ABSTRACT

Title of Dissertation: ALGORITHMS FOR MARKETS: MATCHING AND PRICING

Mahsa Derakhshan
Doctor of Philosophy, 2021

Dissertation Directed by: Professor MohammadTaghi Hajiaghayi
Department of Computer Science

In their most basic form *markets* consist of a collection of resources (goods or services) and a set of agents interested in obtaining them. This thesis is a stepping stone toward answering the most central question in the Econ/CS literature surrounding markets: How should the resources be allocated to the interested parties? The first contribution of this thesis is designing pricing algorithms for modern monetary markets (such as advertising markets) in which resources are sold via auctions. The second contribution is designing matching algorithms for markets in which money often plays little to no role (i.e., matching markets).

Auctions have become the standard method of allocating resources in monetary markets, and when it comes to multi-unit auctions Vickrey–Clarke–Groves (VCG) with *reserve prices* is one of the most well-known and widely used auctions. A reserve price is a minimum price with which the auctioneer is willing to sell the item. In this thesis, we consider optimizing *personalized reserve prices* which are crucial for obtaining a high revenue. To that end, we take a *data-driven* approach where given the buyers' bids in a set of auctions, the goal is to find a single vector of reserve prices (one for each buyer)

that maximizes the total revenue across all these auctions. This problem is shown to be NP-hard, and the best-known algorithm for that achieves a $\frac{1}{2}$ fraction of the optimal revenue. We first present an LP-based algorithm with a 0.68 approximation factor for single-item environments. We then show that this approach can be generalized to get a 0.63-approximation for general multi-unit environments. To achieve these results we develop novel LP-rounding procedures which may be of independent interest.

Matching markets have long held a central place in the mechanism design literature. Examples include kidney exchange, labor markets, and dating platforms. When it comes to designing algorithms for these markets, the presence of uncertainty is a common challenge. This uncertainty is often due to the stochastic nature of the data or restrictions that result in limited access to information. In this thesis, we study the *stochastic matching* problem in which the goal is to find a large matching of a graph whose edges are uncertain but can be accessed via queries. Particularly, we only know the existence probability of each edge but to verify their existence, we need to perform costly queries. Since these queries are costly, our goal is to find a large matching with only a few (a constant number of) queries. For instance, in labor markets, the existence of an edge between a freelancer and an employer represents their compatibility to work with one another, and a query translates to an interview between them which is often a time-consuming process. While this problem has been studied extensively, before our work, the best-known approximation ratio for unweighted graphs was almost $\frac{2}{3}$, and slightly better than $\frac{1}{2}$ for weighted graphs. In this thesis, we present algorithms that find almost optimal matchings despite the uncertainty in the graph (weighted and unweighted) by conducting only a constant number of queries per vertex.

ALGORITHMS FOR MARKETS: MATCHING AND PRICING

by

Mahsa Derakhshan

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2021

Advisory Committee:
Professor MohammadTaghi Hajiaghayi, Chair/Advisor
Professor Lawrence M. Ausubel
Professor Avrim Blum
Professor David Mount
Professor Dana S. Nau
Professor Christos Papadimitriou
Professor David Pennock

Dedication

To my parents.

Acknowledgments

To start, I would like to thank my advisor MohammadTaghi Hajiaghayi. This thesis could not happen without his dedication, advice, and support. I was very fortunate to have such an understanding, supportive and wonderful advisor for my Ph.D. He helped me to grow not only as a researcher but also as a person. I cannot thank him enough.

Thank you as well to everyone else who has mentored me through this journey: Avrim Blum, Negin Golrezaei, Vahab Mirrokni, Christos Papadimitriou, David Pennock, and Alex Slivkins. Also, a special thanks to my dissertation committee for making the time and providing helpful suggestions: Lawrence Ausubel, Avrim Blum, David Mount, Dana S. Nau, Christos Papadimitriou, and David Pennock.

I would also like to thank my collaborators and coauthors for all the things I have learned from them, the brainstorming in meetings, and their support: Mohammad Hossein Bateni, Soheil Behnezhad, Avrim Blum, Sina Dehghani, Hossein Esfandiari, Alireza Farhadi, Negin Golrezaei, Richard Karp, Raimondas Kiveris, Marina Knittel, Silvio Lattanzi, Mohammad Mahdian, Vahideh Manshadi, Vahab Mirrokni, Renato Paes Leme, Christos Papadimitriou, David Pennock, Ronald L. Rivest, Hamed Saleh, Saeed Seddighin, Alex Slivkins, Philip B. Stark, Cliff Stein, Madhu Sudan, Elif Tan, and Hadi Yami.

Thanks as well to my friends for making the complications of the experience of living abroad and the hardships of a Ph.D. much easier: Saba Ahmadi, Nazanin Alipourfard, Alireza Farhadi, Amin Ghiasi, Maryam Negahbani, Kiana Roshanzamir, Ali Shafahi, and Hadi Yami. I could not do this without their emotional support. I would like to thank them all for the many great memories.

I cannot express my gratitude towards my family in words. I want to thank my parents for their dedication to helping me grow and follow my dreams. I could not be where I am without their sacrifices. I am eternally grateful for their unconditional love and support. I also want to thank my sister and brother for their emotional support and encouragement.

Last but not least, I want to thank the person who has been my best friend as far as I remember, my husband (and high school sweetheart) Soheil Behnezhad. He also happens to be the person I have the most academic collaborations with (half of this thesis is the result of such collaborations). His support and unconditional love have helped me to be the person and the researcher that I am today. This thesis would certainly have not been possible without him by my side.

Table of Contents

Chapter 1: Introduction

In their most basic form *markets* consist of a collection of resources (goods or services) and a set of agents interested in obtaining them. This thesis is a stepping stone toward answering the most central question in the Econ/CS literature surrounding markets: How should the resources be allocated to the interested parties? The first contribution of this thesis is designing pricing algorithms for modern monetary markets (such as advertising markets) in which resources are sold via auctions. The second contribution is designing matching algorithms for markets in which money often plays little to no role (i.e., matching markets).

1.1 Auctions

Auctions have become the standard method of allocating resources in monetary markets (e.g., advertising markets). In this thesis, we focus on multi-unit environments in which a set of identical items are sold to a set of unit-demand buyers. First, buyers submit their bids (the amount they are willing to pay for the item), and then the auctioneer determines the allocation of the items and the payments. In such environments, one of the most well-known auctions is the Vickrey-Clarke-Groves (VCG) auction. In the case of a single-item, this auction translates into the second price auction in which the item is allocated to the buyer which the highest bid and the payment equals to the second highest bid.

The VCG mechanism, while *"lovely"* in theory [7], is often criticized for not having a good performance guarantee when it comes to revenue maximization. Hartline and Roughgarden [31] approach this shortcoming by optimizing *reserve prices*, the minimum prices that the seller is willing to sell each item to each buyer. According to a number of theoretical and empirical studies [14, 24, 31, 37], *personalized* reserve prices (i.e., one custom reserve price for each buyer) can significantly improve revenue. In particular, personalized reserves can lead to approximately optimal revenue in quite general settings [31]. This has led to several studies computing an optimal vector of reserve prices [14, 21, 33, 38]. The study of VCG-like mechanisms for revenue optimization is closely aligned with a broader agenda of simple vs. optimal mechanisms [1, 14, 15, 20, 23, 29, 33], and in fact has been one of the starting points for this agenda [31].

In this thesis, we focus on the *eager* version of VCG auctions. [1] Consider a *multi-unit* auction with *unit-demand* buyers: that is, k identical units are available, and each buyer is interested in obtaining only one unit. The auctioneer announces a reserve price r_{b} for each buyer b and then buyers place their bids. We run a VCG auction among the buyers that clear their reserve price (i.e., their bid equals or exceeds their reserve price), and the winners pay the maximum of their own reserve price and their VCG payment. That is, let S be the set of buyers who clear their reserve prices. The first

[1] An alternative, called *lazy* VCG [23], first forms a set of potential winners using a VCG auction, then removes buyers whose bids don't clear their reserve. The eager version is often superior both in theory and in practice [33].

k buyers in S with the highest bids win. Each winning buyer pays the maximum of his/her reserve price and the $(k+1)$-th highest bid.

We adopt a standard data-driven model for computing reserve prices [33, 38]: given a history of buyers' bids over multiple runs of the auction, we compute a reserve price for each buyer to maximize the total revenue attained over the same dataset. An important property of this model is that it does not impose essentially any restrictions on the bid distributions. In particular, buyers' private values can be correlated, in contrast with other models [14, 20, 23, 31] that assume independence. Moreover, any approximation for the data-driven model can also be used in a black-box reduction of Morgenstern and Roughgarden [36] to approximate the *Bayesian Optimization* and *Batch Learning* versions of the problem with (almost) the same approximation-factor.[2]

In the data driven model, this problem was first studied by Roughgarden and Wang [38], who showed that it is APX-hard[3] and gave a 1/2 approximate greedy solution. Furthermore, they prove that their analysis is tight. I.e., their algorithm does not get better than 1/2 approximation for some instances of the problem. This then raises the natural question that whether there is an approximation algorithm with approximation ratio better than 1/2 for a single-item case or more generally? In this thesis, we answer this question in the affirmative and in fact achieve a significantly better approximation ratio of 0.63 for any k (not necessarily constant) [22], and a approximation ratio of 0.68 for the case of a single item [21] (the second price auctions).

[2]In Bayesian Optimization, the buyers' private value distributions are independent and known to the algorithm. In Batch Learning, these distributions are unknown, and we only have access to samples drawn from them.

[3]NP-hard to approximate better a fixed constant factor.

Main Result 1. *Consider the data-driven model for eager VCG auctions. There exists a polynomial-time algorithm for computing personalized reserve prices which achieves a* 0.63*-approximation in expected revenue. Furthermore, this approximation ratio can be improved to* 0.68 *for the special case of a single item (i.e., the second price auction).*

In Chapter 2, we discuss our result for the special case of the second price auctions, and in Chapter 3, we discuss our result for the general case.

1.2 Matchings

Matching markets have long held a central place in the mechanism design literature. Examples include labor markets, dating apps, the school choice program, and most importantly, the life-saving kidney exchange program. Kidney exchange provides a market for patients with end-stage renal failure to swap willing but incompatible donors. Basic information about donors and patients (e.g., blood types) can be used as an early indicator of compatibility. However, they are not conclusive, and extra medical laboratory tests are needed to estimate the odds of a successful transplant more accurately. These medical tests are both time-consuming and expensive; thus, each patient can afford only a limited number of them. As such, a fundamental problem here is to design an algorithm for determining which pairs of patient donors should be tested in order to find a large matching (maximize the number of transplants) given that only a constant number of tests can be performed for any patient.

The problem above was first formalized in the pioneering work of Blum *et al.* [17], as the *stochastic matching problem*, in which the goal is to find a large matching of a graph whose edges are uncertain but can be accessed via queries. Particularly, we only know the existence probability of each edge but to verify their existence, we need to perform costly queries. For instance, in kidney exchange, the existence of an edge between two pairs of donor patients represents their compatibility, and queries are equivalent to the expensive medical tests needed to determine these compatibilities. For this problem, Blum et al. give an algorithm that finds a 1/2-approximate matching and leave it as an open problem to decide the best achievable approximation ratio. In a long line of work [4, 5, 6, 9, 11, 12, 17, 41], the approximation was improved to up to 2/3 for unweighted graphs [4] and slightly above 0.5 for weighted graphs. The best achievable approximation ratio, however, remained a mystery until our work [10], where we prove that the problem admits an $(1-\varepsilon)$-approximation for any desirably small constant $\varepsilon > 0$. Pleasingly, while having an intricate analysis, the construction of our subgraph is through a simple sampling algorithm. We then, generalize our result to the weighted version of the problem [8], and prove that augmenting the same algorithm with a greedy one that favors picking high-weight edges surprisingly guarantees a $(1-\varepsilon)$-approximation of the maximum weight matching using only a constant number of queries per vertex.

Main Result 2. *For any (weighted) graph G, any $p \in (0,1]$, and any $\varepsilon > 0$, there is a subgraph Q of G with maximum degree $O_{\varepsilon,p}(1)$ that achieves a $(1-\varepsilon)$-approximation for the stochastic weighted matching problem.*

We discuss our results for the unweighted and weighted versions of the problem in Chapter 4 and Chapter 5 respectively.

Chapter 2: Personalized Reserve Prices in Second Price Auctions

In this chapter, we focus on data-driven optimization of personalized reserve prices in the *eager* second price auction (i.e., single-unit VCG auction). We have a single item and n unit-demand buyers participating in a set of eager second price auctions. Let A and B respectively be the set of auctions and buyers. We are given a dataset of bids $\boldsymbol{\beta}$ where for any auction $\mathsf{a} \in \mathsf{A}$ and buyer $\mathsf{b} \in \mathsf{B}$, $\beta_{\mathsf{a},\mathsf{b}}$ represents bid of buyer b in auction a. Let r_{b} be the personalized reserve price of buyer $\mathsf{b} \in \mathsf{B}$. Then, given the bids $\{\beta_{\mathsf{a},\mathsf{b}}\}_{\mathsf{b}\in\mathsf{B}}$ in auction $\mathsf{a} \in \mathsf{A}$ and reserve prices $\mathbf{r} = \{r_{\mathsf{b}}\}_{\mathsf{b}\in\mathsf{B}}$, the eager second price (ESP) auction works as follows:

1. First, any buyer b with $\beta_{\mathsf{a},\mathsf{b}} < r_{\mathsf{b}}$ is eliminated. Let $S_{\mathsf{a}} = \{\mathsf{b} : \beta_{\mathsf{a},\mathsf{b}} \geq r_{\mathsf{b}}\}$ be the set of buyers who clear their reserve prices in auction a.

2. When set S_{a} is nonempty, the item is allocated to buyer $\mathsf{b}^\star_{\mathsf{a}} = \arg\max_{\mathsf{b}\in S_{\mathsf{a}}} \{\beta_{\mathsf{a},\mathsf{b}}\}$ who has the highest bid among all the buyers in set S_{a} and is charged

$$\mathsf{Rev}_{\mathsf{a}}(\mathbf{r}) := \max\left\{ r_{\mathsf{b}^\star_{\mathsf{a}}},\ \max_{\mathsf{b}\in S_{\mathsf{a}}, \mathsf{b}\neq \mathsf{b}^\star_{\mathsf{a}}} \{\beta_{\mathsf{a},\mathsf{b}}\} \right\}.$$

Note that S_{a} and $\mathsf{b}^\star_{\mathsf{a}}$ implicitly depend on reserve prices $\mathbf{r}$. Any other buyer $\mathsf{b} \in \mathsf{B}$, $\mathsf{b} \neq \mathsf{b}^\star_{\mathsf{a}}$ is not charged. Further, when set S_{a} is empty, the item is not allocated and $\mathsf{Rev_a}(\mathbf{r}) = 0$.

Given the dataset of bids $\boldsymbol{\beta}$, our goal is to find a vector of personalized reserve prices that maximize revenue of the auctioneer. Note that the reserve prices are the same across all the auctions $\mathsf{a} \in \mathsf{A}$. However, each buyer b is assigned a personalized reserve price r_{b}. Formally, we would like to solve the following optimization problem:

$$\mathsf{ESP}^{\star} \;=\; \max_{\mathbf{r}\in\mathbb{R}^n} \;\; \mathsf{Rev}(\mathbf{r}) := \sum_{\mathsf{a}\in\mathsf{A}} \mathsf{Rev_a}(\mathbf{r})\,. \qquad \text{(ESP-OPT)}$$

Note that, without loss of generality, we assume that the optimal reserve price for buyer b is equal to one of his submitted bids $\{\beta_{\mathsf{a,b}}\}_{\mathsf{a}\in\mathsf{A}}$. Let $\mathsf{R} = \{0, \infty\} \cup \{\beta_{\mathsf{a,b}}\}_{\mathsf{a}\in\mathsf{A},\mathsf{b}\in\mathsf{B}}$. Then, Problem ESP-OPT can be rewritten as $\max_{\mathbf{r}\in\mathsf{R}^n} \; \sum_{\mathsf{a}\in\mathsf{A}} \mathsf{Rev_a}(\mathbf{r})$, which leads to a search space of size $|\mathsf{R}|^n$.

2.1 Results and Techniques

The main contribution of this chapter is a randomized algorithm that returns an 0.684-approximation solution for Problem ESP-OPT.

Theorem 1 (Main Theorem)**.** *There exists a randomized polynomial time algorithm that given a data-set $\{\beta_{\mathsf{a,b}}\}_{\mathsf{a}\in\mathsf{A},\mathsf{b}\in\mathsf{B}}$, outputs a vector of eager reserve prices whose expected revenue is a* 0.684*-approximation of that of the optimal value of Problem* ESP-OPT*, denoted by* $\mathsf{ESP}^{\star}$*.*

To find an approximate solution, the overall idea is to construct an LP whose objective function at its optimal solution provides an upper bound for $\mathsf{ESP}^{\star}$. The LP that takes advantage of a concise representation of the solution space, has a polynomial number of variables and constraints. Then, we use a rounding technique to transform the optimal solution of the LP to a vector of reserve prices. We show that if we consider the reserve prices obtained from the rounding technique and the vector of all-zero reserve prices and choose the one with the maximum revenue, we obtain the desired approximation factor. In Theorem 3, we further show that our analysis of our approximation factor is tight. That is, we provide an example for which our algorithm obtains exactly 0.684 fraction of the optimal value of the LP, i.e., the upper bound on for $\mathsf{ESP}^\star$. Finally, in Theorem 2, we bound the integrality gap of the LP. This characterization shows that no algorithm can obtain more than 0.828 fraction of the LP.

