$$

```
Reduce[c1 - q ≤ 0 && a ≥ 0 && a ≤ 1 && p ≥ 1 / 2 && p ≤ 1, {a, p}, Reals]
```

$$a == 0 \,\&\&\, p == \frac{1}{2}$$

```
Plot3D[c1 - q, {a, 0, 1}, {p, 0.5, 1}]
```



Case 2

```
c2 = q / 2 + ((p - p * q) * a) / (p + a * (1 - p)) +
  p * (1 - q + (b (p + b (p - q) - 2 p * q)) / (p + b (1 - p))) *
   (1 - (a * p - a^2 * (p - q)) / (p + a * (1 - p)))
```

$$\frac{1}{2}\left(\frac{1}{2}+\frac{1}{20}\left(-\frac{1}{2}+p\right)\right)+\frac{a\left(p-\left(\frac{1}{2}+\frac{1}{20}\left(-\frac{1}{2}+p\right)\right)p\right)}{a\,(1-p)+p}+p\left(1-\frac{a\,p-a^2\left(-\frac{1}{2}+\frac{1}{20}\left(\frac{1}{2}-p\right)+p\right)}{a\,(1-p)+p}\right)$$
$$\left(\frac{1}{2}+\frac{1}{20}\left(\frac{1}{2}-p\right)+\frac{b\left(p-2\left(\frac{1}{2}+\frac{1}{20}\left(-\frac{1}{2}+p\right)\right)p+b\left(-\frac{1}{2}+\frac{1}{20}\left(\frac{1}{2}-p\right)+p\right)\right)}{b\,(1-p)+p}\right)$$

```
Simplify[c2 - q]
```

$$\frac{1}{1600}\left(-380-40\,p+\frac{40\,a\,p\,(-21+2\,p)}{a\,(-1+p)-p}-\left(p\left(a\,(40-80\,p)+40\,p+19\,a^2\,(-1+2\,p)\right)\left(b^2\,(19-38\,p)+p\,(-21+2\,p)+b\left(-21+21\,p+2\,p^2\right)\right)\right)\Big/\left((a\,(-1+p)-p)\,(b\,(-1+p)-p)\right)\right)$$

```
Reduce[c2 - q ≤ 0 && a ≥ 0 && a ≤ 1 && b ≥ 0 && b ≤ 1 && p ≥ 1 / 2 && p ≤ 1, {a, b, p}, Reals]
```

$$a == 0\ \&\&\ 0 \le b \le 1\ \&\&\ p == \frac{1}{2}$$

Case 3

```
c3 = q / 2 + ((p - p * q) * a) / (p + a * (1 - p)) +
   p * (1 - (2 p - q) / (2 p)) * (1 - (a * p - a^2 * (p - q)) / (p + a * (1 - p)))
```

$$\frac{1}{2}\left(\frac{1}{2}+\frac{1}{20}\left(-\frac{1}{2}+p\right)\right)+\frac{a\left(p-\left(\frac{1}{2}+\frac{1}{20}\left(-\frac{1}{2}+p\right)\right)p\right)}{a\,(1-p)+p}+$$
$$p\left(1-\frac{-\frac{1}{2}+\frac{1}{20}\left(\frac{1}{2}-p\right)+2\,p}{2\,p}\right)\left(1-\frac{a\,p-a^2\left(-\frac{1}{2}+\frac{1}{20}\left(\frac{1}{2}-p\right)+p\right)}{a\,(1-p)+p}\right)$$

