Query Efficient Weighted Stochastic Matching

Mahsa Derakhshan
Northeastern University, Boston, MA, USA

Mohammad Saneian
Northeastern University, Boston, MA, USA

Abstract

In this paper, we study the *weighted stochastic matching* problem. Let $G = (V, E)$ be a given edge-weighted graph, and let its realization $\mathcal{G}$ be a random subgraph of G that includes each edge $e \in E$ independently with a known probability p_e. The goal in this problem is to pick a sparse subgraph Q of G without prior knowledge of $\mathcal{G}$, such that the maximum weight matching among the realized edges of Q (i.e., the subgraph $Q \cap \mathcal{G}$) in expectation approximates the maximum weight matching of the entire realization $\mathcal{G}$.

It is established by previous work that attaining any constant approximation ratio for this problem requires selecting a subgraph of max-degree $\Omega(1/p)$, where $p = \min_{e \in E} p_e$. On the positive side, there exists a $(1 - \varepsilon)$-approximation algorithm by Behnezhad and Derakhshan [FOCS'20], albeit at the cost of a max-degree having exponential dependence on $1/p$. Within the $O(1/p)$ query regime, however, the best-known algorithm achieves a 0.536 approximation ratio due to Dughmi, Kalayci, and Patel [ICALP'23], improving over the 0.501 approximation algorithm by Behnezhad, Farhadi, Hajiaghayi, and Reyhani [SODA'19].

In this work, we present a 0.68-approximation algorithm with the asymptotically optimal $O(1/p)$ queries per vertex. Our result not only substantially improves the approximation ratio for weighted graphs, but also breaks the well-known 2/3 barrier with the optimal number of queries – even for unweighted graphs. Our analysis involves reducing the problem to designing a randomized matching algorithm on a given stochastic graph with some *variance-bounding* properties. To achieve these properties, we leverage a randomized algorithm by MacRury and Ma [STOC'24] for a variant of online stochastic matching.

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases Sublinear algorithms, Stochastic, Matching

Digital Object Identifier 10.4230/LIPIcs.ICALP.2025.67

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: arXiv:2311.08513

1 Introduction

In the *stochastic weighted matching* problem, we are given an n-vertex weighted graph $G = (V, E)$ along with a parameter $p_e \in (0, 1]$ for any edge $e \in E$. A random subgraph $\mathcal{G}$ of G is then generated by independently including (or realizing) each edge $e \in E$ with probability p_e. Here, we refer to G as the base graph and $\mathcal{G}$ as the realized subgraph. The objective of this problem is to select a subgraph Q of the base graph without the knowledge of its realization such that: (1) Q has a small max-degree, namely a constant with respect to n, and (2) The realized edges of Q (i.e., the graph $Q \cap \mathcal{G}$) contain a large weight approximate matching. We define the approximation ratio as the expected weight of the maximum matching among the realized edges of Q over the expected weight of the maximum weighted matching of $\mathcal{G}$.

One immediate application of the stochastic weighted matching problem is its use as a *matching sparsifier*, which approximates the maximum weighted matching even when random edge failures occur [1]. Additionally, it finds various applications in matching markets, including kidney exchange [11], online labor markets [9, 10], and dating platforms. In these applications, we are provided with the base graph G, but we are tasked with finding a matching in the realized subgraph $\mathcal{G}$. To achieve this, an algorithm can *query* each edge of G to determine whether it is realized. However, these queries often involve time-consuming or costly operations, such as conducting candidate interviews or medical exams. Hence, it is crucial to minimize the number of queries. This can be accomplished by non-adaptively querying a subgraph Q with a small degree while still expecting to find a matching with a large approximation ratio among its realized edges.

A simple lower bound – see e.g. [2] – -shows that attaining any constant approximation ratio for this problem requires selecting a subgraph of max-degree $\Omega(1/p)$ where $p = \min_{e \in E} p_e$. This simply follows from the fact that if the graph G is a clique, one needs $\Omega(1/p)$ queries per vertex to avoid too many singleton vertices. This raises a natural question:

> *What is the best approximation achievable with a subgraph of optimal max-degree $O(1/p)$?*

Prior to this work, the best known approximation with these many queries was obtained in the work of Dughmi, Kalayci, and Patel [12] who obtained a 0.536 approximation, improving over the prior 0.501 approximation of Behnezhad, Farhadi, Hajiaghayi, and Reyhani [9].[1] We note that if is also extensive work on achieving higher approximations by increasing the maximum degree e.g. to $\exp(1/p)$ [6]. We will overview these results later in Subsection 1.1.

Our main result is a significant improvement of the approximation from 0.536 to 0.68. Our approximation improves to 0.73 in the case of bipartite graphs.

▶ **Theorem 1.** *For the weighted stochastic matching problem, there exists an algorithm that picks a $O1/p$ degree subgraph Q of G such that the expected weight of the max-weight realized matching in Q is at least* 0.68 *times the expected weight of the max-weight realized matching in G. The approximation improves to* 0.73 *for bipartite graphs.*

It is also worth noting that our Theorem 1 breaks through the intriguing 2/3-approximation barrier (see [2, 8]) for this problem. Before this work, it was not known whether this bound could be broken by querying a graph of max degree $O(1/p)$ *even in the case of unweighted graphs*. To achieve this result, we demonstrate that the problem can be reduced to designing approximate matching algorithms with specific properties, which we term *variance-bounding* matching algorithms. This reduction implies that further exploration of the stochastic matching problem may be focused on developing such algorithms.

1.1 Related Work

Since its introduction in the pioneering work of Blum, Dickerson, Haghtalab, Procaccia, Sandholm, and Sharma [11], the stochastic matching problem has received considerable attention [11, 2, 3, 18, 10, 9, 1, 7, 8, 6]. It is established by Assadi, Khanna, and Li [2] that attaining any constant approximation ratio for this problem requires selecting a subgraph of maximum degree $\Omega(1/p)$, where $p = \min_{e \in E} p_e$.

[1] We note that the result of [9] requires a maximum degree of $O(\log(1/p)/p)$.

Yamaguchi and Maehara [18] were the first to consider the weighted version of the problem. They provided a $0.5 - \varepsilon$ approximation algorithm via $O(W \log n/p)$ queries per vertex, where W is the largest edge weight. Their query complexity was later improved to $\text{poly}(1/p)$ by Behnezhad and Reyhani [10]. Subsequently, Behnezhad, Farhadi, Hajiaghayi, and Reyhani [9] provided the first algorithm breaking the 0.5 approximation via $O(1/p)$ queries per vertex. Finally, Dughmi, Kalayci, and Patel [12] improved this approximation ratio to 0.536 using an asymptotically optimal number of queries. In a separate line of work, Behnezhad and Derakhshan [6] showed that if $\exp(1/p)$ queries per vertex are allowed, it is possible to improve this approximation ratio up to $1 - \varepsilon$.

For the unweighted version of the problem, the pioneering work of Blum et al. [11] provides a 0.5 approximation via $\text{poly}(1/p)$ queries per vertex. After a series of works [9, 3], this approximation ratio was improved to 2/3 via $O(\log(1/p)/p)$ queries by Assadi and Bernstein [1]. For the special case of unweighted bipartite graphs, Behnezhad, Blum, and Derakhshan improved this approximation ratio to 0.73 [5]. Behnezhad, Derakhshan, and Hajiaghayi [8] were the first to obtain a $(1 - \varepsilon)$-approximation, albeit using $\exp(1/p)$ queries per vertex. Finally, in a recent breakthrough result, Azarmehr, Behnezhad, Ghafari, and Rubinfeld [4] improved this query complexity to $(1/p)^{\exp(1/\varepsilon)}$.

2 Technical Overview

The algorithm we use to construct Q is a quite simple one which was introduced by [9] and subsequently studied by [8, 5]. Given a parameter t, the algorithm starts by drawing t realizations of G drawn from the same distribution as $\mathcal{G}$. Let us represent these random subgraphs by $\mathcal{G}_1, \ldots, \mathcal{G}_t$. We then let Q be the union of max-weight matchings of these graphs. That is

$$Q := \cup_{i \in t} \mathsf{MWM}(\mathcal{G}_i),$$

where $\mathsf{MWM}(\cdot)$ is a deterministic algorithm returning the max-weight matching of a given graph. Since Q is a union of t matchings, it clearly has max-degree t. The challenge, however, is proving that for a t as small as $O1/p$, the realization of Q contains a large weight matching. We provide a constructive proof for this. That is, we design an algorithm for finding a matching with a large approximation ratio in $\mathcal{Q}$, the actual realization of Q. Below, we first briefly review the ideas used by the previous work and then discuss the ingredients we add to achieve our desired result.