2.2 Linear Program

The main challenge in designing an LP formulation for this problem is to find a concise representation of the solution space. Instead of considering all possible assignments of reserves to buyers, we will consider only partial assignments in which we only specify the reserve prices of two buyers. We will call such partial assignment a *profile.* Formally, a profile is a tuple $p = (\mathsf{b}_1, \mathsf{b}_2, r_1, r_2) \in \mathsf{B} \times \mathsf{B} \times \mathsf{R} \times \mathsf{R}$, which represents an assignment of reserve r_1 to buyer b_1 and reserve r_2 to buyer b_2. If it is the case that the reserves are below the corresponding bids in an auction a, i.e. $r_1 \leq \beta_{\mathsf{a},\mathsf{b}_1}$ and $r_2 \leq \beta_{\mathsf{a},\mathsf{b}_2}$, then no matter how the assignment of the remaining reserves, the revenue of this partial assignment is at least $\max\{r_1, \beta_{\mathsf{a},\mathsf{b}_2}\}$ for $\beta_{\mathsf{a},\mathsf{b}_1} \geq \beta_{\mathsf{a},\mathsf{b}_2}$. We also note that given any vector of reserve prices $\mathbf{r}$, the revenue that can be obtained from $\mathbf{r}$ only depends on the reserve price of the highest and second highest bidders that clear the reserve prices.

Next, we formally define the notion of *valid profile* and show that the Problem (ESP-OPT) can be relaxed to find the best consistent distribution over valid profiles in each auction. To define valid profiles, we assume that in each auction a, we have two auxiliary buyers b_0 and b_{00} who always bid zero. That is, $\mathsf{b}_{00}, \mathsf{b}_0 \in \mathsf{B}$, and $\beta_{\mathsf{a},\mathsf{b}_0} = \beta_{\mathsf{a},\mathsf{b}_{00}} = 0$ for any $\mathsf{a} \in \mathsf{A}$.

Definition 2.2.1 (Valid Profiles). *We define the set of valid profiles for auction* a *as the set* $\mathcal{P}_a$ *consisting of all tuples* $(\mathsf{b}_1, \mathsf{b}_2, r_1, r_2) \in \mathsf{B} \times \mathsf{B} \times \mathsf{R} \times \mathsf{R}$ *satisfies the following conditions:*

1. *Bid of buyer* b_1 *is greater than or equal to that of buyer* b_2*; that is,* $\beta_{\mathsf{a},\mathsf{b}_1} \geq \beta_{\mathsf{a},\mathsf{b}_2}$.
2. *Buyer* b_1 *clears his reserve; that is,* $\beta_{\mathsf{a},\mathsf{b}_1} \geq r_1$.
3. *Buyer* b_2 *clears his reserve; that is,* $\beta_{\mathsf{a},\mathsf{b}_2} \geq r_2$.

For any given profile $p \in \mathcal{P}_a$*, we define* $\mathsf{Rev}_{\mathsf{a}}(p) := \max(\beta_{\mathsf{a},\mathsf{b}_2}, r_1)$.

We note that any valid profile corresponds to at least one vector of reserve prices. To see why, observe that we can always obtain $p = (\mathsf{b}_1, \mathsf{b}_2, r_1, r_2)$ by setting $r_{\mathsf{b}_1} = r_1$, $r_{\mathsf{b}_2} = r_2$, and $r_{\mathsf{b}} = \infty$ for any $\mathsf{b} \neq \mathsf{b}_1, \mathsf{b}_2$. Of course, there may exist other vectors of reserve prices that lead to the same profile. We note that by adding buyers b_0 and b_{00} to B, we can define valid profiles to represent the cases in which less than two buyers cleared their reserve prices. We present the cases with one (respectively zero) cleared buyer with valid profile of $(\mathsf{b}_1, \mathsf{b}_0, r_1, 0)$ (respectively $(\mathsf{b}_0, \mathsf{b}_{00}, 0, 0)$).

Definition 2.2.2 (Profiles Associated with Reserve Prices)**.** *Given a vector of reserve prices* $\mathbf{r}$ *we say a valid profile* $p = (\mathsf{b}_1, \mathsf{b}_2, r_1, r_2)$ *is the unique profile associated with* $\mathbf{r}$ *in an auction* $\mathsf{a} \in \mathsf{A}$ *if and only if the following condition hold. After applying the reserve prices* $\mathbf{r}$*, buyer* b_1 *with reserve* r_1 *and buyer* b_2 *with reserve* r_2 *have the highest and second highest cleared bids in auction* a*, respectively.*

Given a vector of reserve prices $\mathbf{r}$ and an auction a, let p be the profile associated with $\mathbf{r}$ in a. Then, with a slight abuse of notation, we define $\mathsf{Rev}_{\mathsf{a}}(\mathbf{r}) = \mathsf{Rev}_{\mathsf{a}}(p)$.

We are now ready to describe our LP. The LP will have two sets of variables:

1. For any auction $\mathsf{a} \in \mathsf{A}$ and any valid profile $p \in \mathcal{P}_{\mathsf{a}}$, define a variable $s_{\mathsf{a},p} \geq 0$ such that $\sum_{p \in \mathcal{P}_{\mathsf{a}}} s_{\mathsf{a},p} \leq 1$. This variable represents a probability distribution over valid profiles in auction a. We refer to $\{s_{\mathsf{a},p} | \mathsf{a} \in \mathsf{A}, p \in \mathcal{P}_{\mathsf{a}}\}$ as a profile-weight.

2. For any buyer $\mathsf{b} \in \mathsf{B}$ and reserve price $r \in \mathsf{R}$, define a variable $q_{\mathsf{b},r} \geq 0$ such that $\sum_{r \in \mathsf{R}} q_{\mathsf{b},r} = 1$. This variable represents be the probability that buyer b is assigned a reserve price of r.

We now discuss the LP constraints. We add constraints relating $s_{\mathsf{a},p}$ and $q_{\mathsf{b},r}$ which will ensure the consistency of probability distributions across all profiles. To define this set of constraints, for every $\mathsf{b} \in \mathsf{B}$, $\mathsf{a} \in \mathsf{A}$, and $r \in \mathsf{R}$, we define set

$$\mathcal{Q}_{\mathsf{b},r,\mathsf{a}} := \{p = (\mathsf{b}_1, \mathsf{b}, r_1, r) : p \in \mathcal{P}_{\mathsf{a}}\} \cup \{p = (\mathsf{b}, \mathsf{b}_2, r, r_2) : p \in \mathcal{P}_{\mathsf{a}}\}, \tag{2.1}$$

which corresponds to all valid profiles of auction a that assign reserve r to buyer b. A natural constraint to add is that the total probability assigned to profiles in $\mathcal{Q}_{\mathsf{b},r,\mathsf{a}}$ is at most the probability that buyer b is assigned to reserve price r. That is,

$$\sum_{p\in\mathcal{Q}_{\mathsf{b},r,\mathsf{a}}} s_{\mathsf{a},p} \leq q_{\mathsf{b},r}\,.$$

Finally, we can put it all together in the following LP:

$$\begin{aligned}
\max_{\mathbf{q},\mathbf{s}} \quad & \sum_{\mathsf{a}\in\mathsf{A}}\sum_{p\in\mathcal{P}_{\mathsf{a}}} s_{\mathsf{a},p}\cdot \mathsf{Rev}_{\mathsf{a}}(p) & & \\
\text{s.t.} \quad & \sum_{p\in\mathcal{P}_{\mathsf{a}}} s_{\mathsf{a},p} \leq 1 & & \forall \mathsf{a} : \mathsf{a}\in\mathsf{A} \\
& \sum_{p\in\mathcal{Q}_{b,r,\mathsf{a}}} s_{\mathsf{a},p} \leq q_{\mathsf{b},r} & & \forall \mathsf{a},\mathsf{b},r : \mathsf{b}\in\mathsf{B}, r\in\mathsf{R}, \mathsf{a}\in\mathsf{A} \\
& \sum_{r\in R} q_{\mathsf{b},r} = 1 & & \forall \mathsf{b} : \mathsf{b}\in\mathsf{B} \\
& s_{\mathsf{a},p} \geq 0 & & \forall \mathsf{a}, p : \mathsf{a}\in\mathsf{A},\ p\in\mathcal{P}_{\mathsf{a}}
\end{aligned} \qquad (\mathsf{Profile\text{-}LP})$$

We start by noting that the LP is a relaxation of the Problem (ESP-OPT):

Lemma 2.2.3 (Upper bound on Revenue)**.** *The solution of* Profile-LP *is an upper bound to* $\mathsf{ESP}^\star$*, i.e., the optimal value of Problem* ESP-OPT*. That is,*

$$\mathsf{ESP}^\star \leq \mathsf{Profile\text{-}LP}\,.$$

Proof. Given reserve prices $\mathbf{r}^\star$ such that $\mathsf{ESP}^\star = \sum_{\mathsf{a}} \mathsf{Rev}_{\mathsf{a}}(\mathbf{r}^\star)$, we construct a feasible solution to the LP as follows. For each $\mathsf{a}\in\mathsf{A}$, we let $s_{\mathsf{a},p} = 1$ for the profile p corresponding to $\mathbf{r}^\star$ (according to Definition 2.2.2) and $s_{\mathsf{a},p} = 0$ for all remaining profiles. Further, we let $q_{\mathsf{b},r^\star_{\mathsf{b}}} = 1$ and $q_{\mathsf{b},r} = 0$ for all remaining reserves. It is straightforward to verify that it is a feasible solution to the Profile-LP and that $\sum_{\mathsf{a}\in\mathsf{A}}\sum_{p\in\mathcal{P}_{\mathsf{a}}} s_{\mathsf{a},p}\cdot \mathsf{Rev}_{\mathsf{a}}(p) = \mathsf{ESP}^\star$. □

Theorem 2 (Integrality Gap of Profile-LP)**.** *There exists a data-set of bids* $\{\beta_{\mathsf{a},\mathsf{b}}\}_{\mathsf{a}\in\mathsf{A},\mathsf{b}\in\mathsf{B}}$ *for which the integrality gap of the LP is at least* 0.828*. That is,*

$$\mathsf{ESP}^\star \le 0.828\cdot(\text{Profile-LP})\,.$$

The proof of Theorem 2 is given in Section 2.5.

2.3 Profile-based LP-rounding (Pro-LPR) Algorithm

In this section, we present an algorithm, called Profile-based LP-rounding (Pro-LPR), that uses the optimal solution of (Profile-LP), $\mathbf{s}^\star$, to devise reserve prices. The algorithm is presented below.

Profile-based LP-rounding (Pro-LPR) Algorithm:

Let $\mathbf{s}^\star$ and $\mathbf{q}^\star$ be the optimal solution of (Profile-LP). Then,

- Rounding procedure: For each buyer $\mathsf{b}\in\mathsf{B}$, independently sample reserve price $r\in\mathsf{R}$ with probability proportional to $q^\star_{\mathsf{b},r}$.
- Let $\mathbf{z}$ be the vector of all zero reserves. Output the best of $\mathbf{r}^{\mathcal{R}}$ and $\mathbf{z}$, i.e.,

$$\mathbf{r}^{\text{out}} = \arg\max_{\mathbf{r}\in\{\mathbf{z},\mathbf{r}^{\mathcal{R}}\}} \mathsf{Rev}(\mathbf{r})\,.$$

In the $\mathsf{Pro\text{-}LPR}$ algorithm, we first round the optimal solution of the (Profile-LP) to construct reserve prices $\mathbf{r}^{\mathcal{R}}$. To do so, for each buyer $\mathsf{b} \in \mathsf{B}$, we independently sample reserve price $r \in \mathsf{R}$ with probability $q^{\star}_{\mathsf{b},r}$, where $\mathbf{q}^{\star}$ (and $\mathbf{s}^{\star}$) is the optimal solution of the (Profile-LP). We then compare revenue under $\mathbf{r}^{\mathcal{R}}$ with that under the zero reserve prices, and return the one that obtains higher revenue. The retuned vector of reserve prices is denoted by $\mathbf{r}^{\text{out}}$.

We now proceed to analyze our algorithm. We show that $\mathbb{E}[\mathsf{Rev}(\mathbf{r}^{\text{out}})]$ is at least a 0.684 fraction of the solution of the Profile-LP and hence at most $0.684 \cdot \mathsf{ESP}^{\star}$, where the expectation is with respect to the randomness in the $\mathsf{Pro\text{-}LPR}$ algorithm. As we show in Lemma 2.3.1, one of the biggest strengths of our LP formulation is that it allows the analysis to decouple the effect of rounding for each individual auction. In this lemma, roughly speaking, we present two conditions under which the $\mathsf{Pro\text{-}LPR}$ algorithm has a good performance. In these conditions, for each auction $\mathsf{a} \in \mathsf{A}$ and $t \geq 0$, we compare the probability that $\text{Rev}_{\mathsf{a}}(\mathbf{r}^{\mathcal{R}})$ is at least t, i.e., $\Pr[\text{Rev}_{\mathsf{a}}(\mathbf{r}^{\mathcal{R}}) \geq t]$, with $\sum_{\{p: p \in \mathcal{P}_{\mathsf{a}}, \text{Rev}(p) \geq t\}} s^{\star}_{\mathsf{a},p}$, which is the sum of the optimal weight of (valid) profiles in auction a that obtains a revenue of at least t. Here, $\text{Rev}_{\mathsf{a}}(\mathbf{r}^{\mathcal{R}})$ is the revenue in auction a under reserve prices $\mathbf{r}^{\mathcal{R}}$. Intuitively, the smaller the gap between $\Pr[\text{Rev}_{\mathsf{a}}(\mathbf{r}^{\mathcal{R}}) \geq t]$ and the aforementioned summation, the better the $\mathsf{Pro\text{-}LPR}$ algorithm performs. Lemma 2.3.1 makes this statement formal by considering the high revenue case of $t \geq \beta^{(2)}_{\mathsf{a}}$ (first condition) and the low revenue case of $t < \beta^{(2)}_{\mathsf{a}}$ (second condition), where $\beta^{(2)}_{\mathsf{a}}$ is the second highest bid in auction a. Note that when the reserve price for the buyer with the highest bid in auction a is set too high, revenue of this auction can be indeed less than the second highest submitted bid $\beta^{(2)}_{\mathsf{a}}$.

Lemma 2.3.1 (Two Conditions)**.** *Let* $\mathbf{s}^\star$ *and* $\mathbf{q}^\star$ *be the optimal solution of (*Profile-LP*) and* $\mathbf{r}^{\mathcal{R}}$ *be a random reserve price obtained from the rounding procedure. If there exists a constant* $c > 0$ *such that for any* $t \geq 0$ *and any auctions* $\mathsf{a} \in \mathsf{A}$*, we have*

$$\sum_{\{p : p \in \mathcal{P}_{\mathsf{a}}, \mathrm{Rev}(p) \geq t\}} s^\star_{\mathsf{a},p} - \Pr[\mathrm{Rev}_{\mathsf{a}}(\mathbf{r}^{\mathcal{R}}) \geq t] \;\leq\; 0 \quad \textit{for } t > \beta^{(2)}_{\mathsf{a}} \tag{2.2}$$

$$\sum_{\{p : p \in \mathcal{P}_{\mathsf{a}}, \mathrm{Rev}(p) \geq t\}} s^\star_{\mathsf{a},p} - \Pr[\mathrm{Rev}_{\mathsf{a}}(\mathbf{r}^{\mathcal{R}}) \geq t] \;\leq\; c \quad \textit{for } t \leq \beta^{(2)}_{\mathsf{a}}\,, \tag{2.3}$$

then Pro-LPR *algorithm is a* $(1+c)^{-1}$*-approximation. That is, it obtains at least* $(1+c)^{-1}$ *fraction of the optimal value of Problem* ESP-OPT*. Here,* $\beta^{(2)}_{\mathsf{a}}$ *is the second highest bid in auction* a *and* $\mathrm{Rev}_{\mathsf{a}}(\mathbf{r}^{\mathcal{R}})$ *is the revenue in auction* a *under reserve prices* $\mathbf{r}^{\mathcal{R}}$*.*

Proof. By integrating over t in Equations (2.2) and (3.5) and adding them up, we get

$$\int_{\beta^{(2)}_{\mathsf{a}}}^{\infty} \Big(\sum_{\{p : p \in \mathcal{P}_{\mathsf{a}}\, \mathrm{Rev}(p) \geq t\}} s^\star_{\mathsf{a},p} - \Pr[\mathrm{Rev}_{\mathsf{a}}(\mathbf{r}^{\mathcal{R}}) \geq t] \Big) dt + \int_{0}^{\beta^{(2)}_{\mathsf{a}}} \Big(\sum_{\{p : p \in \mathcal{P}_{\mathsf{a}}\, \mathrm{Rev}(p) \geq t\}} s^\star_{\mathsf{a},p} - \Pr[\mathrm{Rev}_{\mathsf{a}}(\mathbf{r}^{\mathcal{R}}) \geq t] \Big) dt \;\leq\; c \cdot \beta^{(2)}_{\mathsf{a}}\,.$$

This is simplified as follows

$$\sum_{p \in \mathcal{P}_{\mathsf{a}}} s^\star_{\mathsf{a},p}\, \mathrm{Rev}_{\mathsf{a}}(p) - \mathbb{E}[\mathrm{Rev}_{\mathsf{a}}(\mathbf{r}^{\mathcal{R}})] \leq c \cdot \beta^{(2)}_{\mathsf{a}}\,. \tag{2.4}$$

Note that by Lemma 2.2.3, the optimal value of Problem ESP-OPT, denoted by $\mathsf{ESP}^\star$, is upper bounded by $\mathsf{Rev}(\mathbf{s}^\star)$. That is,

$$\mathsf{ESP}^\star \leq \mathsf{Rev}(\mathbf{s}^\star) = \sum_{\mathsf{a}\in\mathsf{A}} \sum_{p\in\mathcal{P}_\mathsf{a}} s^\star_{\mathsf{a},p} \,\mathrm{Rev}(p)\,. \tag{2.5}$$

Further, the revenue of $\mathsf{Pro\text{-}LPR}$ algorithm, i.e., $\mathbb{E}[\mathrm{Rev}(\mathbf{r}^{\mathrm{out}})]$, is lower bounded by

$$\mathbb{E}[\mathrm{Rev}(\mathbf{r}^{\mathrm{out}})] \geq \max\Big(\sum_{\mathsf{a}\in A} \beta^{(2)}_\mathsf{a}, \mathbb{E}[\mathrm{Rev}(\mathbf{r}^{\mathcal{R}})]\Big)\,. \tag{2.6}$$

To see why this holds note that $\mathsf{Pro\text{-}LPR}$ algorithm returns the best of reserve price $\mathbf{r}^{\mathcal{R}}$ and all zero prices, where the revenue under all zero prices is the sum of the second highest highest bids $\sum_{\mathsf{a}\in A} \beta^{(2)}_\mathsf{a}$. By using Equations (2.4), (2.5), and (2.6), we have

$$\begin{aligned}\mathsf{ESP}^\star - \mathbb{E}[\mathrm{Rev}(\mathbf{r}^{\mathrm{out}})] &\leq c\sum_{\mathsf{a}\in\mathsf{A}} \beta^{(2)}_\mathsf{a} \\ &\leq c\mathbb{E}[\mathrm{Rev}(\mathbf{r}^{\mathrm{out}})]\,. \end{aligned}\tag{2.7}$$

Putting these together, we have

$$\mathbb{E}[\mathrm{Rev}(\mathbf{r}^{\mathrm{out}})] \geq \frac{1}{1+c}\cdot \mathsf{ESP}^\star\,,$$

which is the desired result. □

In the next lemma, we show that the first condition holds. We then dedicate the next section to identifying constant c in the second condition. The proof of the lemma is based on the observations that (i) revenue of any valid profile $(\mathsf{b}_1, \mathsf{b}_2, r_1, r_2)$

is greater than $\beta_{\mathsf{a}}^{(2)}$ if buyer $\mathsf{b}_1 = \mathsf{b}_{\mathsf{a}}^{(1)}$ and his reserve $r_1 > \beta_{\mathsf{a}}^{(2)}$, and (ii) revenue of auction a under reserve prices $\mathbf{r}^{\mathcal{R}}$ is greater than $\beta_{\mathsf{a}}^{(2)}$ if the reserve price of buyer $\mathsf{b}_{\mathsf{a}}^{(1)}$ is less than or equal to his bid and greater than the second highest bid $\beta_{\mathsf{a}}^{(2)}$. Here, buyer $\mathsf{b}_{\mathsf{a}}^{(1)}$ is the buyer with the highest bid in auction a.