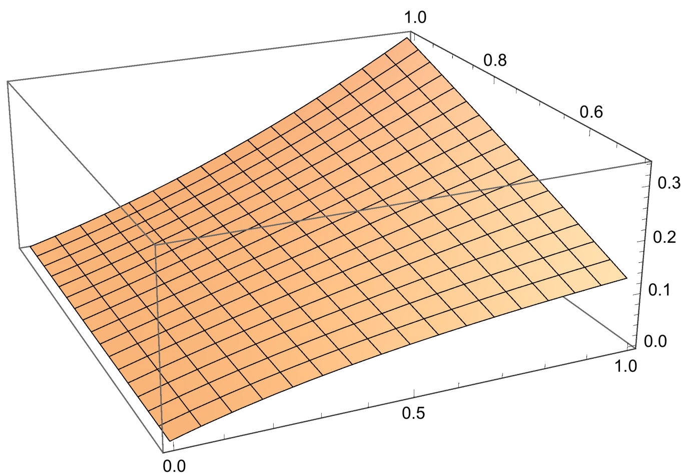
```
Simplify[c3 - q]
```

$$\frac{a\left(40\,p\,(-23+6\,p)-19\,a\left(-19+36\,p+4\,p^2\right)\right)}{3200\,(a\,(-1+p)-p)}$$

```
Reduce[c3 - q ≤ 0 && a ≥ 0 && a ≤ 1 && p ≥ 1 / 2 && p ≤ 1, {a, p}, Reals]
```

$$a == 0\ \&\&\ \frac{1}{2}\le p\le 1$$

```
Plot3D[c3 - q, {a, 0, 1}, {p, 0.5, 1}]
```



Case 4

```
c4 = (p * q - b (p - q)) / (p + b (1 - p)) + ((p - p * q) * a) / (p + a * (1 - p)) +
   p * (1 - (p - b (p - q)) / (p + b (1 - p))) * (1 - (a * p - a^2 * (p - q)) / (p + a * (1 - p)))
```

$$\frac{a\left(p-\left(\frac{1}{2}+\frac{1}{20}\left(-\frac{1}{2}+p\right)\right)p\right)}{a\,(1-p)+p}+\frac{\left(\frac{1}{2}+\frac{1}{20}\left(-\frac{1}{2}+p\right)\right)p-b\left(-\frac{1}{2}+\frac{1}{20}\left(\frac{1}{2}-p\right)+p\right)}{b\,(1-p)+p}+$$
$$p\left(1-\frac{a\,p-a^2\left(-\frac{1}{2}+\frac{1}{20}\left(\frac{1}{2}-p\right)+p\right)}{a\,(1-p)+p}\right)\left(1-\frac{p-b\left(-\frac{1}{2}+\frac{1}{20}\left(\frac{1}{2}-p\right)+p\right)}{b\,(1-p)+p}\right)$$

```
Simplify[c4 - q]
```

$$-\frac{a\,p\,(-21+2\,p)\,(40\,p+(-40+19\,a)\,b\,(-1+2\,p))}{1600\,(a\,(-1+p)-p)\,(b\,(-1+p)-p)}$$

```
Reduce[c4 - q ≤ 0 && a ≥ 0 && a ≤ 1 && p ≥ 1 / 2 && p ≤ 1 && b ≥ 0 && b ≤ 1, {a, b, p}, Reals]
```

$$a == 0\ \&\&\ 0 \le b \le 1\ \&\&\ \frac{1}{2} \le p \le 1$$

Case 5

```
c5 = 2 * ((p - p * q) * a) / (p + a * (1 - p)) +
  p * (1 - (p - a (p - q)) / (p + a (1 - p))) + p * (p - a (p - q)) / (p + a (1 - p)) * Max[0, 1 - 2 a]
```

$$\frac{2\,a\left(p-\left(\frac{1}{2}+\frac{1}{20}\left(-\frac{1}{2}+p\right)\right)p\right)}{a\,(1-p)+p}+p\left(1-\frac{p-a\left(-\frac{1}{2}+\frac{1}{20}\left(\frac{1}{2}-p\right)+p\right)}{a\,(1-p)+p}\right)+\frac{p\left(p-a\left(-\frac{1}{2}+\frac{1}{20}\left(\frac{1}{2}-p\right)+p\right)\right)\mathrm{Max}[0,\,1-2\,a]}{a\,(1-p)+p}$$