2.1 Crucial/Non-crucial Edge Decomposition

The framework utilized to analyze the aforementioned algorithm involves partitioning the edges into two categories: *crucial* and *non-crucial*. Separate arguments are then presented to demonstrate how these edges can be integrated to construct a large weight enough matching in $\mathcal{Q}$. Let x_e denote the probability of edge e being part of the optimal solution, i.e.,

$$x_e = \Pr[e \in \mathsf{MWM}(\mathcal{G})].$$

We define the set of crucial edges, denoted by C, and the set of non-crucial edges, denoted by N, as follows:

$$C = \{e \in E : x_e \geq \tau\} \qquad \text{and} \qquad N = \{e \in E : x_e < \tau\},$$

where τ is a fixed threshold in the order of p. Note that by choosing a sufficiently large value for $t = O1/p$, we can ensure that Q contains nearly all of the crucial edges. To establish

the existence of a large weight matching in $\mathcal{G}$, the first step is to construct a matching M_c exclusively on the crucial edges which is an α-approximation with respect to the contribution of the crucial edges to the optimal solution. (M_c should satisfy some other useful properties which we will discuss later.) The next step is constructing a fractional matching $\boldsymbol{f}$ on the subgraph of non-crucial edges whose endpoints are unmatched in M_c. This fractional matching should satisfy the two following properties: first, for any edge $e \in N$, it holds that $\mathbf{E}[f_e] \simeq x_e$; second, the values of f_e should be small enough to ensure that $\boldsymbol{f}$ has almost no integrality gap. By combining these steps, the framework constructs a matching with weight almost $\alpha \times \mathsf{W}(\mathsf{MWM}(\mathcal{G}) \cap C) + \mathsf{W}(\mathsf{MWM}(\mathcal{G}) \cap N)$. Here $\mathsf{W}(.)$ is a function returning the weight of a given matching.

All papers utilizing this analysis framework require the algorithm used for constructing M_c to match the endpoints of any non-crucial edge e independently. Otherwise, the edge is discarded. This requirement is the main reason why Behnezhad and Derakhshan [6] need to take an exponential number of edges per vertex. To achieve this property, they employ a distributed LOCAL algorithm for constructing M_c, which can lead to a vertex being dependent on the vertices within its $\Omega(\log(\Delta))$ radius ball, where Δ denotes the crucial degree of a vertex. Since potentially $(1/p)^{\log(1/p)}$ non-crucial edges may be discarded for each vertex, these edges need to have small x_e values. Consequently, a small threshold τ and a large t must be chosen. Due to known lower bounds for matching in the LOCAL model [15], one cannot hope to prove desirable approximation ratios for a Q of max-degree $\text{poly}(1/p)$ following this approach.

In this work, we demonstrate that it is possible to relax the requirement regarding the independent matching of endpoints of any non-crucial edge in M_c. Instead, we replace it with an upper bound on the variance of a parameter related to the neighborhood of each vertex. Specifically, it should be possible to pick a subset A of the vertices unmatched by M_c such that:

1. Any non-crucial edge $e = (u, v)$ satisfies $\Pr[\{u, v\} \subset A] \geq \delta$ for a fixed constant $\delta > 0$.
2. Let us define random variable Z_v related to the neighborhood of any vertex v as

$$Z_v = \sum_{e=(u,v)\in N} \frac{x_e}{\Pr[\{u, v\} \subset A]} \mathbb{1}_{u\in A}. \tag{1}$$

   Note that the randomization here is due to A and M_c being random variables themselves. We require the variance of this random variable to be upper-bounded as follows: $\text{Var}(Z_v) \leq \frac{10\tau}{\delta^2}$.

We define an algorithm for finding M_c and A to be a *variance-bounding* matching algorithm (see definition 10) if it satisfies the above-mentioned property (and a few others). We then provide a reduction demonstrating that if M_c is an α-approximation with respect to the contribution of crucial edges to the optimal solution, then it is possible to find an almost $\frac{1}{2-\alpha}$-approximate solution on the realized edges of Q. Our proof strongly relies on independent edge realizations, hence enabling us to break the 2/3 barrier.

The second step of our analysis involves proving the existence of variance-bounding matching algorithms with approximation ratios of almost 0.535 and $1 - 1/e$, respectively, for general graphs and bipartite graphs. In order to do this, we utilize algorithms designed for a variant of online stochastic matchings, particularly batched random-order contention resolution schemes (RCRS). We prove that any α-selectable RCRS can be used to obtain a variance-bounding matching algorithm with an approximation ratio of almost α. We discuss this in Section 6.

3 Preliminaries

3.1 Notation

In the stochastic weighted matching problem, the input is an n-vertex graph $G = (V, E)$, a vector of weights $\boldsymbol{w} = \{w_e : e \in E\}$ and a probability vector $\boldsymbol{p} = \{p_e : p_e \in E\}$. Subgraph $\mathcal{G}$ is a random subgraph of G which contains each edge independently with probability p_e. The goal in this problem is to pick a subgraph Q of G without the knowledge of its realization such that: (1) Q has a small max-degree, namely a constant with respect to n, and (2) The realized edges of Q (i.e., the graph $Q \cap \mathcal{G}$) contain a large weight approximate matching. We define the approximation ratio as

$$\frac{\mathbf{E}\left[\mathsf{W}\left(\mathsf{MWM}(\mathcal{Q})\right)\right]}{\mathbf{E}[\mathsf{W}(\mathsf{MWM}(\mathcal{G}))]},$$

where $\mathcal{Q} = \mathcal{G} \cap Q$ is the realization of Q and $\mathsf{MWM}(.)$ is a deterministic algorithm returning a maximum weighted matching of a given graph, and $\mathsf{W}(M) = \sum_{e \in M} w_e$ is a function returning the weight of a given matching M. We will use $\textsc{opt} = \mathsf{MWM}(\mathcal{G})$ to refer to the maximum matching of the actual realization. We may sometimes abuse notation and use $\textsc{opt}$ to refer to its expected weight when it is clear from the context. Note that while $\textsc{opt}$ is a random variable $\mathbf{E}[\mathsf{W}(\textsc{opt})]$ is just a number. For any edge $e \in E$, we define

$$x_e = \Pr[e \in \textsc{opt}],$$

where the probability is taken over the randomization in $\mathcal{G}$. Similarly, for any vertex $v \in V$ we let $x_v = \Pr[v \in \textsc{opt}]$ be the probability that v is matched in $\textsc{opt}$. By the definition stated below, $\boldsymbol{x}$ is a fractional matching as each vertex joins $\textsc{opt}$ w.p. at most one.

▶ **Definition 2** (Fractional matching). *A* fractional matching $\boldsymbol{x}$ *of a graph* $G = (V, E)$ *is an assignment* $\{x_e\}_{e \in E}$ *to the edges, where* $x_e \in [0, 1]$ *for each edge* $e \in E$*, and for each vertex* $v \in V$*, we have* $x_v := \sum_{e \ni v} x_e \leq 1$*. We use* $|\boldsymbol{x}| := \sum_{e \in E} x_e$ *to denote the size of a fractional matching, and for any subset* $E' \subseteq E$*, we use* $\boldsymbol{x}(E') := \sum_{e \in E'} x_e$*.*

▶ **Definition 3** (Graph hallucination). *We say graph* $\mathcal{H}$ *is a hallucination of graph* H *which is a subgraph of* G *if any edge* $e = (u, v)$ *in* H *is present in* $\mathcal{H}$ *with probability* p_e*.*

Throughout the paper, we use the notation $O_\varepsilon(f(n))$ which means we have assumed ε to be a constant to calculate the complexity of $f(n)$. The max-degree of subgraph Q we find in this paper depends on the smallest p_e amongst all edges, which we refer to as p. In other words

$$p = \min_{e \in E} p_e.$$

In the following table, we list a set of variables and their values, which we will use throughout the paper. Values are defined as functions of $\varepsilon \in (0, 1)$, which is a sufficiently small constant, and $\delta \in (0, 1)$, which we will introduce in Definition 10.

Table 1 Value of the parameters used throughout the paper.

Variable	δ	τ	η	β	γ	c
Value	$\varepsilon^{0.5}$	$20p\varepsilon^5\delta^2$	$\varepsilon/10$	$\varepsilon^2/100$	$\frac{1-\varepsilon^2}{1+3\eta}$	$10/\varepsilon$

3.2 Concentration Inequalities and Probabilistic Tools

In this section, we state the concentration inequalities and some of the probabilistic tools that will be used throughout the paper.