Lemma 2.3.2 (First Condition Holds)**.** *Let* $\mathbf{s}^\star$ *denote an optimal solution of* Profile-LP *and* $\mathbf{r}^{\mathcal{R}}$ *be a random reserve price obtained from the rounding procedure in the* Pro-LPR *algorithm. For any auction* $\mathsf{a} \in A$*, we have*

$$\sum_{\{p: p\in\mathcal{P}_{\mathsf{a}}, \mathrm{Rev}(p)\geq t\}} s^\star_{\mathsf{a},p} - \Pr[\mathrm{Rev}_{\mathsf{a}}(\mathbf{r}^{\mathcal{R}}) \geq t] \;\leq\; 0 \qquad \textit{for } t > \beta_{\mathsf{a}}^{(2)}. \tag{2.8}$$

Proof. The first term in the l.h.s. of (2.8) can be written as

$$\sum_{\{p: p\in\mathcal{P}_{\mathsf{a}}, \mathrm{Rev}(p)\geq t\}} s^\star_{\mathsf{a},p} = \sum_{\{p: p\in\mathcal{P}_{\mathsf{a}}, p=(\mathsf{b}_{\mathsf{a}}^{(1)}, \mathsf{b}_2, r, r_2),\; r\geq t\}} s^\star_{\mathsf{a},p} \;\leq\; \sum_{r\geq t} q_{\mathsf{b}_{\mathsf{a}}^{(1)}, r} = \Pr[\mathrm{Rev}_{\mathsf{a}}(\mathbf{r}^{\mathcal{R}}) \geq t]\,, \tag{2.9}$$

where the first equation holds because revenue of a profile $p \in \mathcal{P}_{\mathsf{a}}$ is $t > \beta_{\mathsf{a}}^{(2)}$ if and only if the bidder with the highest bid in auction a, i.e., $\mathsf{b}_{\mathsf{a}}^{(1)}$, is assigned a reserve price $t > \beta_{\mathsf{a}}^{(2)}$ and the bid of this bidder is greater than t. The second equation holds because of the second set of constraints of (Profile-LP). The last equation follows from the construction of reserve prices $\mathbf{r}^{\mathcal{R}}$. Note that Equation (2.9) verifies condition (2.8). □

2.3.1 Bounding the Constant in the Second Condition

We start by noting that the second condition in Lemma 2.3.1 holds trivially for $c = 1$, which recovers the same approximation factor of 1/2 of [38]. For the rest of the section, we will improve past 1/2 by constructing a non-linear mathematical program to optimize c and then applying the first order conditions in non-linear programming to bound the optimal solution. In Lemma 2.3.3, we show that

$$c = \max_{\theta \in [0,1]} \mathsf{OPT}(\theta)\,,$$

where for any real number $\theta \in [0, 1]$, $\mathsf{OPT}(\theta)$ is defined as follows

$$\begin{aligned} \mathsf{OPT}(\theta) \;=\; \max_{\mathbf{x} \geq 0} \quad & e^{\theta-1}\left[\prod_{i\in[n]}(1-x_i) + \sum_{i\in[n]} x_i \prod_{j\in[n], j\neq i}(1-x_j)\right] \\ \text{s.t.} \quad & \frac{1}{2}\sum_{i\in[n]} x_i = \theta \\ & x_i \leq \theta, \quad \forall i \in [n]\,. \end{aligned} \tag{2.10}$$

Here, n is the number of buyers. Characterizing $\mathsf{OPT}(\theta)$ is technically involved and because of that its details is postponed to Section 2.3.3. There, we show that for any number of buyers $n \geq 2$ and any real number $\theta \in [0, 1]$,

$$\mathsf{OPT}(\theta) \leq 2\left(\sqrt{2}-1\right)e^{\sqrt{2}-2} \approx 0.4612.$$

Then, invoking Lemmas 2.3.1 and 2.3.2, this leads to the approximation factor of

$\frac{1}{1+0.4612} \approx 0.6844$, which is the desired result.

In the next lemma, we formally state the relationship between $\mathsf{OPT}(\theta)$ and the approximation factor of our algorithm.

Lemma 2.3.3 (Second Condition). *Let* $\mathbf{s}^\star$ *denote an optimal solution of* Profile-LP *and* $\mathbf{r}^{\mathcal{R}}$ *be a random reserve price obtained from the rounding procedure in the* Pro-LPR *algorithm. Let*

$$c = \max_{\theta \in [0,1]} \mathsf{OPT}(\theta)\,.$$

Then, for any auction $\mathsf{a} \in A$*, the following equation holds.*

$$\sum_{\{p : p \in \mathcal{P}_{\mathsf{a}}, \mathrm{Rev}(p) \geq t\}} s^\star_{\mathsf{a},p} - \Pr[\mathrm{Rev}_{\mathsf{a}}(\mathbf{r}^{\mathcal{R}}) \geq t] \;\leq\; c \qquad \textit{for } t \leq \beta^{(2)}_{\mathsf{a}}\,.$$

The formal proof of Lemma 2.3.3 due to being lengthy is deferred to Subsection 2.3.2, however we provide some intuition here. For each buyer b, we consider two disjoint subsets of valid profiles $(\mathsf{b}_1, \mathsf{b}_2, r_1, r_2)$ such that (i) revenue of any profile in these subsets is greater than or equal to t, where $t \leq \beta^{(2)}_{\mathsf{a}}$, and (ii) either b_1 or b_2 is equal to buyer b. The factor $1/2$ in the constraint of Problem (2.10) is the artifact of the definition of the subsets and how the summation $\sum_{\{p : p \in \mathcal{P}_{\mathsf{a}}, \mathrm{Rev}(p) \geq t\}} s^\star_{\mathsf{a},p}$ can be written as a function of the optimal weight of the profiles in these subsets; see Equation (2.12) in the proof. We then express $\Pr[\mathrm{Rev}_{\mathsf{a}}(\mathbf{r}^{\mathcal{R}}) \geq t]$ as the probability of the union of two events, where the first event happens if there is at least one cleared buyer with reserve price greater than t, and the second event happens if there are at least two buyers with cleared bids of at least t. We then write this probability as a function of the profile weights in the subsets by taking advantage of the fact that in our rounding procedure, reserve prices are independent across buyers. In particular, we show that this probability is at least one minus the left hand side of the equation in Lemma 2.3.4, stated below. We invoke Lemma 2.3.4 to complete the proof. Observe that the objective function of Problem (2.10) bears significant resemblance to that of the right hand of the equation in Lemma 2.3.4.

Lemma 2.3.4. *Consider a set* $\hat{\mathsf{B}} \subseteq \mathsf{B}$ *with* $|\hat{\mathsf{B}}| \geq 2$.[1] *Given fixed* $x_{1,\mathsf{b}}, x_{2,\mathsf{b}}$ *with* $\mathsf{b} \in \hat{\mathsf{B}}$ *and* $x_{1,\mathsf{b}} + x_{2,\mathsf{b}} \leq 1$, *the following inequality holds:*

$$\begin{aligned}
&\prod_{\mathsf{b}\in\hat{\mathsf{B}}}(1 - x_{1,\mathsf{b}} - x_{2,\mathsf{b}}) + \sum_{\mathsf{b}\in\hat{\mathsf{B}}} x_{2,\mathsf{b}} \prod_{\mathsf{b}'\neq\mathsf{b}}(1 - x_{1,\mathsf{b}'} - x_{2,\mathsf{b}'}) \leq \\
&\prod_{\mathsf{b}\in\hat{\mathsf{B}}}(1 - x_{1,\mathsf{b}}) \left[\prod_{\mathsf{b}\in\hat{\mathsf{B}}}(1 - x_{2,\mathsf{b}}) + \sum_{\mathsf{b}\in\hat{\mathsf{B}}} x_{2,\mathsf{b}} \prod_{\mathsf{b}'\neq\mathsf{b}}(1 - x_{2,\mathsf{b}'})\right].
\end{aligned} \tag{2.11}$$

Proof. Given a partition of $\hat{\mathsf{B}}$ in two sets $\mathsf{B}_1, \mathsf{B}_2$, define the following function:

$$\begin{aligned}
\Phi(\mathsf{B}_1, \mathsf{B}_2) = &\prod_{\mathsf{b}\in\mathsf{B}_1}(1 - x_{1,\mathsf{b}})(1 - x_{2,\mathsf{b}}) \prod_{\mathsf{b}\in\mathsf{B}_2}(1 - x_{1,\mathsf{b}} - x_{2,\mathsf{b}}) + \\
&\sum_{\mathsf{b}\in\mathsf{B}_1\cup\mathsf{B}_2} x_{2,\mathsf{b}} \left[\prod_{\mathsf{b}'\in\mathsf{B}_1,\mathsf{b}'\neq\mathsf{b}}(1 - x_{1,\mathsf{b}})(1 - x_{2,\mathsf{b}}) \prod_{\mathsf{b}'\in\mathsf{B}_2,\mathsf{b}'\neq\mathsf{b}}(1 - x_{1,\mathsf{b}} - x_{2,\mathsf{b}})\right].
\end{aligned}$$

The main claim in the lemma is that $\Phi(\mathsf{B}, \emptyset) \geq \Phi(\emptyset, \mathsf{B})$. We will show that for any $\mathsf{B}_1, \mathsf{B}_2$ and $\hat{\mathsf{b}} \in \mathsf{B}_2$, we have

$$\Phi(\mathsf{B}_1, \mathsf{B}_2) \leq \Phi(\mathsf{B}_1 \cup \{\hat{\mathsf{b}}\}, \mathsf{B}_2 \setminus \{\hat{\mathsf{b}}\})$$

[1] Note that because of the auxiliary buyers, $|\mathsf{B}| \geq 2$.

and the claim will follow by moving the elements from B_2 to B_1 one by one. To simplify notation, define

$$w = \prod_{\mathsf{b}\in\mathsf{B}_1}(1-x_{1,\mathsf{b}})(1-x_{2,\mathsf{b}})\prod_{\mathsf{b}\in\mathsf{B}_2\setminus\{\hat{\mathsf{b}}\}}(1-x_{1,\mathsf{b}}-x_{2,\mathsf{b}})\,.$$

Now we can write:

$$\begin{aligned}\Phi(\mathsf{B}_1,\mathsf{B}_2) =& w\cdot(1-x_{1,\hat{\mathsf{b}}}-x_{2,\hat{\mathsf{b}}}) + w\cdot x_{2,\hat{\mathsf{b}}}\\ &+\sum_{\mathsf{b}\in\mathsf{B}_2;\mathsf{b}\neq\hat{\mathsf{b}}} w\cdot\frac{1-x_{1,\hat{\mathsf{b}}}-x_{2,\hat{\mathsf{b}}}}{1-x_{1,\mathsf{b}}-x_{2,\mathsf{b}}}\cdot x_{2,\mathsf{b}} + \sum_{\mathsf{b}\in\mathsf{B}_1} w\cdot\frac{1-x_{1,\hat{\mathsf{b}}}-x_{2,\hat{\mathsf{b}}}}{(1-x_{1,\mathsf{b}})(1-x_{2,\mathsf{b}})}\cdot x_{2,\mathsf{b}},\end{aligned}$$

and

$$\begin{aligned}\Phi(\mathsf{B}_1\cup\{\hat{\mathsf{b}}\},\mathsf{B}_2\setminus\{\hat{\mathsf{b}}\}) &= w\cdot(1-x_{1,\hat{\mathsf{b}}})(1-x_{2,\hat{\mathsf{b}}}) + w\cdot x_{2,\hat{\mathsf{b}}}\\ &\quad+\sum_{\mathsf{b}\in\mathsf{B}_2;\mathsf{b}\neq\hat{\mathsf{b}}} w\cdot\frac{(1-x_{1,\hat{\mathsf{b}}})(1-x_{2,\hat{\mathsf{b}}})}{1-x_{1,\mathsf{b}}-x_{2,\mathsf{b}}}\cdot x_{2,\mathsf{b}}\\ &\quad+\sum_{\mathsf{b}\in\mathsf{B}_1} w\cdot\frac{(1-x_{1,\hat{\mathsf{b}}})(1-x_{2,\hat{\mathsf{b}}})}{(1-x_{1,\mathsf{b}})(1-x_{2,\mathsf{b}})}\cdot x_{2,\mathsf{b}}.\end{aligned}$$

Our goal here is to show $\Phi(\mathsf{B}_1,\mathsf{B}_2)\le\Phi(\mathsf{B}_1\cup\{\hat{\mathsf{b}}\},\mathsf{B}_2\setminus\{\hat{\mathsf{b}}\})$. We start with comparing the first two terms of $\Phi(\mathsf{B}_1,\mathsf{B}_2)$ and $\Phi(\mathsf{B}_1\cup\{\hat{\mathsf{b}}\},\mathsf{B}_2\setminus\{\hat{\mathsf{b}}\})$:

$$\begin{aligned}w\cdot(1-x_{1,\hat{\mathsf{b}}}-x_{2,\hat{\mathsf{b}}}) + w\cdot x_{2,\hat{\mathsf{b}}} &= w\cdot(1-x_{1,\hat{\mathsf{b}}}) \le w\cdot(1-x_{1,\hat{\mathsf{b}}}+x_{1,\hat{\mathsf{b}}}x_{2,\hat{\mathsf{b}}})\\ &= w\cdot(1-x_{1,\hat{\mathsf{b}}})(1-x_{2,\hat{\mathsf{b}}}) + w\cdot x_{2,\hat{\mathsf{b}}}\,.\end{aligned}$$

We can compare the remaining terms one by one using the fact that:

$$1 - x_{1,\hat{b}} - x_{2,\hat{\mathsf{b}}} \leq (1 - x_{1,\hat{\mathsf{b}}})(1 - x_{2,\hat{\mathsf{b}}})\,.$$

This concludes that $\Phi(\mathsf{B}_1, \mathsf{B}_2) \leq \Phi(\mathsf{B}_1 \cup \{\hat{\mathsf{b}}\}, \mathsf{B}_2 \setminus \{\hat{\mathsf{b}}\})$ as desired. □

2.3.2 Proof of Lemma 2.3.3

We start with a few definitions. Consider a certain auction $\mathsf{a} \in \mathsf{A}$ and all of its valid profiles $p \in \mathcal{P}_{\mathsf{a}}$. Fix some threshold $t \leq \beta_{\mathsf{a}}^{(2)}$ and an optimal solution of (Profile-LP), denoted by $\mathbf{s}^{\star}$. Let set $\mathsf{B}_{\mathsf{a},t}$ be the set of buyers whose bid in auction a is at least t:

$$\mathsf{B}_{\mathsf{a},t} := \{\mathsf{b} \in \mathsf{B}; \beta_{\mathsf{a},\mathsf{b}} \geq t\}\,.$$