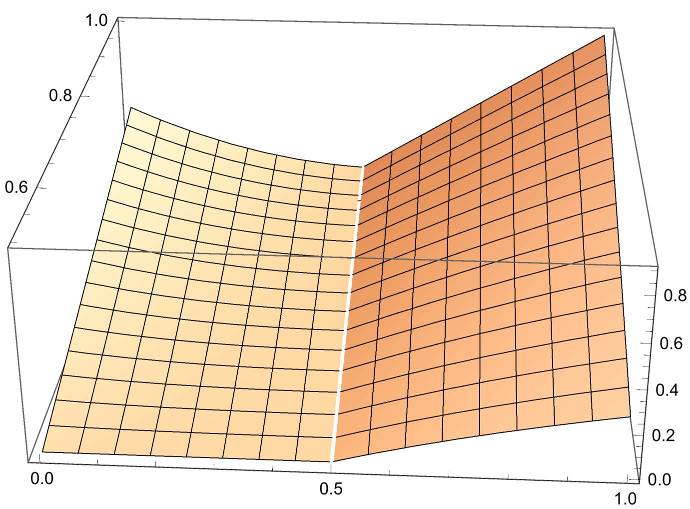
```
Simplify[c5 - q]
```

$$\left(p\,(19+2\,p)+a\left(19-80\,p+4\,p^2\right)+p\,(-40\,p+19\,a\,(-1+2\,p))\,\mathrm{Max}[0,\,1-2\,a]\right)\Big/\left(40\,(a\,(-1+p)-p)\right)$$

```
Reduce[c5 - q ≤ 0 && a ≥ 0 && a ≤ 1 && p ≥ 1 / 2 && p ≤ 1, {a, p}, Reals]
```

$$0 \le a \le \frac{1}{2}\ \&\&\ p == \frac{1}{2}$$

```
Plot3D[c5 - q, {a, 0, 1}, {p, 0.5, 1}]
```



Case 6

```
c6 = q / 2 + ((p - p * q) * a) / (p + a * (1 - p)) +
   p * (1 - (a * p - a^2 * (p - q)) / (p + a * (1 - p))) *
    (1 - (a * p - a^2 * (p - q)) / (p + a * (1 - 2 * p) + a^2 (p - q))) *
    (1 - (p * q + a (q - 2 p + p * q)) / (p + a (1 - 2 p) + a^2 (p - q)))
```

$$\frac{1}{2}\left(\frac{1}{2}+\frac{1}{20}\left(-\frac{1}{2}+p\right)\right)+\frac{a\left(p-\left(\frac{1}{2}+\frac{1}{20}\left(-\frac{1}{2}+p\right)\right)p\right)}{a\,(1-p)+p}+$$
$$p\left(1-\frac{a\,p-a^2\left(-\frac{1}{2}+\frac{1}{20}\left(\frac{1}{2}-p\right)+p\right)}{a\,(1-p)+p}\right)\left(1-\frac{a\,p-a^2\left(-\frac{1}{2}+\frac{1}{20}\left(\frac{1}{2}-p\right)+p\right)}{a\,(1-2\,p)+p+a^2\left(-\frac{1}{2}+\frac{1}{20}\left(\frac{1}{2}-p\right)+p\right)}\right)$$
$$\left(1-\left(\left(\frac{1}{2}+\frac{1}{20}\left(-\frac{1}{2}+p\right)\right)p+a\left(\frac{1}{2}+\frac{1}{20}\left(-\frac{1}{2}+p\right)-2\,p+\left(\frac{1}{2}+\frac{1}{20}\left(-\frac{1}{2}+p\right)\right)p\right)\right)\Big/\left(a\,(1-2\,p)+p+a^2\left(-\frac{1}{2}+\frac{1}{20}\left(\frac{1}{2}-p\right)+p\right)\right)\right)$$