▶ **Proposition 4** (The Efron–Stein Inequality [17]). *Suppose $X_1, ..., X_n, X'_1, ..., X'_n$ are independent random variables with X'_i and X_i having the same distribution for all i. Let $X = (X_1, ..., X_n)$ and $X^{(i)} = (X_1, ..., X_{i-1}, X'_i, ..., X_{i+1}, ..., X_n)$. Then:*

$$\mathrm{Var}(f(X)) \leq \frac{1}{2}\sum_{i=1}^{n} \mathbf{E}\left[\left(f(X) - f(X^{(i)})\right)^2\right]$$

▶ **Proposition 5** (Chebyshev's Inequality). *Let X be a random variable with finite non-zero standard deviation s, (and thus finite expected value μ.) Then for any real number $c > 0$, we have*

$$\Pr[|X - \mu| \geq cs] \leq \frac{1}{c^2}.$$

▶ **Proposition 6** (Law of Total Variance). *Let X be a random variable and Y be a random variable with respect to the same sample space. Then, the variance of X can be expressed as*

$$\mathrm{Var}(X) = \mathbf{E}[\mathrm{Var}(X \mid Y)] + \mathrm{Var}(\mathbf{E}[X \mid Y]).$$

▶ **Definition 7** (Negative Association). *A set of random variables $X_1, ..., X_n$ is said to be negatively associated if for any two disjoint index sets $i, j \subseteq [n]$ and two functions f, g both monotone increasing or both monotone decreasing it holds:*

$$\mathbf{E}[f(X_i : i \in I) \cdot g(X_j : j \in J)] \leq \mathbf{E}[f(X_i : i \in I)] \cdot \mathbf{E}[g(X_j : j \in J)]$$

4 The Algorithm for Selecting Q

In this section, we present a formal statement of the algorithm employed to construct the subgraph Q. We then explain how we can use the tools we provide later in the paper to show quering Q proves Theorem 1 (the main theorem).

In summary, for a given parameter $t = O_\varepsilon(1/p)$, we draw t matchings from the same distribution as OPT (the optimal solution) and define Q as the union of these matchings.

Algorithm 1 Algorithm for constructing Q.

```
Q ← ∅
for i from 1 to t do
    Let G_i be a random realization of G containing each edge independently w.p. p_e.
    Set M_i = μ(G_i)
    Q ← Q ∪ M_i
end for
return Q
```

Let us define subsets of crucial and noncrucial edges as follows.

$$C = \{e \in E : x_e \geq \tau\} \qquad \text{and} \qquad N = \{e \in E : x_e < \tau\}, \tag{2}$$

where $\tau = \theta(\frac{1}{t})$ and $t = \frac{1}{\tau\varepsilon}$ for a sufficiently small $\varepsilon \in (0, 1)$. (The actual value of τ and the other variables used in the paper are presented in Table 1.) Note that in the above algorithm,

matching $M_1, \ldots, M_t$ are independent from each other and come from the same distribution as OPT. This means that for any edge e and $i \in [t]$ we have $\Pr[e \in M_i] = Pr[e \in \text{OPT}] = x_e$. As a result, the subgraph outputted by this algorithm contains almost all the crucial edges. Moreover, it picks any non-crucial edge $e \in N$ with a large enough probability as a function of their x_e. We formally state these properties below in Claim 8 and Claim 9. While the proofs are pretty straightforward, we include them in the full version for the sake of completeness.

▷ Claim 8. Given constant numbers $\tau, \varepsilon \in (0,1)$, Let Q be the output of Algorithm 1 with parameter $t \geq \frac{1}{\tau\varepsilon}$. Any crucial edge $e \in C$ with $x_e \geq \tau$ is present in Q with probability at least $1 - \varepsilon$.

▷ Claim 9. Any edge $e \in E$ is present in Q with probability at least $\min(1/3, tx_e/3)$.

4.1 Proof of the Main Theorem

As discussed previously in Section 2, to prove Theorem 1, we will show that $\mathcal{Q}$ contains a 0.68-approximate matching for general graphs and a 0.73-approximate matching for bipartite graphs. Since $\mathcal{Q}$ is the union of $t = O_\varepsilon(1/p)$ matchings, this proves our main result. In Definition 10, we define *variance-bounding* matching algorithms. In Lemma 11, we prove that for any $\alpha \in (0,1)$ and a small enough constant $\varepsilon > 0$ existence of an α-approximate variance-bounding algorithm implies $\mathcal{Q}$ contains a $(\frac{1}{2-\alpha} - \varepsilon)$-approximate matching. In Lemma 21, we prove the existence of variance-bounding matching algorithms with approximation ratios of $(0.535 - 6\sqrt{\delta})$ and $(1 - 1/e - 6\sqrt{\delta})$, respectively, for general and bipartite graphs where $\delta \in (0,1)$ is a parameter in Definition 10. In Table 1, we choose $\delta = \varepsilon^{0.5}$. By picking a sufficiently small enough ε, this implies that $\mathcal{Q}$ contains a matching with an approximation ratio of

$$\frac{1}{2 - 0.535 + 6\varepsilon^{0.25}} - \varepsilon \geq 0.68$$

for general graphs and approximation ratio of

$$\frac{1}{1 + 1/e + 6\varepsilon^{0.25}} - \varepsilon \geq 0.73$$

for bipartite graphs.

5 The Reduction

In this section, we first introduce *variance-bounding* matching algorithms and then show that the the existence of an α-approximation variance-bounding matching algorithm implies that it is possible to find a $(\frac{1}{2-\alpha} - \varepsilon)$-approximate matching with $O_\varepsilon(1/p)$ queries per vertex.

▶ **Definition 10** (Variance-bounding (VB) matching algorithm). *We call a matching algorithm $\mathcal{VB}$ an α-approximation variance-bounding algorithm if it has the following properties. It takes as input (1) a graph $H = (V, E)$ whose edges are realized independently, each with a given probability p_e forming subgraph $\mathcal{H}$, and (2) a matching $M_{\mathcal{O}}$ of $\mathcal{H}$ found by an arbitrary (potentially randomized) algorithm. The algorithm then outputs a matching M_c of $\mathcal{H}$ and a subset A of vertices that are unmatched in M_c [2] such that:*

(i) *M_c is in expectation an α-approximate matching with respect to $M_{\mathcal{O}}$.*

[2] A is just a subset of unmatched vertices, so some unmatched vertices may not be in A.

(ii) *For any vertex* $v \in V$, $\Pr[v \in A] \geq \Pr[v \notin M_{\mathcal{O}}]$.

(iii) *For any two vertices* u, v *that do not have an edge in* H *the following holds.* $\Pr[\{u, v\} \subset A] \geq \delta$ *for a* $\delta \in (0, 1)$.

(iv) *Given a parameter* $\tau \in (0, 1)$*, let* $\boldsymbol{x}$ *be a fractional matching on* $\overline{H} = (V, \overline{E})$ *(the complement of* H*) with* $x_e \leq \tau$ *for any* $e \in \overline{E}$*. For any vertex* v *variable* Z_v*, defined below, satisfies* $\text{Var}(Z_v) \leq \frac{6\tau}{\delta^2}$.

Let us briefly explain why we need a variance-bounding matching algorithm. We will use this algorithm on all the realized crucial edges (i.e., $H = (V, C)$) and define $M_{\mathcal{O}}$ to be a matching with the expected weight the same as the contribution of the crucial edges to the optimal solution. We formally define these inputs in Definition 12. This gives us a matching M_c on the crucial edges and a subset A of vertices unmatched in M_c. We will then construct a fractional matching $\boldsymbol{f}$ with a small integrity gap exclusively using the (queried and realized) non-crucial edges between vertices in A. We need Property (i) to ensure that M_c is large with respect to the contribution of crucial edges to the optimal solution. Property (ii) ensures that each vertex is available in A with a large enough probability for its non-crucial edges to be able to contribute to $\boldsymbol{f}$ almost as much as their contribution to OPT. We need Property (iii) to ensure that each edge is available for potential contribution to $\boldsymbol{f}$ with a large enough probability. Finally, we will use Property (iv) to prove that constructing $\boldsymbol{f}$ in a particular way does not result in fractional degrees of vertices exceeding one too often. For more details about the importance of this property, see Section 5.2.

▶ **Lemma 11** (The Reduction). *For constant numbers* $\alpha \in (0.5, 1)$ *and* $\varepsilon \in (0, 0.1)$*, existence of an* α*-approximation variance-bounding algorithm* $\mathcal{VB}$ *(from Definition 10) implies a* $(\frac{1}{2-\alpha} - \varepsilon)$ *approximation algorithm for the weighted stochastic matching problem with* $O_\varepsilon(1/p)$ *queries per vertex.*

We will prove that the existence of an α-approximation variance-bounding algorithm implies that querying the subgraph Q which is the output of Algorithm 1 with parameter $t = O_\varepsilon(1/p)$ gives us a $(\frac{1}{2-\alpha} - \varepsilon)$-approximate solution. Before formally proving this in Section 5.3, we need to prove a series of other claims and provide some definitions. Below, we give a brief overview of the proof.

The first step of the proof is using the variance-bounding algorithm on the subgraph of all the crucial edges. Recall that $\mathcal{VB}$ takes as input (1) a graph H whose edges are realized independently, each with a given probability p_e forming subgraph $\mathcal{H}$, and (2) a matching $M_{\mathcal{O}}$ of $\mathcal{H}$ found by an arbitrary (potentially randomized) algorithm. Below, we detail the values assign to these parameters in our reduction.