Note that this set is not empty because $t \leq \beta_{\mathsf{a}}^{(2)}$. In fact, $|\mathsf{B}_{\mathsf{a},t}| \geq 2$, as buyers with the highest and second-highest bids belong to this set. (Recall that because of the auxiliary buyers, $|\mathsf{B}| \geq 2$.) A crucial observation is that the reserve assigned to any buyer $\mathsf{b} \notin \mathsf{B}_{\mathsf{a},t}$ does not affect the event $\mathrm{Rev}_{\mathsf{a}}(\mathbf{r}^{\mathcal{R}}) \geq t$ since such buyers can be neither the winner nor the price setter in an auction with revenue of at least t. Consider a buyer $\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}$. Then, define

$$\mathcal{X}'_{1,\mathsf{b}} = \{p = (\mathsf{b}, \mathsf{b}_2, r_1, r_2) : p \in \mathcal{P}_{\mathsf{a}},\ r_1 \geq t\},$$

$$\mathcal{X}''_{1,\mathsf{b}} = \{p = (\mathsf{b}_1, \mathsf{b}, r_1, r_2) : p \in \mathcal{P}_{\mathsf{a}},\ r_1 < t \text{ and } r_2 \geq t\},$$

$$\mathcal{X}_{2,\mathsf{b}} = \{p = (\mathsf{b}_1, \mathsf{b}_2, r_1, r_2) : p \in \mathcal{P}_{\mathsf{a}}, \mathsf{b} \in \{\mathsf{b}_1, \mathsf{b}_2\}, r_1, r_2 < t \text{ and } \beta_{\mathsf{a},\mathsf{b}_2} \geq t\},$$

and set

$$x_{1,\mathsf{b}} = \sum_{p \in \mathcal{X}'_{1,\mathsf{b}} \cup X''_{1,\mathsf{b}}} s^\star_{\mathsf{a},p} \quad \text{and} \quad x_{2,\mathsf{b}} = \sum_{p \in \mathcal{X}_{2,\mathsf{b}}} s^\star_{\mathsf{a},p} \,.$$

We note that $\mathcal{X}'_{1,\mathsf{b}}$ is the set of all valid profiles $p = (\mathsf{b}, \mathsf{b}_2, r_1, r_2)$ in which reserve of buyer b is at least t. $\mathcal{X}''_{1,\mathsf{b}}$ is the set of all valid profiles $p = (\mathsf{b}_1, \mathsf{b}, r_1, r_2)$ in which reserve of buyer b_1 is less than t and reserve of buyer b is greater than or equal to t. Observe that for all the profiles p in $\mathcal{X}'_{1,\mathsf{b}} \cup \mathcal{X}''_{1,\mathsf{b}}$, reserve of buyer b is at least t. This implies that for all of these profiles, $\mathrm{Rev}(p) \geq t$. We also note that $\mathcal{X}_{2,\mathsf{b}}$ is the set of all valid profiles $p = (\mathsf{b}_1, \mathsf{b}_2, r_1, r_2)$ such that buyer $\mathsf{b} \in \{\mathsf{b}_1, \mathsf{b}_2\}$ and bid of buyer b_2 is at least t. Again, it is easy to see that for any valid profile $p \in \mathcal{X}_{2,\mathsf{b}}$, we have $\mathrm{Rev}(p) \geq t$. Finally, we point that while any profile p in $\mathcal{X}_{2,\mathsf{b}}$ and $\mathcal{X}'_{1,\mathsf{b}} \cup \mathcal{X}''_{1,\mathsf{b}}$ has $\mathrm{Rev}(p) \geq t$, by construction, $\mathcal{X}_{2,\mathsf{b}}$ and $\mathcal{X}'_{1,\mathsf{b}} \cup \mathcal{X}''_{1,\mathsf{b}}$ are disjoint. Therefore, we have

$$\sum_{\{p:\ p \in \mathcal{P}_{\mathsf{a}},\ \mathrm{Rev}(p) \geq t\}} s^\star_{\mathsf{a},p} \;=\; \sum_{\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}} x_{1,\mathsf{b}} + \frac{1}{2} \sum_{\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}} x_{2,\mathsf{b}} \,, \tag{2.12}$$

where the coefficient $\frac{1}{2}$ accounts for double-counting. Define $y_{1,\mathsf{b}}$ as the probability that the sampled reserve of buyer b, i.e., $\mathbf{r}^{\mathcal{R}}_{\mathsf{b}}$, is in $[t, \beta_{\mathsf{a},\mathsf{b}}]$ and $y_{2,\mathsf{b}}$ as the probability that the sampled reserve $\mathbf{r}^{\mathcal{R}}_{\mathsf{b}}$ is in $[0, t)$. By the sampling procedure we know that:

$$y_{1,\mathsf{b}} \geq x_{1,\mathsf{b}} \quad \text{and} \quad y_{2,\mathsf{b}} \geq x_{2,\mathsf{b}} \,. \tag{2.13}$$

Observe that $\mathrm{Rev}_{\mathsf{a}}(\mathbf{r}^{\mathcal{R}}) \geq t$ iff at least one of the two following events happen.

Event $\mathcal{E}_1$: There exists a buyer $\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}$ with a reserve of at least t whose bid is cleared.

Event $\mathcal{E}_2$: There are at least two buyers $\mathsf{b}_1, \mathsf{b}_2 \in \mathsf{B}_{\mathsf{a},t}$ with cleared bids of at least t.

Precisely,

$$\Pr[\mathrm{Rev}_{\mathsf{a}}(\mathbf{r}^{\mathcal{R}}) \geq t] \;=\; \Pr[\mathcal{E}_1 \text{ or } \mathcal{E}_2] \;=\; \Pr[\mathcal{E}_1] + \Pr[\mathcal{E}_2 \text{ and } \bar{\mathcal{E}}_1] \;=\; \Pr[\mathcal{E}_1] + \Pr[\bar{\mathcal{E}}_1]\Pr[\mathcal{E}_2|\bar{\mathcal{E}}_1]\,, \tag{2.14}$$

where

$$\Pr[\mathcal{E}_1] \;=\; 1 - \prod_{\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}} (1 - y_{1,\mathsf{b}}),$$

and

$$\Pr[\mathcal{E}_2|\bar{\mathcal{E}}_1] \;=\; 1 - \prod_{\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}} (1 - \tilde{y}_{2,\mathsf{b}}) - \sum_{\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}} \tilde{y}_{2,\mathsf{b}} \prod_{\mathsf{b}' \neq \mathsf{b}} (1 - \tilde{y}_{2,\mathsf{b}'}) \quad \text{with} \quad \tilde{y}_{2,\mathsf{b}} \;=\; \frac{y_{2,\mathsf{b}}}{1 - y_{1,\mathsf{b}}}\,.$$

This gives us

$$\begin{aligned} \Pr[\mathcal{E}_2 \text{ and } \bar{\mathcal{E}}_1] \;&=\; \Pr[\bar{\mathcal{E}}_1]\Pr[\mathcal{E}_2|\bar{\mathcal{E}}_1] \\ &=\; \Pr[\bar{\mathcal{E}}_1] - \prod_{\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}} (1 - y_{1,\mathsf{b}} - y_{2,\mathsf{b}}) - \sum_{\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}} y_{2,\mathsf{b}} \prod_{\mathsf{b}' \neq \mathsf{b}} (1 - y_{1,\mathsf{b}'} - y_{2,\mathsf{b}'})\,. \end{aligned}$$

Thus, by Equation (2.14), we get

$$\Pr[\mathcal{E}_2 \text{ or } \mathcal{E}_1] \;=\; 1 - \prod_{\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}} (1 - y_{1,\mathsf{b}} - y_{2,\mathsf{b}}) - \sum_{\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}} y_{2,\mathsf{b}} \prod_{\mathsf{b}' \neq \mathsf{b}} (1 - y_{1,\mathsf{b}'} - y_{2,\mathsf{b}'})\,.$$

Now observe that the expression above, i.e., $\Pr[\mathcal{E}_2 \text{ or } \mathcal{E}_1]$, is increasing in both $y_{1,\mathsf{b}}$ and $y_{2,\mathsf{b}}$, $\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}$. To see why $\Pr[\mathcal{E}_2 \text{ or } \mathcal{E}_1]$ is increasing in $y_{2,\mathsf{b}}$, note that

$$\frac{\partial(\Pr[\mathcal{E}_2 \text{ or } \mathcal{E}_1])}{\partial y_{2,\mathsf{b}}} = \sum_{\mathsf{b}' \in \mathsf{B}_{\mathsf{a},t}, \mathsf{b}' \neq \mathsf{b}} y_{2,\mathsf{b}'} \prod_{\mathsf{b}'' \neq \mathsf{b},\mathsf{b}'} (1 - y_{1,\mathsf{b}''} - y_{2,\mathsf{b}''}) \geq 0\,.$$

This and Equation (2.13) imply that:

$$\Pr[\mathcal{E}_2 \text{ or } \mathcal{E}_1] \;\geq\; 1 - \prod_{\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}} (1 - x_{1,\mathsf{b}} - x_{2,\mathsf{b}}) - \sum_{\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}} x_{2,\mathsf{b}} \prod_{\mathsf{b}' \neq \mathsf{b}} (1 - x_{1,\mathsf{b}'} - x_{2,\mathsf{b}'})\,.$$

We now invoke Lemma 2.3.4, stated earlier, to get

$$\Pr[\mathcal{E}_2 \text{ or } \mathcal{E}_1] \;\geq\; 1 - \prod_{\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}} (1 - x_{1,\mathsf{b}}) \left[\prod_{\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}} (1 - x_{2,\mathsf{b}}) + \sum_{\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}} x_{2,\mathsf{b}} \prod_{\mathsf{b}' \neq \mathsf{b}} (1 - x_{2,\mathsf{b}'}) \right]. \tag{2.15}$$

Using Equations (2.12), (2.15), and (2.14), we have

$$\begin{aligned} \sum_{\{p: p \in \mathcal{P}_{\mathsf{a}}, \mathrm{Rev}(p) \geq t\}} s^{\star}_{\mathsf{a},p} - \Pr[\mathrm{Rev}_{\mathsf{a}}(\mathbf{r}^{\mathcal{R}}) \geq t] \;\leq\; & \sum_{\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}} x_{1,\mathsf{b}} + \frac{1}{2} \sum_{\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}} x_{2,\mathsf{b}} - \\ \left(1 - \prod_{\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}} (1 - x_{1,\mathsf{b}}) \left[\prod_{\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}} (1 - x_{2,\mathsf{b}}) + \sum_{\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}} x_{2,\mathsf{b}} \prod_{\mathsf{b}' \neq \mathsf{b}} (1 - x_{2,\mathsf{b}'}) \right] \right). \end{aligned} \tag{2.16}$$

We claim that for any $\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}$, the above expression is non-decreasing in $x_{1,\mathsf{b}}$. To get this, we need to show that the derivative of the above expression w.r.t. $x_{1,\hat{\mathsf{b}}}$ is non-negative. In the other words, we need to show the following equation holds:

$$1 - \prod_{\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}, \mathsf{b} \neq \hat{\mathsf{b}}} (1 - x_{1,\mathsf{b}}) \left[\prod_{\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}} (1 - x_{2,\mathsf{b}}) + \sum_{\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}} x_{2,\mathsf{b}} \prod_{\mathsf{b}' \neq \mathsf{b}} (1 - x_{2,\mathsf{b}'}) \right] \geq 0. \tag{2.17}$$

Since $0 \leq (1-x_{1,\mathsf{b}}) \leq 1$ for any b, it only remains to show that the value of the term in the brackets, i.e., $\prod_{\mathsf{b}\in\mathsf{B}_{\mathsf{a},t}}(1-x_{2,\mathsf{b}})+\sum_{\mathsf{b}\in\mathsf{B}_{\mathsf{a},t}} x_{2,\mathsf{b}} \prod_{\mathsf{b}'\neq\mathsf{b}}(1-x_{2,\mathsf{b}'})$, is always in the range of $[0,1]$. To get this, it suffices to show that there exists an event whose probability can be written as $\prod_{\mathsf{b}\in\mathsf{B}_{\mathsf{a},t}}(1-x_{2,\mathsf{b}})+\sum_{\mathsf{b}\in\mathsf{B}_{\mathsf{a},t}} x_{2,\mathsf{b}} \prod_{\mathsf{b}'\neq\mathsf{b}}(1-x_{2,\mathsf{b}'})$. For any b define a Bernoulli random variable with mean $x_{2,\mathsf{b}}$. Observe that the aforementioned term is equal to the probability of the event in which at most one of these variables is equal to one, assuming that they are independent. Thus, we obtain Equation (2.17), which allows us to assume without loss of generality that $\sum_{\mathsf{b}\in\mathsf{B}_{\mathsf{a},t}} x_{1,\mathsf{b}} + \frac{1}{2}\sum_{\mathsf{b}\in\mathsf{B}_{\mathsf{a},t}} x_{2,\mathsf{b}}= 1$. As a result, we have

$$\sum_{\{p:p\in\mathcal{P}_\mathsf{a},\mathrm{Rev}(p)\geq t\}} s^\star_{\mathsf{a},p} - \Pr[\mathrm{Rev}_\mathsf{a}(\mathbf{r}^\mathcal{R}) \geq t] \leq$$
$$\prod_{\mathsf{b}\in\mathsf{B}_{\mathsf{a},t}}(1-x_{1,\mathsf{b}})\left[\prod_{\mathsf{b}\in\mathsf{B}_{\mathsf{a},t}}(1-x_{2,\mathsf{b}})+\sum_{\mathsf{b}\in\mathsf{B}_{\mathsf{a},t}} x_{2,\mathsf{b}} \prod_{\mathsf{b}'\neq\mathsf{b}}(1-x_{2,\mathsf{b}'})\right],$$

where $\sum_{\mathsf{b}\in\mathsf{B}_{\mathsf{a},t}} x_{2,\mathsf{b}} = 2\theta$, $\sum_{\mathsf{b}\in\mathsf{B}_{\mathsf{a},t}} x_{1,\mathsf{b}} = 1-\theta$. Here, $\theta \in [0,1]$. To complete the proof, we simply use that: $\prod_{\mathsf{b}\in\mathsf{B}_{\mathsf{a},t}}(1-x_{1,\mathsf{b}}) \leq e^{-\sum_{b\in\mathsf{B}_{\mathsf{a},t}} x_{1,\mathsf{b}}} = e^{\theta-1}$. Given how we constructed the variables $x_{2,\mathsf{b}}$, we also need $x_{2,\mathsf{b}} \leq \theta$. Hence,

$$\sum_{\{p:p\in\mathcal{P}_\mathsf{a},\mathrm{Rev}(p)\geq t\}} s^\star_{\mathsf{a},p} - \Pr[\mathrm{Rev}_\mathsf{a}(\mathbf{r}^\mathcal{R}) \geq t] \;\leq\; e^{\theta-1}\left[\prod_{\mathsf{b}\in\mathsf{B}_{\mathsf{a},t}}(1-x_{2,\mathsf{b}})+\sum_{\mathsf{b}\in\mathsf{B}_{\mathsf{a},t}} x_{2,\mathsf{b}} \prod_{\mathsf{b}'\neq\mathsf{b}}(1-x_{2,\mathsf{b}'})\right],$$

where $\sum_{\mathsf{b}\in\mathsf{B}_{\mathsf{a},t}} x_{2,\mathsf{b}} = 2\theta$ and $x_{2,\mathsf{b}} \leq \theta$ for any $\mathsf{b} \in \mathsf{B}_{\mathsf{a},t}$.

2.3.3 Approximation Factor

In this section, we will show that for any given $\theta \in [0, 1]$, we have

$$\mathsf{OPT}(\theta) \leq 2\left(\sqrt{2} - 1\right)e^{\sqrt{2}-2},$$

where $\mathsf{OPT}(\theta)$ is defined in Equation (2.10). Since the constraints of Program (2.10) are linear in x_i's, the first order conditions of Karush-Kuhn-Tucker (KKT) are a necessary condition for optimality [13]. Let

$$F(x, \theta) = e^{\theta-1}\left[\prod_{i\in[n]}(1 - x_i) + \sum_{i\in[n]} x_i \prod_{j\in[n],j\neq i}(1 - x_j)\right].$$

Observe that $F(x, \theta)$ is the objective function of $\mathsf{OPT}(\theta)$. Then, according to the KKT conditions, the optimal solution must satisfy the following constraints for some $\lambda \in \mathbb{R}$, $\mu, \eta \in \mathbb{R}^n_+$:

$$\nabla_x F(x, \theta) + \frac{\lambda}{2}\mathbf{1} - \mu + \eta = 0 \tag{2.18}$$

$$\sum_{i\in[n]} x_i = \frac{1}{2}\theta \tag{2.19}$$

$$\mu_i(x_i - \theta) = 0, \quad \forall i \in [n] \tag{2.20}$$

$$\eta_i x_i = 0, \quad \forall i \in [n] \tag{2.21}$$

$$0 \leq x_i \leq \theta, \quad \forall i \in [n]. \tag{2.22}$$

where $\mathbf{1} \in \mathbb{R}^n$ is the vector of all one.

It is enough to show that $F(x, \theta) \leq 2\big(\sqrt{2} - 1\big)e^{\sqrt{2}-2}$ for any tuple $(x, \theta, \lambda, \mu, \eta)$ satisfying the KKT conditions. A simple consequence of the KKT condition is the following:

Lemma 2.3.5 (KKT Condition)**.** *If $(x, \theta, \lambda, \mu, \eta)$ satisfies the KKT conditions for Problem (2.10), then if x_k and x_t are such that $0 < x_k < \theta$ and $0 < x_t < \theta$, then $x_k = x_t$.*

Proof. By conditions (2.20) and (2.21), we must have $\mu_k = \eta_k = 0$. Plugging that into condition (2.18), we get that

$$\partial F/\partial x_k + \lambda/2 = 0\,.$$

This implies that

$$\sum_{i \neq k} x_i \prod_{j \neq i,k} (1 - x_j) + \frac{\lambda}{2} = 0\,.$$

Let $Q = \prod_{i \in [n]}(1 - x_i)$ and $S = \sum_{i \in [n]} \frac{x_i}{1-x_i}$. Then, the above condition can be written as

$$\frac{Q}{1 - x_k} \sum_{i \neq k} \frac{x_i}{1 - x_i} + \frac{\lambda}{2} = 0 \quad \Rightarrow \quad \frac{Q}{1 - x_k}(S - \frac{x_k}{1 - x_k}) + \frac{\lambda}{2} = 0\,.$$

This is further simplified as follows

$$\big(SQ + \frac{\lambda}{2}\big) - \big(SQ + Q + \lambda\big)x_k + \frac{\lambda}{2}x_k^2 = 0\,.$$

The polynomial $p(y) := \big(SQ + \frac{\lambda}{2}\big) - \big(SQ + Q + \lambda\big)y + \frac{\lambda}{2}y^2$ is quadratic with $\frac{d^2p}{d^2y} \geq 0$ and $p(1) = -Q < 0$. Thus, $p(y) = 0$ has an unique solution with $y < 1$. This implies x_k is uniquely determined as a function of S, Q, and λ. By the same argument, x_t is also a solution to the same equation and hence $x_k = x_t$. □

Lemma 2.3.5 leads to the following corollary.