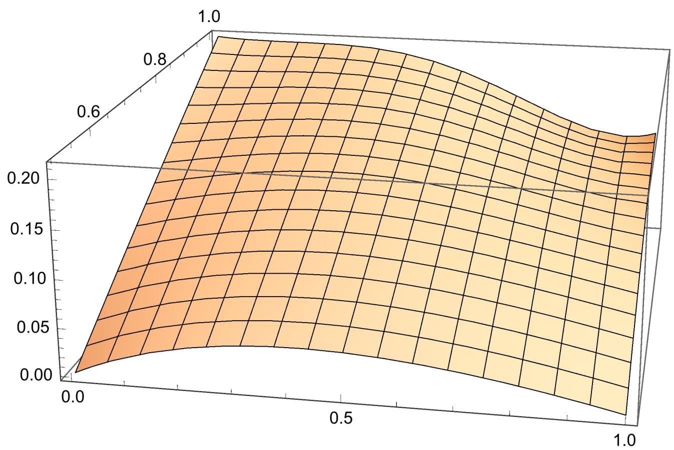
```
Simplify[c6 - q]
```

$$\Big(-1444\,a^4\,(1-2\,p)^2\,p+40\,p^2\left(19-40\,p+4\,p^2\right)-40\,a\,p\left(-38+179\,p-170\,p^2+8\,p^3\right)+a^2\left(760-5921\,p+15\,320\,p^2-11\,348\,p^3-176\,p^4\right)+19\,a^3\left(-19+261\,p-772\,p^2+644\,p^3+16\,p^4\right)\Big)\Big/\left(80\,(a\,(-1+p)-p)\left(a\,(40-80\,p)+40\,p+19\,a^2\,(-1+2\,p)\right)\right)$$

```
Reduce[c6 - q ≤ 0 && a ≥ 0 && a ≤ 1 && p ≥ 1 / 2 && p ≤ 1, {a, p}, Reals]
```

$$\left(a == 0\ \&\&\ p == \frac{1}{2}\right)\ ||\ \left(a == 1\ \&\&\ p == \frac{1}{2}\right)$$

```
Plot3D[c6 - q, {a, 0, 1}, {p, 0.5, 1}]
```



Case 7

```
c7 = q / 2 + ((p - p * q) * a) / (p + a * (1 - p)) +
  p * (1 - (a * p - a^2 * (p - q)) / (p + a * (1 - p))) *
   (1 - (a * p - a^2 * (p - q)) / (p + a * (1 - 2 * p) + a^2 (p - q))) *
   (1 - (p * q + a (q - p - p * q) + a^2 (p - q)) / (p + a (1 - 2 p) + a^2 (p - q)))
```

$$\frac{1}{2}\left(\frac{1}{2}+\frac{1}{20}\left(-\frac{1}{2}+p\right)\right)+\frac{a\left(p-\left(\frac{1}{2}+\frac{1}{20}\left(-\frac{1}{2}+p\right)\right)p\right)}{a\,(1-p)+p}+ p\left(1-\frac{a\,p-a^2\left(-\frac{1}{2}+\frac{1}{20}\left(\frac{1}{2}-p\right)+p\right)}{a\,(1-p)+p}\right)\left(1-\frac{a\,p-a^2\left(-\frac{1}{2}+\frac{1}{20}\left(\frac{1}{2}-p\right)+p\right)}{a\,(1-2\,p)+p+a^2\left(-\frac{1}{2}+\frac{1}{20}\left(\frac{1}{2}-p\right)+p\right)}\right) \left(1-\left(\left(\frac{1}{2}+\frac{1}{20}\left(-\frac{1}{2}+p\right)\right)p+a^2\left(-\frac{1}{2}+\frac{1}{20}\left(\frac{1}{2}-p\right)+p\right)+a\left(\frac{1}{2}+\frac{1}{20}\left(-\frac{1}{2}+p\right)-p-\left(\frac{1}{2}+\frac{1}{20}\left(-\frac{1}{2}+p\right)\right)p\right)\right)\Big/\left(a\,(1-2\,p)+p+a^2\left(-\frac{1}{2}+\frac{1}{20}\left(\frac{1}{2}-p\right)+p\right)\right)\right)$$

```
Simplify[c7 - q]
```

$$\left(40\,p^2\left(19-40\,p+4\,p^2\right)-40\,a\,p\left(-38+179\,p-174\,p^2+16\,p^3\right)-19\,a^3\left(19-181\,p+380\,p^2-196\,p^3+16\,p^4\right)+a^2\left(760-5921\,p+13\,960\,p^2-9108\,p^3+784\,p^4\right)\right)\Big/\left(80\,(a\,(-1+p)-p)\,\left(a\,(40-80\,p)+40\,p+19\,a^2\,(-1+2\,p)\right)\right)$$

```
Reduce[c7 - q ≤ 0 && a ≥ 0 && a ≤ 1 && p ≥ 1 / 2 && p ≤ 1, {a, p}, Reals]
```

$$\left(a == 0\ \&\&\ p == \frac{1}{2}\right)\ ||\ \left(a == 1\ \&\&\ p == \frac{1}{2}\right)$$

```
Plot3D[c7 - q, {a, 0, 1}, {p, 0.5, 1}]
```