▶ **Definition 12** (H and $M_{\mathcal{O}}$). *In our reduction we choose the following values for* $\mathcal{H}$ *and* $M_{\mathcal{O}}$*:*

1. *We set* H *to be the subgraph of all the crucial edges* C*. In other words,* $H = (V, C)$.
2. *We set* $M_{\mathcal{O}} = \mathsf{MWM}(\mathcal{H} \cup \mathcal{N}^\star) \cap \mathcal{H}$*, where* $\mathcal{H} = \mathcal{G} \cap H$ *is the actual realization of all the crucial edges, and* $\mathcal{N}^\star$ *is a random hallucination (refer to Definition 3) of the non-crucial edges containing each edge independently with probability* p_e*. Note that* $M_{\mathcal{O}}$ *is a matching only on the realized crucial edges.*

In the remainder of this paper, when referring to a variance-bounding algorithm without specifying the input, we assume that the variables H and $M_{\mathcal{O}}$ are defined according to Definition 12. Executing the variance-bounding algorithm $\mathcal{VB}$ with these predefined inputs gives us a matching M_c on the critical edges and a subset of unmatched vertices denoted as

A. Using Property (i), we prove (in Claim 13) that M_c is an α approximation with respect to the contribution of the crucial edges to the optimal solution. Since due to Claim 8, $\mathcal{Q}$ contains any crucial edge with probability at least $1-\varepsilon$, this implies that $Q \cup M_c$ weights, in expectation, at least $(1-\varepsilon)\alpha$ times the contribution of crucial edges to OPT. It is important to note that we apply the algorithm $\mathcal{VB}$ to all realized critical edges, not exclusively those within Q (the queried ones). This approach ensures that the output of $\mathcal{VB}$, consisting of M_c and set A, is independent of the choice of Q, as outlined in Claim 13.

The next step of the reduction is using the non-crucial edges among vertices in A to construct a fractional matching $\boldsymbol{f}$. In Lemma 20, we use properties of $\mathcal{VB}$ to ensure that the expected contribution of any non-crucial edge to $\boldsymbol{f}$ is almost the same as its contribution to the optimal solution. We then use the fact that these edges are non-crucial (hence have small f_es) to prove in Lemma 15 that $\boldsymbol{f}$ has almost no integrity gap. Putting these pieces together, we prove that either union of this rounded matching and M_c is an $(\frac{1}{2-\alpha}-\varepsilon)$ approximate solution, or simply only using the crucial edges in $\mathcal{Q}$ gives us this approximation.

In the following claim, we prove two basic properties about M_c and set A and their relation to the set of non-crucial edges in $\mathcal{Q}$.

▷ Claim 13. Let M_c and A be the outputs of an α-approximation variance-bounding algorithm which takes as input the subgraph $H=(V,C)$ and matching $M_{\mathcal{O}}$ defined in 12. We have the followings for M_c and A:

1. The expected weight of matching M_c is at least α times the weight of matching $\text{OPT} \cap C$ (i.e., the contribution of the crucial edges to the optimal solution).
2. Let Q be the subgraph of edges we choose to query. For any non-crucial edge $e \in N$, the event $e \in \mathcal{Q}$ is independent of both M_c and A.

Proof. To prove the first item of this claim, we will first show that matching $M_{\mathcal{O}}$ defined in 2 has the same expected weight as the contribution of the crucial edges to the optimal solution. In other words, $\mathbf{E}[\mathsf{W}(\text{OPT} \cap C)] = \mathbf{E}[M_{\mathcal{O}}]$. Recall that we have defined $M_{\mathcal{O}} = \mathsf{MWM}(\mathcal{H} \cup \mathcal{N}^\star) \cap \mathcal{H}$, where $\mathcal{H} = \mathcal{G} \cap H$ is the actual realization of all the crucial edges, and $\mathcal{N}^\star$ is a random hallucination of the non-crucial edges containing each edge independently with probability p_e. This implies that $\mathcal{H} \cup \mathcal{N}^\star$ comes from the same distribution as $\mathcal{G}$ and as result $M_{\mathcal{O}}$ is drawn from the same distribution as OPT. For any crucial edge $e \in C$ this gives us $\Pr[e \in M_{\mathcal{O}}] = \Pr[e \in \text{OPT} \cap C]$ and $\mathbf{E}[\mathsf{W}(\text{OPT} \cap C)] = \mathbf{E}[M_{\mathcal{O}}]$. This proves the first part of the claim since due to Definition 10, property (i) we know M_c is an α approximation with respect to $M_{\mathcal{O}}$.

(To prove the second part of this claim, observe that event $e \in \mathcal{Q}$ is a function of Q and the realization of non-crucial edges, while M_c and A are obtained from running a variance-bounding matching algorithm with inputs $H=(V,C)$ and $M_{\mathcal{O}} = \mathsf{MWM}(\mathcal{H} \cup \mathcal{N}^\star) \cap H$. Here, $\mathcal{H}$ is the actual realization of all the crucial edges while $\mathcal{N}^\star$ is a random hallucination of the non-crucial edges (not the actual realization). Graph $H=(V,C)$ and function $\mathsf{MWM}(.)$ are deterministic which means the only randomization in determining values of M_c and A comes from $\mathcal{H} \cup \mathcal{N}^\star$. Since edges of $\mathcal{G}$ are realized independently, $\mathcal{H} \cup \mathcal{N}^\star$ is independent of the actual realization of the non-crucial edges. It clearly is also independent of the choice of Q. This implies that knowing the outcome of event $e \in \mathcal{Q}$ does not change the distribution of M_c and A; hence they are independent. ◁

5.1 A Fractional Matching on the Non-crucial Edges

In this section, we will construct a fractional matching on the non-crucial edges to augment the matching we get from running a variance-bounding matching on the crucial edges. Given a variance-bounding algorithm $\mathcal{VB}$, let M_c and A be the output of $\mathcal{VB}$ with inputs given according to Definition 12. To begin, let us define a variable g_e for any non-crucial edge as follows:

$$g_e = \frac{x_e}{\Pr[e \in \mathcal{Q}] \Pr[u, v \in A]}, u \tag{3}$$

where x_e is defined as

$$x_e = \Pr[e \in \text{OPT}].$$

Ideally, for constructing our fractional matching, we would assign a fractional value of g_e to edge e whenever $e \in \mathcal{Q}$ and both of its endpoints are in A. Since these events are independent due to Claim 13, their joint probability is $\Pr[e \in \mathcal{Q}] \Pr[u, v \in A]$. By constructing a fractional matching in this manner, we achieve $\mathbf{E}[f_e] = x_e$ for any edge e and $\mathbf{E}[\boldsymbol{f} \cdot \boldsymbol{w}] = \mathbf{E}[\mathsf{W}(\text{OPT})]$.

However, the challenge lies in the fact that constructing $\boldsymbol{f}$ in this way may result in it not being a valid fractional matching, as certain vertices may have a fractional degree greater than one. In other words, $\sum_{(u,v)\in N} f_{(u,v)} > 1$ may occur for some vertices $v \in V$. To address this issue, we first scale down the fractional values by a small amount. Subsequently, we discard any vertex whose fractional degree still exceeds one. The challenge then becomes demonstrating that this event does not significantly reduce the expected size of the fractional matching. We formally state the algorithm for constructing a fractional matching on the non-crucial edges in Algorithm 2.

Algorithm 2 A fractional matching on the realized non-crucial edges.

Let M_c and A be the outputs of an α-approximation variance-bounding matching algorithm with inputs given according to Definition 12.
Let $\boldsymbol{f}$ be an empty fractional matching on the subgraph of non-crucial edges.
Let $\varepsilon \in (0, 1)$ be a small given constant (the same as the one used in Table 1).
Set $\gamma = (1 - \varepsilon^2)/(1 + \frac{3\varepsilon}{10})$.
for each $e = (u, v) \in N$ **do**
    Let $g_e = x_e / \Pr[e \in \mathcal{Q} \text{ and } \{u, v\} \subset A]$
    if $e \in \mathcal{Q}$ and both endpoints are in A **then**
        Set $f_e = g_e \gamma$
    else
        Set $f_e = 0$
    end if
end for
If the fractional degree of a vertex v in $\boldsymbol{f}$ exceeds one (i.e., $\sum_{e \ni v} f_e > 1$), zero out the fractional value of its edges.
return $\boldsymbol{f}$

Since our ultimate goal is to demonstrate the existence of a large weight integral matching on $\mathcal{Q}$ rather than a fractional one, let us first address the integrality gap of the fractional matching produced by this algorithm. We first prove an upper bound of ε^3 for g_e of non-crucial edges in Claim 14. We then use this in Lemma 15 to prove that the output of Algorithm 2 has a small integrality gap. To help with the flow of the paper, both proofs are deferred to the full version.

▷ Claim 14. By choosing a sufficiently small constant $\varepsilon > 0$ in Algorithm 2, we get $g_e \leq \varepsilon^3$ for all non-crucial edges.