Corollary 2.3.6. *We can bound* $\mathsf{OPT}(\theta) \leq \max_{k\in\mathbb{Z},k\geq 2} \max[\mathsf{OPT}^1(\theta,k), \mathsf{OPT}^2(\theta,k)]$, *where*

$$\mathsf{OPT}^1(\theta,k) = e^{\theta-1}\left(1 - \frac{2\theta}{k}\right)^{k-1}\left(1 - \frac{2\theta}{k} + 2\theta\right)$$

$$\mathsf{OPT}^2(\theta,k) = e^{\theta-1}\left[\left(1 - \frac{\theta}{k}\right)^{k} + \theta(1-\theta)\left(1 - \frac{\theta}{k}\right)^{k-1}\right].$$

Proof. As stated earlier, in order to maximize the objective function $\mathsf{OPT}(\theta)$, it is enough to consider feasible solutions x satisfying the KKT conditions. To do so, we use Lemma 2.3.5 to narrow down such solutions.

Since for any $i \in [n]$, $x_i \leq \theta$ and $\sum_{i\in[n]} x_i = 2\theta$, we an only have the following three cases:

- Case 1: Two variables in the support have value θ and by constraint $\sum_{i\in[n]} x_i = 2\theta$, the rest of them are zero. In that case, $\mathsf{OPT}(\theta) = \mathsf{OPT}^1(\theta, 2)$.

- Case 2: One variable has value θ and by Lemma 2.3.5, the rest $n-1 \geq 2$ variables in the support have value $\theta/(n-1)$. In that case, $\mathsf{OPT}(\theta) = \mathsf{OPT}^2(\theta, n-1)$.

- Case 3: All variables in the support are strictly below θ. In this case, by Lemma 2.3.5, $x_i = \theta/n$ for $n \geq 3$, and the solution is $\mathsf{OPT}(\theta) = \mathsf{OPT}^1(\theta, n)$.

$\square$

Lemma 2.3.7. *For any $\theta \in [0,1]$ and $k \geq 2$, we have $\mathsf{OPT}^1(\theta, k) \leq 2\big(\sqrt{2}-1\big)e^{\sqrt{2}-2}$.*

Proof. For each $k \geq 0$, define $\theta^*(k) = \arg\max_{\theta \in [0,1]} \mathsf{OPT}^1(\theta, k)$. By solving

$$\partial \mathsf{OPT}^1(\theta, k)/\partial\theta = 0$$

we obtain the following expression for $\theta^*(k)$:

$$k^2(2\theta^*(k) - 1) + 4(k-1)(\theta^*(k))^2 = 0\,.$$

The aforementioned equation has two solutions, only one of which is in $[0,1]$. Thus,

$$\theta^*(k) = \frac{k\big(k - \sqrt{k^2+4k-4}\big)}{4-4k}. \tag{2.23}$$

We need to show that for any $k \geq 2$, we have $\mathsf{OPT}^1(\theta^*(k), k) \leq 2\big(\sqrt{2}-1\big)e^{\sqrt{2}-2} \approx 0.461$. For $k = 2$, we have $\mathsf{OPT}^1(\theta^*(k), k) = 2\big(\sqrt{2}-1\big)e^{\sqrt{2}-2}$. For $k < 40$, we can verify this inequality numerically. For $k \geq 40$, we define and upper bound:

$$U(\theta, k) = \frac{2\theta+1}{e^{\theta+1}(1-\frac{2\theta}{k})}.$$

and show that for any $\theta \in [0,1]$ and $k \geq 40$,

$$\mathsf{OPT}^1(\theta, k) \leq U(\theta, k) \leq U(\theta, 40) \leq 0.459 < 2\Big(\sqrt{2}-1\Big)e^{\sqrt{2}-2}\,.$$

For the first inequality note that:

$$\mathsf{OPT}^1(\theta,k) = e^{\theta-1}\left[\left(1-\frac{2\theta}{k}\right)^{k-1}\left(1+(k-1)\frac{2\theta}{k}\right)\right] \tag{2.24}$$

$$< e^{\theta-1}\left[\left(1-\frac{2\theta}{k}\right)^{k}\left(1-\frac{2\theta}{k}\right)^{-1}(1+2\theta)\right] \le U(\theta,k). \tag{2.25}$$

For the second inequality, we use the fact that for any θ, $U(\theta,k)$ is decreasing in k. To find an upper-bound for value of $U(\theta,40) = \frac{(2\theta+1)}{e^{\theta+1}(1-\frac{\theta}{20})}$, we take derivative of that which is

$$\frac{\partial U(\theta,40)}{\partial\theta} = \frac{20(2\theta^2-39\theta+21)}{e^{\theta+1}(\theta-20)^2}\,.$$

By solving $\frac{\partial U(\theta,40)}{\partial\theta} = 0$, we obtain that maximum of $U(\theta,40)$ is at $\theta = \frac{1}{4}\big(39-\sqrt{1353}\big)$ and

$$U\left(\frac{1}{4}\Big(39-\sqrt{1353}\Big),40\right) < 0.459\,.$$

This completes the proof. □

Lemma 2.3.8. *For any $\theta \in [0,1]$ and $k \ge 2$, we have $\mathsf{OPT}^2(\theta,k) \le 0.46 < 2\big(\sqrt{2}-1\big)e^{\sqrt{2}-2}$.*

Proof. Observe that

$$e^{1-\theta}\mathsf{OPT}^2(\theta,k) = \left(1-\frac{\theta}{k}\right)^{k} + \theta(1-\theta)\left(1-\frac{\theta}{k}\right)^{k-1} \tag{2.26}$$

$$\le \left(1-\frac{\theta}{k}\right)^{k} + \frac{1}{4}\left(1-\frac{\theta}{k}\right)^{k} = \frac{5}{4}\left(1-\frac{\theta}{k}\right)^{k}, \tag{2.27}$$

where the first inequality holds because $\max_{\theta\in[0,1]}\theta(1-\theta) = \frac{1}{4}$ and $1-\frac{\theta}{k} \le 1$. Finally, note that $e^{\theta-1} \cdot \frac{5}{4}(1-\frac{\theta}{k})^k$ is decreasing for $\theta \in [0,1]$, Thus, we can bound $\mathsf{OPT}^2(\theta, k)$ by the value of $e^{\theta-1} \cdot \frac{5}{4}(1-\frac{\theta}{k})^k$ at $\theta = 0$ which is $5/(4e) < 0.46$. □

2.4 Tightness of the Analysis

In this section, we show that the analysis of our algorithm is tight, i.e., we construct an example for which the performance of the algorithm matches the 0.684 approximation factor.

To make the construction cleaner, we can define the weighted version of our problem in which each auction $\mathsf{a} \in \mathsf{A}$ has an associated weight $w_\mathsf{a} > 0$, and the objective is to maximize $\sum_{\mathsf{a}\in\mathsf{A}} w_\mathsf{a} \cdot \mathsf{Rev}_\mathsf{a}(\mathbf{r})$. Note that if the weights are integers, this is exactly the same as the original problem, replacing each weighted auction by w_a (unweighted) copies. Even if w_a's are not integers, it is easy to see that the algorithm and the analysis generalize with essentially no change to the weighted case (the only modification involves adding weighs to the objective function in the LP). In other words, if the objective were the weighted revenue, we would still get 0.684 approximation factor by applying a similar algorithm. Furthermore, any lower bound to the weighted case translates to the unweighted case by replacing a weighted auction a by $\lfloor N w_a \rfloor$ unweighted copies for some large N.

Theorem 3 (Tightness of the Analysis)**.** *There exist a weighted instance $\{w_\mathsf{a}\}_{\mathsf{a}\in\mathsf{A}}$, $\{\beta_{\mathsf{a},\mathsf{b}}\}_{\mathsf{a}\in\mathsf{A},\mathsf{b}\in\mathsf{B}}$ and an optimal LP solution $\mathbf{s}, \mathbf{q}$ such that*

$$\max\left(\mathbb{E}\left[\sum_{\mathsf{a}} w_\mathsf{a} \operatorname{Rev}_\mathsf{a}(\mathbf{r}^{\mathcal{R}})\right], \sum_{\mathsf{a}} w_\mathsf{a} \operatorname{Rev}_\mathsf{a}(\mathbf{0})\right) \leq 0.684 \cdot \operatorname{Rev}(\mathbf{s}).$$

Proof. Fix $\theta = \sqrt{2} - 1$ and $c = (1-\theta^2)e^{\theta-1}$. Consider an instance with three weighted auctions and $n = k+3$ buyers described by the following table:

Weights w_{a} \ Bids	$\beta_{\mathsf{a},1}$	$\dots$	$\beta_{\mathsf{a},k}$	$\beta_{\mathsf{a},k+1}$	$\beta_{\mathsf{a},k+2}$	$\beta_{\mathsf{a},k+3}$
$1/(c+1)$	1	$\dots$	1	1	1	0
$c/(c+1)$	0	$\dots$	0	0	0	$1+\varepsilon$
ε	$1+\varepsilon$	$\dots$	$1+\varepsilon$	$1+\varepsilon$	$1+\varepsilon$	0

Now, consider the following solution to the Profile-LP. For the first auction,

- the profile $p = (i, \mathsf{b}_0, 1, 0)$ has $s_{\mathsf{a},p} = (1-\theta)/k$ for $i \in [k]$. In this profile, the i-th buyer is reserve priced at 1 and the second buyer is the dummy buyer;
- the profile $p = (k+1, k+2, 0, 0)$ has weight $s_{\mathsf{a},p} = \theta$. In this profile, both buyers $k+1$ and $k+2$ have zero reserve prices. Observe that the revenue under this profile is 1 due to the highest second price.

For the second auction, we consider only one profile:

- the profile $p = (k+3, \mathsf{b}_0, 1+\varepsilon, 0)$ has $s_{\mathsf{a},p} = 1$. In this profile, the $(k+3)$-th buyer is reserve prices at 1 and the second buyer is the dummy buyer.

And for the third auction, we have:

- the profile $p = (i, \mathsf{b}_0, 1+\varepsilon, 0)$ has $s_{\mathsf{a},p} = \theta/k$ for $i \in [k]$. In this profile, the i-th buyer is reserve priced at $1+\varepsilon$ and the second buyer is the dummy buyer.

- the profile $p = (k+1, k+2, 1+\varepsilon, 1+\varepsilon)$ has weight $s_{\mathsf{a},p} = 1-\theta$. In this profile, both buyers $k+1$ and $k+2$ have reserve price $1+\varepsilon$ and thus the revenue is $1+\varepsilon$.

For this solution, we define the q variables as follows.

- For buyers $i \in [k]$, we set $q_{i,1} = (1-\theta)/k$ and $q_{i,1+\varepsilon} = 1 - q_{i,1}$.
- For buyers $i = k+1, k+2$, we set $q_{i,0} = \theta$ and $q_{i,1+\varepsilon} = 1 - q_{i,1}$.
- For buyer $k+3$, we set $q_{k+3,1+\varepsilon} = 1$.

It is easy to see that this solution is feasible and that it is the optimal solution to Problem (Profile-LP). This is so because for any auction, any profile that has a positive weight yield the maximum revenue for that auction. Note that for simplicity in the formulation of revenue, we can remove the terms that are a factor of ε since they can be arbitrary small and are negligible. We argue that the rounding procedure produces a $1/(c+1)$ approximation. First notice that the vector of zero reserves obtains revenue of $1/(c+1)$.

Now, we compute the expected revenue from rounding. After rounding, the reserve of any buyer $i \in [k]$ is either 1 or $1+\varepsilon$, the reserve of buyers $k+1$ and $k+2$ is either zero or $1+\varepsilon$, and reserve of buyer $k+3$ is always $1+\varepsilon$. Thus, by letting ε go to zero, the expected revenue from rounding is given by

$$\frac{1}{c+1}\left[1 - \left(1 - \frac{1-\theta}{k}\right)^k \cdot (1-\theta^2)\right] + \frac{c}{c+1},$$

where the first term is the revenue of first auction and the second term, i.e., $\frac{c}{c+1}$, is the revenue of the second auction.[2] To see why the latter holds note that in the first auction, we always get a revenue of one unless none of the first k buyers have a reserve of one and neither buyers $k+1$ nor buyer $k+2$ have a reserve of zero. As $k \to \infty$, the expected revenue after rounding becomes:

$$\frac{1}{c+1}\left[1 - e^{\theta-1} \cdot (1-\theta^2)\right] + \frac{c}{c+1} = \frac{1-c}{c+1} + \frac{c}{c+1} = \frac{1}{c+1},$$

where the first equation holds because $c = (1-\theta^2)e^{\theta-1}$. The above equation is the desired result because the optimal revenue is at most 1 and $1/(c+1) = 0.684$. The latter follows from $c = (1-\theta^2)e^{\theta-1}$ and $\theta = \sqrt{2}$. □

2.5 Integrality Gap

In this section, we give an upper-bound of 0.828 for the integrality gap of the LP. This implies that any rounding procedure for our LP formulation will obtain at most 0.828 fraction of the optimal value of the LP. In particular, we show Theorem 2, which we restate here for convenience.

Theorem 2 (Integrality Gap of Profile-LP). *There exists a data-set of bids* $\{\beta_{\mathsf{a},\mathsf{b}}\}_{\mathsf{a}\in\mathsf{A},\mathsf{b}\in\mathsf{B}}$ *for which the integrality gap of* Profile-LP *is at least* $2(\sqrt{2}-1) \approx 0.828$. *That is,*

$$\mathsf{ESP}^\star \leq 2(\sqrt{2}-1) \cdot \mathsf{LP}^\star,$$

where $\mathsf{LP}^\star$ *is the optimal fraction solution of the* Profile-LP *and* $\mathsf{ESP}^\star$ *is its optimal*

[2] We do not include the revenue of the third auction because we would like to take ε to zero and in that case, the revenue of the third auction approaches zero.

integral solution.

Proof. Given n buyers, an integer $k > 0$, $\delta = 1/k$ and a constant $\lambda \in (0, 1)$ to be determined later, consider an instance built as follows:

- **Type one Auctions:** For any buyer, $\mathsf{b} \in [n]$, we have an auction in which all the bids are zero except the bid of buyer b. Precisely, buyer b has a bid of λn.
- **Type two Auctions:** For any pair of buyers b_1 and b_2, there are k copies of an auction in which b_1 and b_2 bid $\delta = 1/k$ and the rest of the buyers bid 0. We assume that $\lambda n > \delta$.

For this instance, consider the fractional solution that assigns $s_{a,p} = 1/2$ for any auction a of type two and profiles $(\mathsf{b}_1, \mathsf{b}_0, \delta, 0)$ and $(\mathsf{b}_2, \mathsf{b}_0, \delta, 0)$. For the rest of the valid profiles of auction a, we set $s_{a,p}$ to zero. Note that b_1 and b_2 are the buyers with nonzero bids in auction a and b_0 is a dummy buyer. Moreover, for any auction a of type one, in which buyer b has a nonzero bid, we have $s_{a,p} = 1/2$ for profile $p = (\mathsf{b}, \mathsf{b}_0, \lambda n, 0)$. For the rest of the valid profiles of this auction, we set $s_{a,p}$ to zero. In this solution for any buyer b, we have $q_{\mathsf{b},\delta} = 1/2$ and $q_{\mathsf{b},\lambda n} = 1/2$. One can simply verify that this solution satisfies all the constraints of the LP and as a result, it is a valid fractional solution. The optimal value of the LP is therefore bounded by:

$$\mathsf{LP}^\star \geq \sum_{\mathsf{a}} \mathrm{Rev}_{\mathsf{a}}(\mathbf{s}) = n \cdot \frac{\lambda}{2} n + \binom{n}{2} \cdot k \cdot \delta = \frac{1+\lambda}{2} \cdot n^2 + o(n^2)\,,$$

where the first term corresponds to the revenue from auctions of type one and the second term corresponds to the revenue of auctions of type two. To bound $\mathsf{ESP}^\star$, we note that in the optimal solution of Problem (ESP-OPT), the reserve of each buyer is either δ or λn. Given that the buyers are symmetric, the value of the optimal solution depends only on the number of buyers with each reserve. Let t be the number of buyers with reserve λn. Then, we can write:

$$\mathsf{ESP}^\star = \max_{0 \le t \le n} \left[t \cdot \lambda n + (n-t) \cdot \delta + \binom{n}{2} - \binom{t}{2} \right].$$

By taking $\delta \to 0$, we obtain,

$$\mathsf{ESP}^\star = \max_{0 \le t \le n} \left[t \cdot \lambda n + \binom{n}{2} - \binom{t}{2} \right].$$

Since the term inside the maximum is a quadratic function of t, the optimal integral solution should be $t = \lambda n + o(n)$. This is so because the optimal integral solution t deviates from the optimal fractional solution (which is $\lambda n + 1/2$) by at most 1. Substituting that in the expression of $\mathsf{ESP}^\star$, we get

$$\mathsf{ESP}^\star = \frac{1+\lambda^2}{2} \cdot n^2 + o(n^2)\,.$$

Taking $n \to \infty$, we get

$$\frac{\mathsf{ESP}^\star}{\mathsf{LP}^\star} \le \frac{(1+\lambda^2)n^2 + o(n^2)}{(1+\lambda)n^2 + o(n^2)} \to \frac{1+\lambda^2}{1+\lambda}.$$

We can choose the parameter $\lambda = \sqrt{2} - 1$ to minimize the above expression, which leads to a ratio of $2(\sqrt{2} - 1) \approx 0.828$. □

Chapter 3: Personalized Reserve Prices in VCG Auctions

In this chapter we focus on data-driven optimization of personalized reserve prices in the *eager* VCG auction. We have k identical items and n unit-demand buyers participating in a set of eager VCG auctions. Let A and B respectively be the set of auctions and buyers. We are given a dataset of bids $\boldsymbol{\beta}$ where for any auction $\mathsf{a} \in \mathsf{A}$ and buyer $\mathsf{b} \in \mathsf{B}$, $\beta_{\mathsf{a},\mathsf{b}}$ represents bid of buyer b in auction a. Let r_{b} be the personalized reserve price of buyer b. Then, given the bid vector $\boldsymbol{\beta}_{\mathsf{a}}$ for auction a and reserve price vector $\mathbf{r}$, the eager VCG auction (EVCG) works as follows.

1. Any buyer b with $\beta_{\mathsf{a},\mathsf{b}} < r_{\mathsf{b}}$ is eliminated. Let $S_{\mathsf{a}} = \{\mathsf{b} : \beta_{\mathsf{a},\mathsf{b}} \geq r_{\mathsf{b}}\}$ be the set of buyers who clear their reserve prices in auction a.

2. An item is allocated to a buyer b if there are at most $k-1$ buyers in S_{a} whose bid is greater than $\beta_{\mathsf{a},\mathsf{b}}$.

3. Pick the *supporting buyer* b_s to be a buyer in set S_{a} such that there are exactly k buyers in S_{a} whose bid in auction a is greater than β_{a,b_s}.[1] ($\beta_{\mathsf{a},\mathsf{b}_s}$ is the VCG payment of any winner.)

4. Any buyer b who receives an item is charged $\max(r_{\mathsf{b}}, \beta_{\mathsf{a},\mathsf{b}_s})$, otherwise they are

[1] As usual, we assume that no two buyers have the same bids, or we can break ties based on their IDs.

not charged.

Given the dataset of bids $\boldsymbol{\beta}$, our goal is to find a vector of personalized reserve prices that maximize revenue of the auctioneer. Note that the reserve prices are the same across all the auctions $\mathsf{a} \in \mathsf{A}$. However, each buyer b is assigned a personalized reserve price r_{b}. We assume, w.l.o.g. that the optimal reserve price for any buyer is equal to one of their submitted bids. Let $\mathsf{R} = \{\beta_{\mathsf{a},\mathsf{b}} : \mathsf{a} \in \mathsf{A}, \mathsf{b} \in \mathsf{B}\}$. Formally, we would like to solve the following optimization problem:

$$\mathsf{EVCG}^{\star} \;=\; \max_{\mathbf{r} \in \mathsf{R}^n} \mathsf{Rev}(\mathbf{r}), \quad \text{where} \quad \mathsf{Rev}(\mathbf{r}) := \textstyle\sum_{\mathsf{a} \in \mathsf{A}} \mathsf{Rev}_{\mathsf{a}}(\mathbf{r})$$

and $\mathsf{Rev}_{\mathsf{a}}(\mathbf{r})$ is the total payment in action a given the vector of reserve prices $\mathbf{r}$. Note that to solve this problem we face a search space of size $|\mathsf{R}|^n$ which is exponential in the input size.

3.1 Results and Techniques

Our algorithm consists of two main parts. First, we design a polynomial-size Integer Linear Program (ILP) to describe the optimal solution. By removing the integrality constraints, we obtain a polynomial-size LP, which gives us a fractional solution. The second part is LP-rounding. Using the optimal solution of the LP, we construct two different integral solutions. We then show that the best of three vectors: these two solutions and the all-zero vector of reserve prices, is a 0.63-approximation.

To write the LP, we first need a polynomial-size representation of the solution space in which the revenue can be computed using a linear function. The natural representations (e.g., a vector of reserve prices, or a reserve price per buyer) fail as they result in either an exponential-size solution space or a nonlinear revenue function. We come up with an alternative concise representation, based on the following observation: to compute the revenue of an auction, we do not need to know the reserve prices of all the buyers. Rather, it suffices to only know the reserve prices of the winners and the VCG payment, which is the $(k+1)$-th highest cleared bid. The buyer who has this bid is called the *supporting buyer*. We represent a solution based on its outcomes in different auctions, where the "outcome" specifies who are the winners and who is the supporting buyer, and what are their respective reserve prices. The revenue from each auction can be computed using a linear function based on its outcome; therefore, the overall revenue can also be written as a linear function which is sum of the revenue across all of these auctions.

In the previous chapter, we use a similar approach for the single item case, which falls short in the general case. There, we capture all pertinent information about an auction in a single "profile" which is then used to compute the revenue of the auction, and use these profiles to write an ILP. However, we need exponentially many profiles to extend this approach to the general case, essentially because we need information about all the winners in order to compute the auction's revenue.

We proceed as follows. Instead of capturing all information about the winners in a single profile, we partition this information into several sub-profiles, each containing only a single winner and the supporting buyer. One complication is that these sub-profiles should not contradict each other (e.g., having different supporting buyers).

This issue gets even more complicated when we relax the integrality constraint of the ILP. We resolve this by introducing new variables and constraints to our LP.

Next, we use the optimal solution to the LP to construct two vectors of reserve prices which we refer to as *inflated reserves* and *discounted reserves*. For each buyer b, we use the LP solution to choose a threshold t_{b} to determine if a reserve price is too high or too low. We construct two probability distributions: one over reserve prices above t_{b}, and another over reserve prices below t_{b}. We use these distributions to draw, resp., the inflated and discounted reserve prices.

Let us provide some intuition for why we choose these two different vectors of reserve prices. Recall that each buyer pays the maximum of its reserve price and the VCG payment. Let us partition the winners' payments into two types: one is from winners who pay their reserve prices, and another is from those paying the VCG payment. Note that setting smaller reserve prices results in clearing more bids, and thus a larger VCG payment. Roughly speaking, second-type revenue from the *discounted reserves* should be larger than that of the *inflated reserves*, while the opposite might hold for the first-type revenue. Intuitively, if most of the optimal revenue is from type-one payments *and* high reserve prices (at least t_{b} for each buyer b), then we expect the *inflated reserves* to give us a high revenue. Otherwise, if the optimal revenue is a combination of type-one and type-two payments from small reserve prices, we expect a high revenue from the *discounted reserves*. But what if the type-two payments form a substantial portion of the revenue? This is where the vector of all-zero reserve prices comes into play. The all-zero reserve prices obtain the maximum possible revenue one can get from the type-two payments as all the bids are cleared in this case.

We analyze these three solutions simultaneously to lower bound the revenue of each solution as a function of the other two. By exploiting structural properties of our problem and the LP, we reduce the problem of finding the approximation factor to a complex non-linear optimization problem. This reduction, and solving this optimization problem, are the most technically challenging parts of the analysis (see Section 3.4).

To further highlight the significance of rounding our fractional solutions in two different ways, we investigate the performance of the simple rounding procedure used in the previous chapter which only outputs a single integral solution. Roughly speaking, that simple rounding directly use the fractional solution of the LP as a probability distribution over the reserve prices, and for each buyer independently draw a reserve price from that. We show that for a large number of items, this approach fails to beat the greedy algorithm. More precisely, for any given constant $0 < \varepsilon$, we construct an example for which the solution obtained using this rounding procedure gets at most $0.5 + \varepsilon$ fractional of the optimal revenue. We describe this example in Section 3.5.

3.2 The Algorithm

In this section we provide an LP-based algorithm for finding a vector of personalized reserve prices given a dataset of bids. We observe that to be able to describe the optimal solution of the problem using polynomially many linear constraints, we need a concise representation of the solution space. In Section 3.2.1, we explain this representation. In Section 3.2.2, we use this representation to design an LP and prove that its objective function in its optimal solution is an upper bound for the revenue of the optimal solution of the problem. Finally, in Section 3.2.3, we provide our rounding procedure that uses the optimal solution of the LP and outputs a vector of reserve prices.

3.2.1 An Alternative Solution Space

In this section, our goal is to give a concise representation of the solution space which will help us to write our linear program. We need this representation to have a polynomial size and it should be possible to compute its revenue using a linear function. As mentioned before, we base our design on the observation that to compute the revenue of an auction we do not need to have the reserve price of all the buyers. Rather, it only suffices to know the reserve prices of the winners and bid of the supporting buyer. The main idea here is to have variables that capture the outcome of auctions (i.e., who the winners and the supporting buyer are and their reserve prices.) instead of just having variables for reserve price of buyers. We will define *valid profiles* and *valid sub-profiles* of an auction to capture its outcome. Roughly speaking, the revenue obtained from each auction can be computed in a linear way based on its outcome; thus the overall revenue can also be written as a linear function which is sum of the revenue across all these auction.

Definition 3.2.1 (Valid Profiles)**.** *We define the set of valid profiles of an auction* a *as the set* $\mathcal{P}_{\mathsf{a}}$ *consisting of all tuples* $(\mathsf{b}_1, \ldots, \mathsf{b}_{k+1}, r_1, \ldots, r_{k+1}) \in \mathsf{B}^{k+1} \times \mathsf{R}^{k+1}$ *that satisfy the following conditions:*

1. *For any* $i, j \in [k+1]$ *where* $i < j$*, bid of buyer* b_i *is greater than or equal to that of buyer* b_j *in auction* a*; that is,* $\beta_{\mathsf{a},\mathsf{b}_i} \geq \beta_{\mathsf{a},\mathsf{b}_j}$*.*

2. *For any* $i \in [k+1]$ *buyer* b_i *clears his reserve,* r_i*, in auction* a*; that is,* $\beta_{\mathsf{a},\mathsf{b}_i} \geq r_i$*.*

Valid profiles are defined to capture the outcome of an auction given a set of reserve prices. However, note that each profile consists of at least $k+1$ buyers; thus, to be able to capture all the possible scenarios (for example when fewer than $k+1$ buyers clear their reserves in an auction), we add $k+1$ auxiliary buyers $\hat{\mathsf{b}}_1 \dots \hat{\mathsf{b}}_{k+1}$ to B who bid zero in all the auctions. Also, w.l.o.g., we assume that their reserves are always set to zero as well.

As mentioned previously, in the previous chapter we use a concept similar to valid profiles to write our linear program for $k = 1$. In that LP, we have a variable for any pair of auction and valid profile. However, it does not work here since it results in having exponentially many variables. To overcome this, we define *valid sub-profiles* of an auction that only contains information about a single winner and the supporting buyer as defined below.

Definition 3.2.2 (Sub-profiles)**.** *We define the set of valid Sub-profiles of an auction* a *as the set* $\mathcal{S}_\mathsf{a}$ *consisting of all tuples* $(\mathsf{b}_1, \mathsf{b}_2, r_1, r_2) \in \mathsf{B}^2 \times \mathsf{R}^2$ *that satisfy the following conditions:*

1. *Bid of buyer* b_1 *is greater than or equal to that of buyer* b_2 *in auction a; that is,* $\beta_{\mathsf{a},\mathsf{b}_1} \geq \beta_{\mathsf{a},\mathsf{b}_2}$*.*

2. *Buyers* b_1 *and* b_2 *clear their reserves in auction* a *if reserve prices* r_1 *and* r_2 *are*

set for them respectively; that is, $\beta_{\mathsf{a},\mathsf{b}_1} \geq r_1$ *and* $\beta_{\mathsf{a},\mathsf{b}_2} \geq r_2$*.*

For any given $p = (\mathsf{b}_1, \mathsf{b}_2, r_1, r_2) \in \mathcal{S}_{\mathsf{a}}$*, we have* $\mathsf{Rev}_{\mathsf{a}}(p) := \max(\beta_{\mathsf{a},\mathsf{b}_2}, r_1)$*.*

Given a vector of reserve prices $\mathbf{r}'$, we say a sub-profile $(\mathsf{b}_1, \mathsf{b}_2, r_1, r_2) \in \mathcal{S}_{\mathsf{a}}$ happens in auction a after applying $\mathbf{r}'$, iff $r'_{\mathsf{b}_1} = r_1$, $r'_{\mathsf{b}_2} = r_2$, buyer b_1 is a winner in auction a and buyer b_2 is the supporting buyer in this auction. Moreover, we say two sub-profiles $(\mathsf{b}_1, \mathsf{b}_2, r_1, r_2)$ and $(\mathsf{b}'_1, \mathsf{b}'_2, r'_1, r'_2)$ are compatible iff $\mathsf{b}'_2 = \mathsf{b}_2$ and $r'_2 = r_2$ which means that they have the same information about the supporting buyer. Moreover, we say a set P of valid sub-profiles are compatible iff they are pairwise compatible and $|P| = k$. Since we have added $k+1$ auxiliary buyers whose bid is always cleared in all the auction, we can assume that we always have exactly k winners and a supporting buyer.

To explain how a solution is represented using these sub-profiles, we consider a vector of reserve prices $\mathbf{r}$ and construct its representation in this new solution space. For any auction a and any sub-profile $p \in \mathcal{S}_{\mathsf{a}}$, we have a variable $s_{\mathsf{a},p}$ which is equal to one iff sub-profile p happens in auction a after applying vector of reserve prices $\mathbf{r}$. Otherwise we have $s_{\mathsf{a},p} = 0$. We say vector $\mathbf{s}$ constructed in this way is the representation of $\mathbf{r}$ in the profile space. As mentioned above, this representation allows us to compute the revenue of each auction using a linear function. Recall that for any sub-profile $p = (\mathsf{b}_1, \mathsf{b}_2, r_1, r_2)$ we have $\mathsf{Rev}_{\mathsf{a}}(p) := \max(\beta_{\mathsf{a},\mathsf{b}_2}, r_1)$, thus we can write

$$\mathsf{Rev}_{\mathsf{a}}(\mathbf{r}) = \sum_{p \in \mathcal{S}_{\mathsf{a}}} s_{\mathsf{a},p} \cdot \mathsf{Rev}_{\mathsf{a}}(p).$$

This function is linear since we have polynomially many valid sub-profiles; thus, for any valid sub-profile p we can simply compute $\mathsf{Rev}_{\mathsf{a}}(p)$ in advance and treat it as a constant in the LP.

3.2.2 The Linear Program

In this section we first design an integer linear program (ILP) then remove its integrality constraints to get an LP. We start by introducing the variables of our ILP. We have four vectors of random variables $\mathbf{s}$, $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{y}'$ as defined below.

1. For any auction $\mathsf{a} \in \mathsf{A}$ and any sub-profile $p \in \mathcal{S}_{\mathsf{a}}$, we have a variable $s_{\mathsf{a},p} \in \{0,1\}$ which is equal to one iff sub-profile p happens in auction a. This set of variables should satisfy constraint $\sum_{p \in \mathcal{S}_{\mathsf{a}}} s_{\mathsf{a},p} \leq k$ as at most k sub-profiles can happen in an auction.

2. For any buyer $\mathsf{b} \in \mathsf{B}$ and any reserve price $r \in \mathsf{R}$ we have a variable $x_{\mathsf{b},r} \in \{0,1\}$. Reserve price r is assigned to buyer b iff $x_{\mathsf{b},r} = 1$. For this type of variables we enforce the necessary constraint $\sum_{r \in \mathsf{R}} x_{\mathsf{b},r} = 1$ in our LP since each buyer has exactly one reserve price.

3. For any buyer $\mathsf{b} \in \mathsf{B}$, any auction a and any reserve price $r \in \mathsf{R}$, we have a variable $y_{\mathsf{b},r,a} \in \{0,1\}$ that is equal to one iff buyer b is assigned a reserve price of r as a winner in auction a.

4. For any buyer $\mathsf{b} \in \mathsf{B}$, any auction a and any reserve price $r \in \mathsf{R}$, we have a variable $y'_{\mathsf{b},r,a} \in \{0,1\}$ that is equal to one iff buyer b is assigned a reserve price of r as the supporting buyer in auction a.

Roughly speaking, the vector of variables $\mathbf{x}$ is used in the LP to ensure that in different auctions, solution $\mathbf{s}$ does not assign different reserve prices to the same buyer. Moreover variables $\mathbf{y}$ and $\mathbf{y}'$ are to ensure that the sub-profiles in an auction are compatible with each other.