▶ **Lemma 15.** *Consider the fractional matching $\boldsymbol{f}$ produced by Algorithm 2. There exists an integral matching on the non-crucial edges of $\mathcal{Q}$ between vertices in A with weight at least $(1-\varepsilon/2)\boldsymbol{f}\cdot\boldsymbol{w}$.*

Survival of vertices and non-crucial edges. For any vertex $v \in V$, we say v *survives* Algorithm 2 iff it is in set A, and it is not killed in Line 13 of the algorithm (i.e., its fractional degree is not reduced to zero). We also say an edge e survives the algorithm iff both its endpoints survive (regardless of whether it is in $\mathcal{Q}$ or not).

5.2 Expected Weight of the Fractional Matching

Let $\boldsymbol{f'}$ denote the value of $\boldsymbol{f}$ constructed by Algorithm 2 before it zeroes out certain f_e values in Line 13. As discussed earlier in this section, it is evident that $\mathbf{E}[f'_e] = \gamma x_e$ for any edge in $e \in N$. Since γ deviates from one by a small constant, the expected weight of $\boldsymbol{f'}$ is a sufficiently large approximation relative to the contribution of the non-crucial edges to the optimal solution. Thus, the primary challenge lies in proving that we do not incur a substantial loss by zeroing out certain f_e values in Line 13. Roughly speaking, we only have the opportunity to use an edge $e = (u, v)$ whenever it is in $\mathcal{Q}$ and its endpoints are in A (i.e., $f'_e \neq 0$), and we lose this opportunity if at least one of its endpoints does not survive Algorithm 2. That is, we have:

$$\Pr[f_e \neq 0] = \Pr[f'_e \neq 0] - \Pr[u \text{ or } v \text{ does not survive } \mid f'_e \neq 0].$$

To quantify the amount of loss per edge, we need to upper-bound $\Pr[u \text{ or } v \text{ do not survive } \mid f'_e \neq 0]$ and show that it is significantly smaller compared to $\Pr[f'_e \neq 0]$. Note that here, $f'_e \neq 0$ is not independent of e's end-points surviving since it contributes to their fractional degree. Furthermore, $e \in Q$ is correlated, albeit negatively (see Claim 25), with the existence of its neighboring non-crucial edges in Q, which may also impact the fractional degrees of u and v in $\boldsymbol{f'}$. However, it is still helpful to first upper-bound the probability of u and v surviving without conditioning on $f'_e \neq 0$. The intuition behind this is that since f'_e is very small (i.e., upper-bounded by ε^3 due to Claim 14), its impact on the fractional degree of each endpoint is insignificant. Moreover, since $e \in Q$ is negatively associated with $e' \in Q$ for any non-crucial $e' \neq e$ connected to u or v, conditioning on $e \in Q$ does not increase their $f'_{e'}$. To upper-bound $\Pr[v \text{ does not survive}]$ for any vertex v, let us define

$$Y_v = \sum_{e=(u,v)\in N} g_e \cdot \mathbb{1}_{u\in A} \cdot \mathbb{1}_{e\in\mathcal{Q}}. \tag{4}$$

Since we set $f'_e = g_e\gamma$ iff $e \in Q$ and $\{u, v\} \subset A$, whenever vertex v is present in A we have

$$Y_v/\gamma = \sum_{e=(u,v)\in N} f'_e.$$

Hence, vertex v survive Algorithm 2 iff $Y_v/\gamma \leq 1$. In Lemma 16, we prove that random variable Y_v is concentrated around its mean for any vertex. This analysis crucially relies on Property (iv) of variance-bounding algorithms. This would have been enough if we knew $\mathbf{E}[Y_v]$ is sufficiently close to one. While we do not exactly have this, we can use the

second property of the variance-bounding algorithms to show $\mathbf{E}[Y_v | v \in A] \leq 1$. This is only doable thanks to Property (ii). Combining all these together, we are able to finally prove in Lemma 17 that Y_v is sufficiently close to one, with a sufficiently large portability.

Since both Lemma 16 and Lemma 17 have lengthy and technical proofs, we respectively allocate Section A.1 and another section in the full version to present detailed proofs for them. Finally, we put the pieces together in Lemma 20 to demonstrate that $\mathbf{E}[f_e]$ is sufficiently large compared to x_e (the contribution of e to the optimal solution).

▶ **Lemma 16.** *For any vertex $v \in V$ define random variable*

$$Y_v = \sum_{e=(u,v)\in N} g_e \cdot \mathbb{1}_{u\in A}\mathbb{1}_{e\in\mathcal{Q}},$$

where $g_e = \frac{x_e}{\Pr[e\in\mathcal{Q}]\cdot\Pr[\{u,v\}\subset A]}$. The following inequality holds for these random variables.

$$\Pr\Big[|Y_v - \mathbb{E}[Y_v]| \geq \eta\Big] \leq \beta$$

for $\beta = \frac{\varepsilon^2}{100}$ and $\eta = \frac{\varepsilon}{10}$.

▶ **Lemma 17.** *For any vertex $v \in V$ we have:*

$$\Pr[Y_v \geq 1 + 3\eta] \leq \beta$$

The proof is deferred to the full version of the paper.

▶ **Definition 18.** *For a vertex u' we define $Y_v(-u')$ to be the summation that we have for Y_v except for the edge $e' = (u', v)$. Formally:*

$$Y_v(-u') = \sum_{e=(u,v)\in N, u\neq u'} g_e \cdot \mathbb{1}_{u\in A}\mathbb{1}_{e\in\mathcal{Q}},$$

▶ **Lemma 19.** *For every edge $e' = (v, u')$ and constant $\lambda \in (0, 1)$ we have:*

$$\Pr[Y_v(-u') > \lambda] \geq \Pr[Y_v(-u') > \lambda \mid e' \in \mathcal{Q}]$$

▶ **Lemma 20.** *For every non-crucial edge $e = (u, v)$ we have $\mathbf{E}[f_e] \geq (1 - \varepsilon/2) \cdot x_e$.*

Due to space constraints, we defer the proof of these two lemmas to the full version.

5.3 Proof of Lemma 11 (The Reduction)

In this section, we will put all the pieces together to formally prove Lemma 11. Let Q be the subgraph outputted by Algorithm 1. We prove that the existence of an α-approximation variance-bounding matching algorithm means $\mathcal{Q}$, the realization of Q, contains a $\frac{1}{2-\alpha} - \varepsilon$ approximate solution. Since Q is the union of $t = \frac{1}{\tau\varepsilon}$ matchings, plugging in the value of $\tau = p\varepsilon^5\delta^2$ from Table 1 implies Q has max-degree $O_\varepsilon(1/p)$. Therefore, to prove this lemma, it suffices to show that $\mathcal{Q}$ contains a $\frac{1}{2-\alpha} - \varepsilon$ approximate solution.

Let M_c and A be the outputs of the α-approximation variance-bounding algorithm on inputs specified in Definition 12. Recall that M_c is a matching on the crucial edges and A is a subset of vertices unmatched in M_c. Let σ be the ratio of the optimal solution that comes from the crucial edges. That is

$$\sigma = \frac{\sum_{e\in C} \Pr[e \in \textsc{opt}] w_e}{\mathsf{W}(\textsc{opt})}.$$

Due to Claim 13, we know that the expected weight of M_c is $\alpha\sigma$ fraction of the optimal solution. Furthermore, we showed in Claim 8 that any crucial edge belongs to Q with probability at least $(1-\varepsilon)$. As a result we have

$$\mathbf{E}[\mathsf{W}(M_c \cap \mathcal{Q})] \geq (1-\varepsilon)\alpha\sigma\mathsf{W}(\textsc{opt}). \tag{5}$$

The next step is to use the non-crucial edges among vertices in A to augment $M_c \cap \mathcal{Q}$. In Lemma 20, we prove that it is possible to construct a fractional matching $\boldsymbol{f}$ on the non-crucial edges among vertices in A such that for any non-crucial edge

$$\mathbf{E}[f_e] \geq (1-\varepsilon/2)\Pr[e \in \textsc{opt}].$$

Hence, $\mathbf{E}[\boldsymbol{f}\boldsymbol{w}] \geq (1-\sigma)(1-\varepsilon/2)\mathsf{W}(\textsc{opt})$. As a result of Lemma 15 it is possible to round $\boldsymbol{f}$ and achieve an integral matching M_n such that

$$\mathbf{E}[\mathsf{W}(M_n)] \geq (1-\varepsilon/2)(1-\sigma)(1-\varepsilon/2)\mathsf{W}(\textsc{opt}) \geq (1-\varepsilon)(1-\sigma)\mathsf{W}(\textsc{opt}). \tag{6}$$

Putting Equation (5) and Equation (6) together implies the existence of a matching in $\mathcal{Q}$ with expected weight at least

$$(1-\varepsilon)\alpha\sigma\mathsf{W}(\textsc{opt}) + (1-\varepsilon)(1-\sigma)\mathsf{W}(\textsc{opt}) = (1-\varepsilon)\mathsf{W}(\textsc{opt})(1-\sigma+\alpha\sigma).$$

We claim that the best of this matching and simply taking the max-weight matching among the crucial edges of $\mathcal{Q}$ gives us the desired approximating ratio. Since each crucial edge belongs to Q w.p. at least $1-\varepsilon$, its realization contains a matching with expected weight at least $(1-\varepsilon)$ times the contribution of crucial edges to the optimal solution which is $(1-\varepsilon)\sigma\mathsf{W}(\textsc{opt})$. The best of these two solutions gives us the approximation ratio of at least

$$(1-\varepsilon)\cdot\max(\sigma, 1-\sigma+\alpha\sigma) \geq (1-\varepsilon)\frac{1}{2-\alpha} \geq \frac{1}{2-\alpha} - \varepsilon.$$

Hence, this implies that the realization of subgraph Q with max-degree $O_\varepsilon(1/p)$ contains a $(\frac{1}{2-\alpha}-\varepsilon)$- approximate solution completing the proof of Lemma 11.