To be able to write the constraints of our LP, we first need the following definitions. Let $\mathcal{Q}_{\mathsf{b},\mathsf{a}} := \{(\mathsf{b}, \mathsf{b}_2, r_1, r_2) \in \mathcal{S}_{\mathsf{a}} \; | \mathsf{b}_2 \in \mathsf{B}, r_1 \in \mathsf{R}, r_2 \in \mathsf{R}\}$ denote the set of valid sub-profiles of auction a in which buyer b is the winner and $\mathcal{Q}'_{\mathsf{b},\mathsf{a}} = \{(\mathsf{b}_1, \mathsf{b}, r_1, r_2) \in \mathcal{S}_{\mathsf{a}} \; | \mathsf{b}_1 \in \mathsf{B}, r_1 \in \mathsf{R}, r_2 \in \mathsf{R}\}$ is the set of valid sub-profiles of auction a in which buyer b is the supporting buyer. Moreover, let us define $\mathcal{Q}_{\mathsf{b},r,\mathsf{a}} := \{(\mathsf{b}, \mathsf{b}_2, r, r_2) \in \mathcal{Q}_{\mathsf{b},\mathsf{a}} | \mathsf{b}_2 \in \mathsf{B}, r_2 \in \mathsf{R}\}$ and $\mathcal{Q}'_{\mathsf{b},r,\mathsf{a}} := \{(\mathsf{b}_1, \mathsf{b}, r_1, r) \in \mathcal{Q}'_{\mathsf{b},\mathsf{a}} | \mathsf{b}_1 \in \mathsf{B}, r_1 \in \mathsf{R}\}$. We are now ready to write our LP (ILP without the integrality constraints) which we present in Figure 1.

$$
\begin{aligned}
\max_{\mathbf{x},\mathbf{s}} \quad & \sum_{\mathsf{a} \in \mathsf{A}} \sum_{p \in \mathcal{S}_{\mathsf{a}}} s_{\mathsf{a},p} \cdot \mathsf{Rev}_{\mathsf{a}}(p) & & \\
\text{s.t.} \quad & y_{\mathsf{b},r,\mathsf{a}} = \sum_{p \in \mathcal{Q}_{\mathsf{b},r,\mathsf{a}}} s_{\mathsf{a},p} & \forall \mathsf{a}, \mathsf{b}, r : \mathsf{b} \in \mathsf{B}, \mathsf{a} \in \mathsf{A}, r \in \mathsf{R} \quad & (1) \\
& y'_{\mathsf{b},r,\mathsf{a}} = \sum_{p \in \mathcal{Q}'_{\mathsf{b},r,\mathsf{a}}} \frac{s_{\mathsf{a},p}}{k} & \forall \mathsf{a}, \mathsf{b}, r : \mathsf{b} \in \mathsf{B}, \mathsf{a} \in \mathsf{A}, r \in \mathsf{R} \quad & (2) \\
& y_{\mathsf{b},r,\mathsf{a}} + y'_{\mathsf{b},r,\mathsf{a}} \le x_{\mathsf{b},r} & \forall \mathsf{a}, \mathsf{b}, r : \mathsf{b} \in \mathsf{B}, \mathsf{a} \in \mathsf{A}, r \in \mathsf{R}, \quad & (3) \\
& \sum_{\substack{p \in \mathcal{Q}_{\mathsf{b}_2,\mathsf{a}} \\ \cap \mathcal{Q}'_{\mathsf{b}_1,\mathsf{a}}}} s_{\mathsf{a},p} \le \sum_{r \in \mathsf{R}} y'_{\mathsf{b}_1,r,\mathsf{a}} & \forall \mathsf{a}, \mathsf{b}_1, \mathsf{b}_2 : \mathsf{b}, \mathsf{b}_2 \in \mathsf{B}, \mathsf{a} \in \mathsf{A} \quad & (4) \\
& \sum_{p \in \mathcal{S}_{\mathsf{a}}} s_{\mathsf{a},p} \le k & \forall \mathsf{a} : \mathsf{a} \in \mathsf{A} \quad & (5) \\
& \sum_{r \in \mathsf{R}} x_{\mathsf{b},r} = 1 & \forall \mathsf{b} : \mathsf{b} \in \mathsf{B} \quad & (6) \\
& s_{\mathsf{a},p} \ge 0 & \forall \mathsf{a}, p : \mathsf{a} \in \mathsf{A}, \; p \in \mathcal{S}_{\mathsf{a}} \quad & (7)
\end{aligned}
$$

Figure 3.1: The Linear Program

In the rest of this chapter we use $\mathbf{s}^\star$ to refer to $\mathbf{s}$ from an optimal solution of the LP. To be able to use the optimal solution of the LP as a benchmark in analyzing the approximation-factor of our algorithm, we need to show that it is indeed an upper bound for the optimal integral solution.

Lemma 3.2.3. *The optimal revenue is upper bounded by* $\sum_{\mathsf{a}\in\mathsf{A}} \sum_{p\in\mathcal{S}_{\mathsf{a}}} s^{\star}_{\mathsf{a},p} \cdot \mathsf{Rev}_{\mathsf{a}}(p)$.

We have Section 3.7 designated to the formal proof of this lemma, and also give an informal overview of that here. Roughly speaking, to prove this lemma, it suffices to show that the constraints of the LP are all necessary for the consistency of the variables that we have defined. Below we give some intuition about each constraint and why it is necessary.

Constraint (1) is due to the fact that for any auction a, buyer b, and reserve price r, variable $y_{\mathsf{b},r,\mathsf{a}}$ indicates whether or not buyer b has a reserve r and is a winner in auction a. This constraint ensures that value of $y_{\mathsf{b},r,\mathsf{a}}$ is consistent with whether or not there is a profile happening in auction a in which buyer b is a winner as is assigned a reserve price of r. Constraint (2) is similar to the previous one but for the supporting buyers. Constraint (3) is because the reserve prices assigned to a buyer in different auctions should be consistent. Moreover, imposing constraint (4), on the sub-profiles is to make sure that the sub-profiles that happen in an auction can form valid profiles. Consider an auction a and buyers b_1 and b_2. The right-hand-side of this constraint is the probability with which buyer b_1 is the supporting buyer in auction a, and the left-hand-side is the probability with which buyer b_2 is a winner while buyer b_1 is the supporting buyer which should obviously be smaller than the probability that buyer b_2 is a supporting buyer. Finally, constraints (5), (6), and (7) are by definition of variables. This LP upper bounds the optimal solution (the proof is deferred to the appendix).

3.2.3 The LP-Rounding Algorithm

In this section, given an optimal solution of the LP we generate an integral solution for the problem. The input of the algorithm is the vector $\mathbf{x}$ from an optimal solution of the LP and a parameter $\beta \in [0,1]$, which we fix later.

1. For any buyer b let t_{b} be the maximum number in R that satisfies $\sum_{r<t_{\mathsf{b}}} x_{b,r} \leq \beta$.

2. Define vectors $\mathbf{f}$ and $\mathbf{f}'$ as follows: For any $r \in \mathsf{R}$ where $r < t_{\mathsf{b}}$ set $f_{b,r} := x_{b,r}/\beta$ and $f'_{b,r} := 0$. For any $r > t_{\mathsf{b}}$, set $f_{b,r} := 0$ and $f'_{b,r} := x_{b,r}/(1-\beta)$. Finally for $r = t_b$, set $f_{b,r} := 1 - \sum_{r'<t_b} f_{b,r'}$ and $f'_{b,r} := 1 - \sum_{r'>t_b} f'_{b,r'}$.

3. Construct $\mathbf{r}$ the vector of *discounted* reserve prices as follows: For any buyer b independently choose a random reserve price $r_{\mathsf{b}} \in \mathsf{R}$ such that for any $\rho \in \mathsf{R}$, we have $\Pr[r_{\mathsf{b}} = \rho] = f_{\mathsf{b},\rho}$.

4. Construct $\mathbf{r}'$ the vector of *inflated* reserve prices as follows: For any buyer b independently choose a random reserve price $r'_{\mathsf{b}} \in \mathsf{R}$ such that for any $\rho \in \mathsf{R}$ we have $\Pr[r'_{\mathsf{b}} = \rho] = f'_{\mathsf{b},\rho}$.

5. Let $\mathbf{z}$ be the vector of all zero reserve prices.

6. Between $\mathbf{z}$, $\mathbf{r}$ and $\mathbf{r}'$ return the one with higher revenue which is as follows:

$$\arg\max_{\boldsymbol{\nu} \in \{\mathbf{z},\mathbf{r},\mathbf{r}'\}} \mathsf{Rev}(\nu).$$

We will use $\mathbf{r}$ and $\mathbf{r}'$ to respectively refer to the vector of discounted and inflated reserve prices constructed in this algorithm.

Remark 3.2.4. *For the sake of simplicity in the analysis, we assume w.l.o.g., that for any auction* a*,* t_b *satisfies* $\sum_{r<t_\mathsf{b}} x_{b,r} = \beta$*. This implies that for any* $r \in \mathsf{R}$*, if* $r < t_b$ *we have* $\Pr[r_b = r] = x_{\mathsf{b},r}/\beta$ *and* $\Pr[r'_b = r] = 0$*. Otherwise, if* $r \geq t_b$*, we have* $\Pr[r_b = r] = 0$ *and* $\Pr[r'_b = r] = x_{\mathsf{b},r}/(1-\beta)$*.*

In the next section we show how we can use specific features of the three solutions $\mathbf{z}$, $\mathbf{r}$ and $\mathbf{r}'$ to get our desired approximation-factor.

3.3 Approximation Factor

In this section, we prove our main theorem by giving a lower bound for the revenue obtained from the vector of reserve prices outputted by the rounding algorithm.

Let us start by giving some definitions that will be used throughout this section. Define $\beta_\mathsf{a}^{(k+1)}$ to be the $(k+1)$-th highest bid in any auction a. Note that this is different from the bid of the supporting buyer in auction a as the supporting buyer has the $(k+1)$-th highest bid after removing the buyers whose bid is not cleared. Moreover, given a threshold τ, let us denote by $W_\mathsf{a}(\mathbf{r}, \tau)$ the number of winners in auction a whose payment is greater than or equal to τ using the vector of reserve prices $\mathbf{r}$. Similarly, $W_\mathsf{a}(\mathbf{r}', \tau)$ is the number of winners who pay at least τ using vector of reserve prices $\mathbf{r}'$ in auction a. Note that $W_\mathsf{a}(\mathbf{r}, \tau)$ and $W_\mathsf{a}(\mathbf{r}', \tau)$ are both random variables.

The following lemma establishes sufficient conditions for the algorithm to output an approximate solution. For any given auction $\mathsf{a} \in \mathsf{A}$ and a real number $\tau \geq 0$, we define

$$\Phi_a(\tau) := \sum_{\substack{p:p\in\mathcal{S}_{\mathsf{a}},\\ \mathsf{Rev}_{\mathsf{a}}(p)\geq\tau}} s^{\star}_{\mathsf{a},p} - (1-\beta)\mathbb{E}[W_a(\mathbf{r}',\tau)] - \beta\mathbb{E}[W_{\mathsf{a}}(\mathbf{r},\tau)].$$

Lemma 3.3.1. *Suppose there exist absolute constants $\beta \in (0,1)$ and $c \in (0,1)$ such that,*

$$\Phi_a(\tau) \leq 0 \qquad \textit{if} \quad \tau > \beta_{\mathsf{a}}^{(k+1)},\ \textit{and} \tag{3.1}$$

$$\Phi_a(\tau) \leq kc \qquad \textit{if} \quad \tau \leq \beta_{\mathsf{a}}^{(k+1)} \tag{3.2}$$

for any auction $\mathsf{a} \in \mathsf{A}$. Then, the algorithm with parameter β outputs a $\frac{1}{1+c}$-approximate solution.

Proof. Consider an arbitrary auction a. By integrating over τ in $\Phi_a(\tau)$, we obtain:

$$\int_{(\beta_{\mathsf{a}}^{(k+1)},\infty)} \Phi_a(\tau)\ d\tau\ +\ \int_{(0,\beta_{\mathsf{a}}^{(k+1)}]} \Phi_a(\tau)\ d\tau \leq kc\beta_{\mathsf{a}}^{(k+1)}.$$

By simplifying this we get

$$\sum_{p\in\mathcal{S}_{\mathsf{a}}} s_{\mathsf{a},p}\cdot\mathsf{Rev}_{\mathsf{a}}(p) - (1-\beta)\mathbb{E}[\mathsf{Rev}_{\mathsf{a}}(\mathbf{r}')] - \beta\mathbb{E}[\mathsf{Rev}_{\mathsf{a}}(\mathbf{r})] \leq kc\beta_{\mathsf{a}}^{(k+1)}. \tag{3.3}$$

Recall that the output of our algorithm is the best of $\mathbf{r}$, $\mathbf{r}'$, and $\mathbf{z}$ where $\mathbf{z}$ is the vector of all-zero reserve prices and its revenue is $k\beta_{\mathsf{a}}^{(k+1)}$ since by applying that, the players in the k first positions win the k items and pay bid of the buyer in the $(k+1)$-th position. Therefore, the expected revenue achieved from the output of our algorithm is at least

$$\mu := \max\left(\mathbb{E}[\mathsf{Rev}(\mathbf{r})], \mathbb{E}[\mathsf{Rev}(\mathbf{r}')], k \cdot \sum_{\mathsf{a}\in\mathsf{A}} \beta_{\mathsf{a}}^{(k+1)}\right).$$

Based on Equation 3.3 we have

$$\mathsf{Rev}(\mathbf{s}^\star) - (1-\beta)\mu - \beta\mu - c\mu \leq 0.$$

where $\mathsf{Rev}(\mathbf{s}^\star) := \sum_{\mathsf{a}\in\mathsf{A}} \sum_{p\in\mathcal{S}_\mathsf{a}} s^\star_{\mathsf{a},p} \cdot \mathsf{Rev}_\mathsf{a}(p)$. This implies $\mathsf{Rev}(\mathbf{s}^\star) - (1+c)\mu \leq 0$, and as a result

$$\frac{\mathsf{Rev}(s^\star)}{1+c} \leq \max\big(\mathbb{E}[\mathsf{Rev}(\mathbf{r})], \mathbb{E}[\mathsf{Rev}(\mathbf{r}')], k\beta_{\mathsf{a}}^{(k+1)}\big).$$

Further by Lemma 3.2.3, we know that $\mathsf{Rev}(s^\star)$ is an upper bound for the revenue of the optimal solution; thus by setting $\beta = \beta'$ in the rounding algorithm its output is at least a $\frac{1}{1+c}$-approximate solution. □

Having, Lemma 3.3.1, it suffices to prove that Equation 3.1 always holds and find values for parameters β and c that satisfy Equation 3.2. We address the former in the following lemma, and the latter in Lemma 3.3.3.

Lemma 3.3.2. *For auction* a *and any* $\tau > \beta_{\mathsf{a}}^{(k+1)}$ *we have*

$$\sum_{\substack{p:p\in\mathcal{S}_\mathsf{a} \\ \mathsf{Rev}_\mathsf{a}(p)\geq\tau}} s^\star_{\mathsf{a},p} - (1-\beta)\mathbb{E}[W_a(\mathbf{r}', \tau)] - \beta\mathbb{E}[W_\mathsf{a}(\mathbf{r}, \tau)] \leq 0.$$

Proof. By definition of $W_{\mathsf{a}}(\mathbf{r}', \tau)$ and $W_{\mathsf{a}}(\mathbf{r}, \tau)$ for any $\tau > \beta_{\mathsf{a}}^{(k+1)}$, we have

$$(1-\beta)\mathbb{E}[W_a(\mathbf{r}', \tau)] + \beta\mathbb{E}[W_{\mathsf{a}}(\mathbf{r}, \tau)] = (1-\beta)\mathbb{E}\left[\min\left(\sum_{\mathsf{b}\in\mathsf{B}} 1_{r'_b>\tau}, k\right)\right] + \beta\mathbb{E}\left[\min\left(\sum_{\mathsf{b}\in\mathsf{B}} 1_{r_b>\tau}, k\right)\right].$$

Recall that $\beta_{\mathsf{a}}^{(k+1)}$ is defined in a way that there are exactly k buyers with bids greater than $\beta_{\mathsf{a}}^{(k+1)}$ in auction a. As a result, there are at most k buyers for whom $\Pr[r'_b > \tau]$ or $\Pr[r_b > \tau]$ is nonzero. This means that we can rewrite the inequality as

$$\begin{aligned}(1-\beta)\mathbb{E}[W_a(\mathbf{r}', \tau)] + \beta\mathbb{E}[W_{\mathsf{a}}(\mathbf{r}, \tau)] &= (1-\beta)\mathbb{E}\left[\sum_{\mathsf{b}\in\mathsf{B}} 1_{r'_b>\tau}\right] + \beta\mathbb{E}\left[\sum_{\mathsf{b}\in\mathsf{B}} 1_{r_b>\tau}\right] \\ &= \mathbb{E}\left[\sum_{\mathsf{b}\in\mathsf{B}} 1_{r_b>\tau}\right].\end{aligned}$$

Observe that by construction of $\mathbf{r}$ and $\mathbf{r}'$ we have

$$\begin{aligned}\mathbb{E}[1_{r_b>\tau}] &= \Pr[r_{\mathsf{b}} > \tau] = \frac{1}{\beta}\sum_{r\in(\tau, t_{\mathsf{b}})} x_{\mathsf{b},r} \quad \text{and} \\ \mathbb{E}[1_{r'_b>\tau}] &= \Pr[r'_{\mathsf{b}} > \tau] = \frac{1}{1-\beta}\sum_{r: r\geq t_{\mathsf{b}}, r>\tau} x_{\mathsf{b},r}.\end{aligned} \tag{3.4}$$

Moreover, by putting the first and third constraints of the LP together, for any buyer b and any reserve price $r > \tau$, we get $\sum_{p\in Q_{\mathsf{b},r,\mathsf{a}}} s^{\star}_{\mathsf{a},p} \leq x_{\mathsf{b},r}$. Note that for any $p = (\mathsf{b}_1, \mathsf{b}_2, r_1, r_2) \in \mathcal{S}_{\mathsf{a}}$, by definition, we have $\beta_{\mathsf{a}}^{(k+1)} \geq \beta_{\mathsf{a},\mathsf{b}_2}$, which means that if we have

$\mathsf{Rev}_\mathsf{a}(p) > \tau$, then $\mathsf{Rev}_\mathsf{a}(p) = r_1$ and $p \in Q_{\mathsf{b}_1,r_1,\mathsf{a}}$. This results in the following equations.

$$\sum_{\mathsf{b}\in\mathsf{B}} \sum_{r\in(\tau,t_\mathsf{b})} x_{\mathsf{b},r} \geq \sum_{\substack{p:p\in\mathcal{S}_\mathsf{a},\\ \mathsf{Rev}_\mathsf{a}(p)<t_\mathsf{b},\\ \mathsf{Rev}_\mathsf{a}(p)>\tau}} s^\star_{\mathsf{a},p}, \quad \text{and} \quad \sum_{\mathsf{b}\in\mathsf{B}} \sum_{\substack{r:r\geq t_\mathsf{b},\\ r>\tau}} x_{\mathsf{b},r} \geq \sum_{\substack{p:p\in\mathcal{S}_\mathsf{a},\\ \mathsf{Rev}_\mathsf{a}(p)\geq t_\mathsf{b},\\ \mathsf{Rev}_\mathsf{a}(p)>\tau}} s^\star_{\mathsf{a},p}.$$

Combining these with the equations in (3.4) gives us the following equation and concludes the proof.

$$(1-\beta)\mathbb{E}[W_a(\mathbf{r}',\tau)] + \beta\mathbb{E}[W_\mathsf{a}(\mathbf{r},\tau)] \geq \sum_{\substack{p:p\in\mathcal{S}_\mathsf{a}\\ \mathsf{Rev}_\mathsf{a}(p)>\tau}} s^\star_{\mathsf{a},p}.$$

□

The following lemma is the most technically challenging part of the analysis. Therefore, we have Section 3.4 assigned to its proof.