6 A Variance-Bounding Matching Algorithm

In this section, we discuss the existence of variance-bounding matching algorithms (defined in 10) for general and bipartite graphs. We will show that any α-selectable batched random-order contention resolution schemes (RCRS) [13] can be used to get a variance bounding matching algorithm with an approximation ratio of almost α.

▶ **Lemma 21** (Variance-bounding Matching Lemma). *For any sufficiently small constant $\delta \in (0,1)$, there exists a $(0.535 - 6\sqrt{\delta})$ approximation variance-bounding matching algorithm for general graphs and a $(1-1/e-6\sqrt{\delta})$ approximation one for bipartite graphs, which satisfy the third property with parameter δ.*

To prove this lemma, we will design an algorithm that achieves the properties of a variance-bounding matching algorithm discussed in Definition 10. Let us start with the definition of batched RCRS, which we will use in the design of our algorithms.

Batched RCRS. Suppose we are given a graph $G = (V, E)$ along with a fractional matching $\boldsymbol{y}$ on the graph. The graph is revealed in an online manner as follows. Vertices arrive in a uniformly random order given by a permutation π. Upon the arrival of a vertex v, the status of the edges connecting v to vertices before it (i.e., all vertices u that $\pi_u < \pi_v$) are revealed, namely a *batch* of edges. Then, at most, one of the edges in the batch becomes *active* such that

$$\Pr[e \text{ becomes active}] = y_e$$

for any edge e in the batch. A batched RCRS decides, upon the arrival of each vertex, irrevocably whether to select the active edge (if any exists). At any point in time, the selected edges must form a matching. Given a parameter α a batched RCRS is called α-selectable if it picks each active element with probability at least α.

The best-known batched RCRS for general graphs is by MacRury and Ma [16] and has $\alpha = 0.535$. For bipartite graphs, there is a batched RCRS with $\alpha = 1 - \frac{1}{e}$ due to Gamlath, Kale, and Svensson. [14].

▶ **Proposition 22** ([16] and [14]). *There exists a* 0.535*-selectable batched RCRS for general graphs and a* $(1 - 1/e)$*-selectable batched RCRS for bipartite graphs.*

We state our variance-bounding matching algorithm formally in Algorithm 3. The algorithm starts by drawing a random permutation π over the vertices uniformly at random. We then let the vertices arrive in the order given by this permutation. Upon arrival of a vertex v, we look at the realization of its edges to the vertices with smaller π_u. Then, a random process decides which one of its edges (if any) becomes active. We explain this process in Definition 23. The process is designed in a way that the probability of each edge becoming active is $(1 - 6\sqrt{\delta}) \Pr[e \in M_{\mathcal{O}}]$ where $M_{\mathcal{O}}$ is the random matching in the statement of Lemma 21 and δ is the parameter from the third property of variance-bounding matching algorithms.

▶ **Definition 23.** *(Edge Activation Process) The activation probability of the edges in this process comes from matching* $M_{\mathcal{O}}$ *and parameter* δ *in the statement of Definition 10. (Recall that* $M_{\mathcal{O}}$ *is a matching on the realized crucial edges.) See Definition 12 for what we set* $M_{\mathcal{O}}$ *to. Let us define*

$$y_e = \Pr[e \in M_{\mathcal{O}}], \tag{7}$$

where the randomization comes from the realization of edges in $\mathcal{H}$ *and the algorithm for finding* $M_{\mathcal{O}}$*. Note that* $\boldsymbol{y}$ *is a fractional matching since each vertex joins* $M_{\mathcal{O}}$ *with probability at most one. Moreover, define set* $E_{v,\pi} = \{u \in E : \pi_u < \pi_v\}$ *to be all of* v*'s edges to vertices* u *with* $\pi_u < \pi_v$*. After looking at the realization of all these edges, let* y'_e *be the probability of* e *being in* $M_{\mathcal{O}}$ *conditioned on the realization of* $E_{v,\pi}$*. That is*

$$y'_e = \Pr[e \in M_{\mathcal{O}} \mid \textit{realization of edges in } E_{v,\pi}]. \tag{8}$$

We activate at most one of the realized edges at random such that the probability of any remaining realized edge e *being the active one is* $g(e) = (1 - 6\sqrt{\delta})y'_e$*. This is possible since* y'_e *of the realized edges sums up to at most one.*

We now have all the required tools to design our algorithm (stated below) which we claim satisfies the properties stated in Lemma 21.

Algorithm 3 Variance-bounding Matching on $H = (V, E)$.

$M_c \leftarrow \emptyset$, $A \leftarrow \emptyset$
Let π be a permutation over vertices in V picked uniformly at random.
for each v in V in the order of π **do**
    Let $E_{v,\pi} = \{u \in E : \pi_u < \pi_v\}$ be all of v's edges to vertices u with $\pi_u < \pi_v$.
    At most one edge $e \in E_{v,\pi}$ becomes active, as described by the edge activation process (Definition 23) according to $M_{\mathcal{Q}}$.
    if $e = (v, u)$ becomes active and u is unmatched in M_c **then**
        The RCCRS from Proposition 22 decides whether e joins matching M_c or not.
    end if
end for
Mark (kill) all vertices who had at least one active edge at some point in the algorithm.
Let A be the set of remaining alive vertices.
return M_c and A

▷ Claim 24. For any permutation π in Algorithm 3, the probability of any edge e becoming active is $(1 - 6\sqrt{\delta})\Pr[e \in M_{\mathcal{O}}]$

Proof. We defer the proof of this claim to the full version due to space constraints. ◁

Now, we are ready to prove Lemma 21 by showing that Algorithm 3 satisfies the properties of a variance-bounding matching algorithm. We defer this to the full version.

References

1 Sepehr Assadi and Aaron Bernstein. Towards a unified theory of sparsification for matching problems. In Jeremy T. Fineman and Michael Mitzenmacher, editors, *2nd Symposium on Simplicity in Algorithms, SOSA 2019, January 8-9, 2019, San Diego, CA, USA*, volume 69 of *OASIcs*, pages 11:1–11:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/OASICS.SOSA.2019.11.

2 Sepehr Assadi, Sanjeev Khanna, and Yang Li. The Stochastic Matching Problem with (Very) Few Queries. In *Proceedings of the 2016 ACM Conference on Economics and Computation, EC '16, Maastricht, The Netherlands, July 24-28, 2016*, pages 43–60, 2016. doi:10.1145/2940716.2940769.

3 Sepehr Assadi, Sanjeev Khanna, and Yang Li. The Stochastic Matching Problem: Beating Half with a Non-Adaptive Algorithm. In *Proceedings of the 2017 ACM Conference on Economics and Computation, EC '17, Cambridge, MA, USA, June 26-30, 2017*, pages 99–116, 2017. doi:10.1145/3033274.3085146.

4 Amir Azarmehr, Soheil Behnezhad, Alma Ghaffari, and Ronitt Rubinfeld. Stochastic matching via in-n-out local computation algorithms. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, 2025.

5 Soheil Behnezhad, Avrim Blum, and Mahsa Derakhshan. Stochastic vertex cover with few queries. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 1808–1846. SIAM, 2022. doi:10.1137/1.9781611977073.73.

6 Soheil Behnezhad and Mahsa Derakhshan. Stochastic weighted matching:$(1-\varepsilon)$ approximation. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1392–1403. IEEE, 2020.

7 Soheil Behnezhad, Mahsa Derakhshan, Alireza Farhadi, MohammadTaghi Hajiaghayi, and Nima Reyhani. Stochastic matching on uniformly sparse graphs. In Dimitris Fotakis and Evangelos Markakis, editors, *Algorithmic Game Theory - 12th International Symposium, SAGT*

2019, Athens, Greece, September 30 - October 3, 2019, Proceedings, volume 11801 of *Lecture Notes in Computer Science*, pages 357–373. Springer, 2019. doi:10.1007/978-3-030-30473-7_24.

8 Soheil Behnezhad, Mahsa Derakhshan, and MohammadTaghi Hajiaghayi. Stochastic matching with few queries: (1-ε) approximation. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1111–1124. ACM, 2020. doi:10.1145/3357713.3384340.

9 Soheil Behnezhad, Alireza Farhadi, MohammadTaghi Hajiaghayi, and Nima Reyhani. Stochastic Matching with Few Queries: New Algorithms and Tools. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2855–2874, 2019. doi:10.1137/1.9781611975482.177.

10 Soheil Behnezhad and Nima Reyhani. Almost Optimal Stochastic Weighted Matching with Few Queries. In *Proceedings of the 2018 ACM Conference on Economics and Computation, Ithaca, NY, USA, June 18-22, 2018*, pages 235–249, 2018. doi:10.1145/3219166.3219226.

11 Avrim Blum, John P. Dickerson, Nika Haghtalab, Ariel D. Procaccia, Tuomas Sandholm, and Ankit Sharma. Ignorance is Almost Bliss: Near-Optimal Stochastic Matching With Few Queries. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC '15, Portland, OR, USA, June 15-19, 2015*, pages 325–342, 2015. doi:10.1145/2764468.2764479.

12 Shaddin Dughmi, Yusuf Hakan Kalayci, and Neel Patel. On sparsification of stochastic packing problems. In Kousha Etessami, Uriel Feige, and Gabriele Puppis, editors, *50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023, Paderborn, Germany*, volume 261 of *LIPIcs*, pages 51:1–51:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ICALP.2023.51.

13 Tomer Ezra, Michal Feldman, Nick Gravin, and Zhihao Gavin Tang. Online stochastic max-weight matching: prophet inequality for vertex and edge arrival models. In *Proceedings of the 21st ACM Conference on Economics and Computation*, pages 769–787, 2020. doi:10.1145/3391403.3399513.

14 Buddhima Gamlath, Sagar Kale, and Ola Svensson. Beating greedy for stochastic bipartite matching. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2841–2854. SIAM, 2019. doi:10.1137/1.9781611975482.176.

15 Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. Local computation: Lower and upper bounds. *Journal of the ACM (JACM)*, 63(2):1–44, 2016. doi:10.1145/2742012.

16 Calum MacRury and Will Ma. Random-order contention resolution via continuous induction: Tightness for bipartite matching under vertex arrivals. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1629–1640, 2024. doi:10.1145/3618260.3649788.

17 J Michael Steele. An efron-stein inequality for nonsymmetric statistics. *The Annals of Statistics*, 14(2):753–758, 1986.

18 Yutaro Yamaguchi and Takanori Maehara. Stochastic Packing Integer Programs with Few Queries. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 293–310, 2018. doi:10.1137/1.9781611975031.21.

A Appendix

A.1 Proof of Lemma 16

We devote this section to prove Lemma 16 due to it being lengthy and technical.

▶ **Lemma 16** (restated). *For any vertex $v \in V$ define random variable*

$$Y_v = \sum_{e=(u,v)\in N} g_e \cdot \mathbb{1}_{u\in A}\mathbb{1}_{e\in\mathcal{Q}},$$

where $g_e = \frac{x_e}{\Pr[e\in\mathcal{Q}]\cdot\Pr[\{u,v\}\subset A]}$. *The following inequality holds for these random variables.*

$$\Pr\Big[|Y_v - \mathbb{E}[Y_v]| \geq \eta\Big] \leq \beta$$

for $\beta = \frac{\varepsilon^2}{100}$ *and* $\eta = \frac{\varepsilon}{10}$.

To prove the desired concentration bound on Y_v we begin by bounding its variance. This will allow us to apply Chebyshev inequality (Proposition 5) to prove our desired bound. Let us first examine the random variables that affect Y_v's value. One collection is the set of variables for presence of vertices after running Algorithm 3 in set A, i.e., $S_A = \{\mathbb{1}_{u\in A} : u \in V\}$ and the second collection is the edges being present in $\mathcal{Q}$, i.e. $S_Q = \{\mathbb{1}_{e\in\mathcal{Q}} : e = (u,v) \in N\}$.

By the law of total variance (Proposition 6) we have:

$$\mathrm{Var}[Y_v] = \mathbf{E}[\mathrm{Var}(Y_v \mid S_A)] + \mathrm{Var}[\mathbf{E}(Y_v \mid S_A)]$$

We will later prove that

$$\mathbf{E}[\mathrm{Var}[Y_v|S_A]] \leq 60\cdot(\varepsilon^6+\varepsilon^5) \tag{9}$$

To bound the term $\mathrm{Var}[\mathbf{E}(Y_v \mid S_A)]$ let us first examine what $\mathbf{E}(Y_v \mid S_A)$ is.

$$\begin{aligned}
\mathbf{E}[Y_v|S_A] &= \sum_{e=(u,v)\in N} \mathbf{E}[g_e\cdot\mathbb{1}_{u\in A}\cdot\mathbb{1}_{e\in Q} \mid S_A] \\
&= \sum_{e=(u,v)\in N} \mathbf{E}\left[\frac{x_e}{\Pr[e\in\mathcal{Q}]\cdot\Pr[\{u,v\}\subset A]}\cdot\mathbb{1}_{u\in A}\cdot\mathbb{1}_{e\in Q} \mid S_A\right] \\
&= \sum_{e=(u,v)\in N} \mathbf{E}\left[\frac{\mathbb{1}_{e\in Q}}{\Pr[e\in\mathcal{Q}]}\right]\mathbf{E}\left[\frac{x_e}{\Pr[\{u,v\}\subset A]}\cdot\mathbb{1}_{u\in A} \mid S_A\right] \\
&= \sum_{e=(u,v)\in N} \mathbf{E}\left[\frac{x_e}{\Pr[\{u,v\}\subset A]}\cdot\mathbb{1}_{u\in A}\right]
\end{aligned}$$

To go from the second to the third line, we are using the fact that A and $\mathcal{Q}$ are independent due to Claim 13. Note that the term in the last line is Z_v in Lemma 16. Applying the lemma with the fractional matching x being the edges having $x_e < \tau$ we get that:

$$\mathrm{Var}[\mathbf{E}(Y_v \mid S_A)] \leq \frac{10\tau}{\delta^2}$$

Adding this with what we have from Equation (9) and applying law of total variance we get

$$\mathrm{Var}(Y_v) \leq 60\cdot(\varepsilon^6+\varepsilon^5) + \frac{10\tau}{\delta^2} \tag{10}$$

Since $\tau = 20p\varepsilon^5\delta^2$ by setting ε to be a small enough constant, we can get the bound $\mathrm{Var}[Y_v] \leq \frac{\varepsilon^4}{10^4}$. This will bound the standard deviation of Y_v by $\frac{\varepsilon^2}{100}$ which is used when applying Chebyshev's Inequality (See Proposition 5) on the random variable Y_v. Now that we have $s \leq \frac{\varepsilon^2}{100}$, by applying Chebyshev's Inequality, we get

$$\Pr\Big[|Y_v - \mathbb{E}[Y_v]| \geq c\cdot s\Big] \leq \frac{1}{c^2} \tag{11}$$

Note that we wanted to bound the probability that Y_v deviates from its mean by η. Now if we have $\eta \geq c\cdot s$, we have

$$\Pr\Big[|Y_v - \mathbb{E}[Y_v]| \geq \eta\Big] \leq \Pr\Big[|Y_v - \mathbb{E}[Y_v]| \geq c\cdot s\Big] \tag{12}$$

By replacing value of $\eta = \frac{\varepsilon}{10}$ and the fact that $s \leq \frac{\varepsilon^2}{100}$ we can see that it is enough to set $c = \frac{\varepsilon}{10}$ to satisfy $\eta \geq c \cdot s$. Therefore by combining (11) and (12) we get

$$\begin{aligned}\Pr\Big[|Y_v - \mathbb{E}[Y_v]| \geq \eta\Big] &\leq \Pr\Big[|Y_v - \mathbb{E}[Y_v]| \geq c \cdot s\Big] \\ &\leq \frac{1}{c^2} \\ &\leq \frac{\varepsilon^2}{100} = \beta\end{aligned}$$

Now that we proved the statement of the lemma using Equation (9), we prove it which states $\mathbf{E}[\mathrm{Var}[Y_v|S_A]] \leq 60 \cdot (\varepsilon^6 + \varepsilon^5)$. Our first step is to see how random variables in S_Q behave. First of all, random variables in S_Q are not independent since $\mathcal{Q}$ is a collection of matchings, for two incident edges e_1 and e_2, when e_1 is present in one of the matchings e_2 will not be in that matching. This intuition might make us believe that for edges relevant to S_Q because they all intersect at the vertex v their presence in $\mathcal{Q}$ is pairwise **negatively correlated**. This is in fact true and for proving it we prove a stronger fact about the random variables which is negative association which implies negative correlation. (see Definition 7 for definition).

▶ **Lemma 25.** *Random variables in S_Q are negatively associated.*

Proof. See the full version for the proof. ◀

By definition, negative association implies negative correlation. This means Lemma 25 implies that for two edges $e_1 = (u_1, v), e_2 = (u_2, v)$ such that $e_1, e_2 \in N$ we have:

$$Cov(\mathbb{1}_{e_1 \in \mathcal{Q}}, \mathbb{1}_{e_2 \in \mathcal{Q}}) \leq 0 \tag{13}$$

Let us take an arbitrary realization of variables in S_A and call it $\boldsymbol{A}$. Our plan is, given this fixed $\boldsymbol{A}$, first upper-bound $\mathrm{Var}[Y_v|\boldsymbol{A}]$. Then, using that, find an upper bound for $\mathrm{Var}[Y_v]$. At last, we apply Proposition 5 to prove the statement of the lemma.

Define the random variable

$$X_u = (g_{(u,v)} \cdot \mathbb{1}_{u \in A} \cdot \mathbb{1}_{e \in \mathcal{Q}} | \boldsymbol{A}).$$

We can see that if $\mathbb{1}_{u \in A} = 0$, X_u is always equal to zero, and the inequalities discussed further will be trivial for $\mathrm{Var}[X_u]$. In the case that $\mathbb{1}_{u \in A} = 1$, $X_u = (g_{(u,v)} \cdot \mathbb{1}_{e \in \mathcal{Q}})$. We can see that $(Y_v|\boldsymbol{A}) = \sum_{(u,v) \in N} X_u$. Now, we are ready to bound the variance of Y_v conditioned on $\boldsymbol{A}$. The first step is to bound the variance of X_u:

$$\mathrm{Var}[X_u] = \mathrm{Var}[g_e \cdot \mathbb{1}_{u \in A} \cdot \mathbb{1}_{e \in \mathcal{Q}} | \boldsymbol{A}] \leq \mathrm{Var}[g_e \cdot \mathbb{1}_{e \in \mathcal{Q}} | \boldsymbol{A}] \tag{14}$$

This is because when we have fixed A, in the case that $\mathbb{1}_{u \in A} = 0$ then variance of X_u is zero and in the case that $\mathbb{1}_{u \in A} = 1$ the bound in Equation (14) holds.

Now we know that $\mathrm{Var}[X_u] = \mathbf{E}[X_u^2] - \mathbf{E}[X_u]^2 \leq \mathbf{E}[X_u^2]$ so from Equation (14) we get:

$$\mathrm{Var}[X_u] \leq \mathbf{E}[X_u^2] \leq \mathbf{E}[(g_e \cdot \mathbb{1}_{u \in \mathcal{Q}} | \boldsymbol{A})^2] \leq \mathbf{E}[(g_e \cdot \mathbb{1}_{u \in \mathcal{Q}})^2] \leq \Pr[e \in \mathcal{Q}] \cdot g_e^2 \tag{15}$$

Note that we can remove the condition on $\boldsymbol{A}$ because variables in S_Q and S_A are independent. The last step comes from the fact that with probability $\Pr[e \in \mathcal{Q}]$, $(g_e \cdot \mathbb{1}_{u \in \mathcal{Q}})^2]$ equals g_e^2 and it is zero otherwise. To make further progress, we need a bound on $\Pr[e \in Q]$. The following lemma addresses this.

Expanding g_e in Equation (15), we get:

$$\begin{aligned}\mathrm{Var}[X_u] &\leq \Pr[e \in \mathcal{Q}] \cdot \left(\frac{x_e}{p_e \cdot \Pr[e \in Q] \cdot \Pr[\{u,v\} \subset A]}\right)^2 \\ &\leq \frac{x_e^2}{p_e \cdot \Pr[e \in Q] \cdot (\Pr[\{u,v\} \subset A])^2} \\ &\leq \frac{x_e^2}{p_e \cdot \Pr[e \in Q] \cdot \delta^2} \end{aligned} \tag{16}$$

To go from the first line to the second, first note the distinction between Q and $\mathcal{Q}$ in the equation above. By definition of $e \in \mathcal{Q}$ being $e \in Q \cap e \in \mathcal{G}$ we can see that $\Pr[e \in \mathcal{Q}] = p_e \cdot \Pr[e \in Q]$. This is because $e \in \mathcal{G}$ is independent of $e \in Q$ since Q is constructed on hallucinations of $\mathcal{G}$. To go from the second line to the third line note that in Lemma 21, we showed $\Pr[\{u,v\} \subset A] \geq \delta$.

Moreover, from Claim 9 we know that $\Pr[e \in Q] \geq \min(1/3, tx_e/3)$ so we consider two cases:

Case 1: $\Pr[e \in Q] \geq \frac{t \cdot x_e}{3}$. Combining this and (16) we get:

$$\begin{aligned}\mathrm{Var}[X_u] &\leq \frac{x_e^2}{p_e \cdot \Pr[e \in Q] \cdot \delta^2} \\ &\leq \frac{3x_e^2}{p_e \cdot t \cdot x_e \cdot \delta^2} \\ &\leq \frac{3x_e}{p_e \cdot t \cdot \delta^2}\end{aligned}$$

Case 2: $\Pr[e \in Q] \geq \frac{1}{3}$. Combining this and (16) we get:

$$\mathrm{Var}[X_u] \leq \frac{x_e^2}{p_e \cdot \Pr[e \in Q] \cdot \delta^2} \leq \frac{3x_e^2}{p_e \cdot \delta^2}$$

Now that we have a bound on all $\mathrm{Var}[X_u]$'s we are ready to bound $\mathrm{Var}[Y|\boldsymbol{A}]$. The following proposition is what we need.

▶ **Proposition 26.** *Let X be a random variable written as the sum of random variables $X_1, ..., X_n$. So we have $X = \sum_{i=n}^{n} X_i$. Then we have:*

$$\mathrm{Var}[X] = \sum_{i=1}^{n} \mathrm{Var}[X_i] + 2 \cdot \sum_{i=1}^{n} \sum_{j>i}^{n} Cov(X_i, X_j)$$

In (13) we argued that all variables in S_Q are negatively correlated. Recall the definition of $X_u = g_{(u,v)} \cdot \mathbb{1}_{u \in A} \cdot \mathbb{1}_{e \in \mathcal{Q}}$. Because we have fixed $\boldsymbol{A}$ all X_u's will be equal to zero or $g_{(u,v)} \cdot \mathbb{1}_{e \in \mathcal{Q}}$. Hence we can argue that $Cov(X_u, X_w) \leq 0$. This is because if at least one of them is equal to zero then $Cov(X_u, X_w) = 0$. Otherwise, since g_e's are constants sign of $Cov(X_u, X_w)$ will be the same as $Cov(\mathbb{1}_{(u,v) \in \mathcal{Q}}, \mathbb{1}_{(w,v) \in \mathcal{Q}})$.

Therefore, by applying Proposition 26 to all X_u's and the fact that they are pairwise negatively correlated we get:

$$\mathrm{Var}[Y_v|\boldsymbol{A}] \leq \sum_u \mathrm{Var}[X_u] \leq \sum_u \max\left(\frac{3x_e}{p_e \cdot t \cdot \delta^2}, \frac{3x_e^2}{p_e \cdot \delta^2}\right) \leq \sum_u \frac{3x_e}{p_e \cdot t \cdot \delta^2} + \sum_u \frac{3x_e^2}{p_e \cdot \delta^2} \tag{17}$$

For brevity, we are writing $\sum_u$ instead of $\sum_{(u,v)\in N}$ for all the equations here. To bound the first sum, note that $t = \frac{1}{20\cdot\varepsilon^6\cdot\delta^2\cdot p}$ and also $\sum_{(u,v)\in N} x_e \leq 1$ therefore we have:

$$\sum_u \frac{3x_e}{p_e \cdot t \cdot \delta^2} \leq \sum_u \frac{3 \cdot 20 \cdot \varepsilon^6 \cdot \delta^2 \cdot p_{min} \cdot x_e}{p_e \cdot \delta^2} \leq \sum_u 60 \cdot \varepsilon^6 \cdot x_e \leq 60 \cdot \varepsilon^6 \tag{18}$$

To bound the second sum, note that for non-crucial edges, we have $x_e \leq \tau$. Since we have $\tau = 20 p_{min} \varepsilon^5 \delta^2$ we get:

$$\sum_u \frac{3x_e^2}{p_e \cdot \delta^2} \leq \sum_u \frac{3 \cdot \tau \cdot x_e}{p_e \cdot \delta^2} \leq \sum_u \frac{3 \cdot 20 \cdot \varepsilon^5 \cdot \delta^2 \cdot p_{min} \cdot x_e}{p_e \cdot \delta^2} \leq \sum_u 60 \cdot \varepsilon^5 \cdot x_e \leq 60 \cdot \varepsilon^5 \tag{19}$$

Putting things together we get, $\text{Var}[Y_v|\boldsymbol{A}] \leq 60 \cdot (\varepsilon^6 + \varepsilon^5)$. Now since we have proved this for any arbitary $\boldsymbol{A}$ we can remove the condition on $\boldsymbol{A}$ and get:

$$\mathbf{E}[\text{Var}[Y_v|S_A]] \leq 60 \cdot (\varepsilon^6 + \varepsilon^5) \tag{20}$$

which is exactly Equation (9) so the proof is complete.