Lemma 3.3.3. *Setting $\beta = 0.55$ and $c = 0.58$, the following inequality holds for any auction $\mathsf{a} \in \mathsf{A}$ and any $0 < \tau \leq \beta_\mathsf{a}^{(k+1)}$.*

$$\sum_{\substack{p:p\in\mathcal{S}_\mathsf{a},\\ \mathsf{Rev}_\mathsf{a}(p)\geq\tau}} s^\star_{\mathsf{a},p} - (1-\beta)\mathbb{E}[W_\mathsf{a}(\mathbf{r}',\tau)] - \beta\mathbb{E}[W_\mathsf{a}(\mathbf{r},\tau)] \ \leq \ kc. \tag{3.5}$$

We are now ready to prove our main result. Below we restate the main theorem and prove it using the lemmas in this section.

Theorem 4. *There exists an algorithm with running time polynomial in the input size that outputs a vector of reserve prices $\mathbf{r}^o$ such that $\mathsf{Rev}(\mathbf{r}^o)$ is at least a 0.63 fraction of the revenue achieved from the optimal vector of reserve prices.*

Proof. First, note that the LP designed in Section 3.2.2 has polynomially many variables and constraints. To design our algorithm, we first solve the LP, then given an optimal solution of that use the LP-rounding procedure to output a vector of reserve prices. Since the LP rounding procedure has a polynomial running time, the total running time of the algorithm is polynomial as well.

To analyze the approximation factor of the algorithm we use Lemma 3.3.1, Lemma 3.3.3 and Lemma 3.3.2. The first lemma states that if there exists a constant $c \geq 0$ and a valuation for parameter β that satisfies Equation 3.1 for any $\tau > \beta_{\mathsf{a}}^{(k+1)}$, and satisfies Equation 3.2 for any $\tau \leq \beta_{\mathsf{a}}^{(k+1)}$, then our rounding algorithm is a $\frac{1}{1+c}$-approximation. In Lemma 3.3.3, we prove that Equation 3.1 holds for any $\beta \in (0, 1)$ and any $\tau > \beta_{\mathsf{a}}^{(k+1)}$. Moreover, based on Lemma 3.3.2 we have that by setting $\beta = 0.55$, Equation 3.2 holds for any $\tau \leq \beta_{\mathsf{a}}^{(k+1)}$ and $c = 0.58$. This implies that by setting $\beta = 0.55$ in the rounding algorithm, its output is a 0.63-approximation of the optimal solution since $\frac{1}{1+0.58} > 0.63$. □

3.4 Proof of Lemma 3.3.3

In this section, we will consider an arbitrary auction $\mathsf{a} \in \mathsf{A}$ and any constant $0 < \tau \leq \beta_{\mathsf{a}}^{(k+1)}$, and focus on finding a constant c and a valuation for β that satisfy

$$\sum_{\substack{p:p\in\mathcal{S}_{\mathsf{a}},\\ \mathsf{Rev}_{\mathsf{a}}(p)\geq\tau}} s^{\star}_{\mathsf{a},p} - (1-\beta)\mathbb{E}[W_{\mathsf{a}}(\mathbf{r}',\tau)] - kc \leq \beta\mathbb{E}[W_{\mathsf{a}}(\mathbf{r},\tau)].$$

Denote by $\mathsf{B}_{\mathsf{a},\tau}$ the set of buyers whose bid in auction a is greater than or equal to τ. Formally, we have $\mathsf{B}_{\mathsf{a},\tau} := \{\mathsf{b} \in \mathsf{B} : \beta_{\mathsf{a},\mathsf{b}} \geq \tau\}$. Throughout this section, since we assume that a can be any arbitrary auction from A, we will abbreviate all notations by dropping a for simplicity when clear from the context. Let us define function $F(\mathbf{s}^\star, \tau)$ as follows.

$$F(\mathbf{s}^\star, \tau) = \sum_{\substack{p: p \in \mathcal{S}_{\mathsf{a}}, \\ \mathsf{Rev}_{\mathsf{a}}(p) \geq \tau}} s^\star_{\mathsf{a},p} - (1-\beta)\mathbb{E}[W_{\mathsf{a}}(\mathbf{r}', \tau)] \tag{3.6}$$

We will consider different values of $F(\mathbf{s}^\star, \tau)$ as a function of β and give a lower bound for $\mathbb{E}[W_{\mathsf{a}}(\mathbf{r}, \tau)]$ based on that. Consider a buyer $\mathsf{b} \in \mathsf{B}_{\mathsf{a},\tau}$. Let us define Bernoulli random variables $p_{\mathsf{b},\tau}$ and q_{b} to be respectively equal to one iff $r_{\mathsf{b}} \in [\tau, \beta_{\mathsf{a},\mathsf{b}}]$ and equal to one iff $r_{\mathsf{b}} \in [0, \beta_{\mathsf{a},\mathsf{b}}]$. Moreover, let $P_\tau = \sum_{\mathsf{b} \in \mathsf{B}_\tau} p_{b,\tau}$ and $Q_\tau = \sum_{\mathsf{b} \in \mathsf{B}_\tau} q_{\mathsf{b}}$.

Claim 3.4.1. *The expected revenue obtained from the vector of reserve prices* $\mathbf{r}$ *is as follows.*

$$\mathbb{E}[W_{\mathsf{a}}(\mathbf{r}, \tau)] \geq \max\left(k.\Pr[Q_\tau > k], \mathbb{E}[\min(P_\tau, k)]\right)$$

Proof. Note that Q_τ is a random variable representing the number of buyers whose bid is cleared and is greater than or equal to τ; therefore, $Q_\tau > k$ is the event in which at least $k+1$ buyers have cleared their bid of at least τ which results in k items being sold with a price of at least τ. Moreover, P_τ denotes the number of buyers whose bid is cleared with a reserve of at least τ; thus, we sell at least $\min(P_\tau, k)$ of our items with a price of at least τ. This means that the expected number of items that are sold with a price of at least τ is lower bounded by $\max(k.\Pr[Q_\tau > k], \mathbb{E}[\min(P_\tau, k)])$. $\square$

Let $\mathcal{T}_{\mathsf{a},\tau}$ be the set of sub-profiles in $\mathcal{S}_\mathsf{a}$ whose revenue is at least τ. In the other words,

$$\mathcal{T}_{\mathsf{a},\tau} := \{p \in \mathcal{S}_\mathsf{a} | \mathsf{Rev}_\mathsf{a}(p) \geq \tau\}.$$

We partition $\mathcal{T}_{\mathsf{a},\tau}$ to three disjoint subsets denoted by $\mathcal{J}^+_\tau, \mathcal{J}^-_\tau$, and $\mathcal{L}_\tau$ as follows. Set $\mathcal{J}^+_\tau$ is the set of sub-profiles in $\mathcal{T}_\tau$ that capture the scenarios in which the supporting buyer has a bid smaller than τ and the winner's reserve price is greater than or equal to its threshold t_b (defined in the algorithm).

$$\mathcal{J}^+_\tau := \{p = (\mathsf{b}', \mathsf{b}, r', r) \in \mathcal{T}_\tau | \, \mathsf{b} \notin \mathsf{B}_\tau \text{ and } r' \geq t_{b'}\}.$$

We similarly define $\mathcal{J}^-_{\mathsf{b},\tau}$ to be the set of sub-profiles in $\mathcal{T}_\tau$ in which the supporting buyer has a bid smaller than τ and the reserve price of the winner is below its threshold t_b.

$$\mathcal{J}^-_\tau := \{p = (\mathsf{b}', \mathsf{b}, r', r) \in \mathcal{T}_\tau | \, \mathsf{b} \notin \mathsf{B}_\tau \text{ and } r' < t_{b'}\}.$$

Moreover, $\mathcal{L}_\tau$ defined below denotes the set of sub-profiles in $\mathcal{S}_\mathsf{a}$ that capture the scenarios in which the supporting buyer has a bid greater than or equal to τ.

$$\mathcal{L}_\tau := \{p = (\mathsf{b}', \mathsf{b}, r', r) \in \mathcal{T}_\tau | \, \mathsf{b} \in \mathsf{B}_\tau\}.$$

Further, based on this set, we define

$$\delta_\tau := \sum_{\mathsf{b} \in \mathsf{B}_\tau} \sum_{p \in \mathcal{L}_\tau \cap \mathcal{Q}'_{\mathsf{b}}} s^\star_{\mathsf{a},p}/k. \tag{3.7}$$

Observe that the defined subsets of $\mathcal{T}_\tau$ satisfy $\mathcal{T}_\tau = \mathcal{J}^+_\tau \cup \mathcal{J}^-_\tau \cup \mathcal{L}_\tau$.

Given Claim 3.4.1, we now need to find a lower bound for $\max\big(k.\Pr[Q_\tau > k], \mathbb{E}[\min(P_\tau, k)]\big)$ as a function of $F(s^\star, \tau)$ and δ_τ. To get this, we start by giving lower bounds for $\mathbb{E}[Q_\tau]$ and $\mathbb{E}[P_\tau]$ in the following section, then use the expected value of these random variables to bound the value of the functions $k.\Pr[Q_\tau > k]$ and $\mathbb{E}[\min(P_\tau, k)]$ in Section 3.4.2. Note that all these bound will be functions of $F(s^\star, \tau)$ and δ_τ.

3.4.1 lower bounds for $\mathbb{E}[P_\tau]$ and $\mathbb{E}[Q_\tau]$

In this section, we start by investigating useful facts about set $\mathcal{T}_\tau$, random variables $\mathbf{p}$ and $\mathbf{q}$ and the relation between them which finally leads to lower bounds for $\mathbb{E}[P_\tau]$ and $\mathbb{E}[Q_\tau]$. Let us mention that to prevent interruptions to the flow of the chapter, proofs of some of the lemmas in this section are deferred to Section 3.6.

We start by obtaining a lower bound for $\mathbb{E}[P_\tau]$, for which we make the three claims below.

Claim 3.4.2. *The following holds:* $\sum_{p \in \mathcal{L}_\tau} s^\star_{\mathsf{a},p} = k\delta_\tau$.

Proof. Recalling definition (3.7), it suffices to show $\bigcup_{\mathsf{b} \in \mathsf{B}_\tau} (\mathcal{L}_\tau \cap \mathcal{Q}'_{\mathsf{b}}) = \mathcal{L}_\tau$, as it results in

$$\delta_\tau = \sum_{\mathsf{b} \in \mathcal{L}_\tau} s^\star_{\mathsf{a},p}/k.$$

A valid sub-profile $p = (\mathsf{b}_1, \mathsf{b}_2, r_1, r_2)$ is in $\bigcup_{\mathsf{b} \in \mathsf{B}_\tau} \mathcal{Q}'_\mathsf{b}$ iff $\mathsf{b}_2 \in \mathsf{B}_\tau$ which also means $\mathsf{Rev}_\mathsf{a}(p) = \max(r_1, \beta_{\mathsf{a},\mathsf{b}}) \geq \tau$ and $p \in \mathcal{T}_\tau$. To complete the proof observe that this is indeed the definition of set $\mathcal{L}_\tau$ which is $\mathcal{L}_\tau := \{(\mathsf{b}', \mathsf{b}, r', r) \in \mathcal{T}_\tau | \, \mathsf{b} \in \mathsf{B}_\tau\}$. □

Claim 3.4.3. *For any buyer* $\mathsf{b} \in \mathsf{B}$ *we have*

$$\mathbb{E}[p_{b,\tau}] \geq \frac{1}{\beta}\Big(\sum_{p \in \mathcal{J}_\tau^- \cap \mathcal{Q}_\mathsf{b}} s^\star_{\mathsf{a},p}\Big).$$

Proof. By construction, for vector of reserve prices $\mathbf{r}$ and any buyer b we have

$$\mathbb{E}[p_{\mathsf{b},\tau}] = \Pr[r_\mathsf{b} \in [\tau, \beta_{\mathsf{a},\mathsf{b}}]] = \sum_{r \in [\tau, \beta_{\mathsf{a},\mathsf{b}}]} f_{\mathsf{b},r},$$

where $f_{\mathsf{b},r} = x_{b,r}/\beta$ for any $r < t_\mathsf{b}$ as defined in the algorithm. This yields that

$$\mathbb{E}[p_{\mathsf{b},\tau}] \geq \sum_{\substack{r: r < t_\mathsf{b}, \\ r \in [\tau, \beta_{\mathsf{a},\mathsf{b}}]}} \frac{x_{b,r}}{\beta} \geq \sum_{\substack{r: r < t_\mathsf{b}, \\ r \in [\tau, \beta_{\mathsf{a},\mathsf{b}}]}} \sum_{p \in \mathcal{Q}_{\mathsf{b},r,\mathsf{a}}} \frac{s_{\mathsf{a},p}}{\beta},$$

where the second inequality is by the first and third constraints of the LP. To complete the proof it suffices to show that

$$(\mathcal{J}_\tau^- \cap \mathcal{Q}_\mathsf{b}) \subset \bigcup_{\substack{r: r < t_\mathsf{b}, \\ r \in [\tau, \beta_{\mathsf{a},\mathsf{b}}]}} \mathcal{Q}_{\mathsf{b},r,\mathsf{a}}.$$

Observe that we have

$$(\mathcal{J}_\tau^- \cap \mathcal{Q}_\mathsf{b}) = \{(\mathsf{b}, \mathsf{b}_2, r, r_2) \in \mathcal{T}_\tau | \, r \in \mathsf{R}, r_2 \in \mathsf{R}, \mathsf{b}_2 \in \mathsf{B}_\tau \text{ and } r < t_\mathsf{b}\}.$$

Moroever, note that for any sub-profile $p = (\mathsf{b}, \mathsf{b}_2, r, r_2) \in \mathcal{T}_\tau$ we have $\mathsf{Rev}_\mathsf{a}(p) = \max(r, \beta_{\mathsf{a},\mathsf{b}_2}) \geq \tau$, thus if $\beta_{\mathsf{a},\mathsf{b}_2} < \tau$ then $r \geq \tau$. As a result we have

$$(\mathcal{J}_\tau^- \cap \mathcal{Q}_\mathsf{b}) \subset \{(\mathsf{b}, \mathsf{b}_2, r, r_2) \in \mathcal{S}_\mathsf{a} | \, r \in \mathsf{R}, r_2 \in \mathsf{R}, \mathsf{b}_2 \in \mathsf{B}_\tau \text{ and } r < t_\mathsf{b}\}.$$

Further, since $\mathcal{Q}_{\mathsf{b},r,\mathsf{a}} := \{(\mathsf{b}, \mathsf{b}_2, r, r_2) \in \mathcal{S}_\mathsf{a} | \mathsf{b}_2 \in \mathsf{B}, r_2 \in \mathsf{R}\}$, then

$$\bigcup_{\substack{r: r < t_\mathsf{b}, \\ r \in [\tau, \beta_{\mathsf{a},\mathsf{b}}]}} \mathcal{Q}_{\mathsf{b},r,\mathsf{a}} = \{(\mathsf{b}, \mathsf{b}_2, r, r_2) \in \mathcal{S}_\mathsf{a} | \, r \in \mathsf{R}, r_2 \in \mathsf{R}, \mathsf{b}_2 \in \mathsf{B}, r \geq \tau, r \geq \beta_{\mathsf{a},\mathsf{b}}, r < t_\mathsf{b}\}.$$

Note that any $(\mathsf{b}, \mathsf{b}_2, r, r_2) \in \mathcal{S}_\mathsf{a}$ satisfies $r \geq \beta_{\mathsf{a},\mathsf{b}}$, therefore we get

$$(\mathcal{J}_\tau^- \cap \mathcal{Q}_\mathsf{b}) \subset \bigcup_{\substack{r: r < t_\mathsf{b}, \\ r \in [\tau, \beta_{\mathsf{a},\mathsf{b}}]}} \mathcal{Q}_{\mathsf{b},r,\mathsf{a}}. \quad \square$$

The following lemma is the last piece that we need to get the desired lower bound for $\mathbb{E}[P_\tau]$ in Lemma 3.4.5. The proof of this lemma due to being lengthy is deferred to Section 3.6.

Lemma 3.4.4. *The following inequality holds.*

$$F(\mathbf{s}^\star, \tau) \leq \sum_{p \in \mathcal{J}_\tau^-} s^\star_{\mathsf{a},p} + \sum_{p \in \mathcal{L}_\tau} s^\star_{\mathsf{a},p}.$$

Lemma 3.4.5. *We have the following lower bound for* $\mathbb{E}[P_\tau]$:

$$\mathbb{E}[P_\tau] \geq \frac{F(\mathbf{s}^\star, \tau) - k\delta_\tau}{\beta}.$$

Proof. Based on Lemma 3.4.4 we have

$$F(\mathbf{s}^\star, \tau) - \sum_{p \in \mathcal{L}_\tau} s^\star_{\mathsf{a},p} \leq \sum_{p \in \mathcal{J}^-_\tau} s^\star_{\mathsf{a},p}.$$

Combining this by $\sum_{p \in \mathcal{L}_\tau} s^\star_{\mathsf{a},p} = k\delta_\tau$ from Claim 3.4.2 and diving both sides by β gives us:

$$\frac{F(\mathbf{s}^\star, \tau) - k\delta_\tau}{\beta} \leq \frac{1}{\beta} \sum_{p \in \mathcal{J}^-_\tau} s^\star_{\mathsf{a},p}$$

We conclude the proof by noting that as a result of Claim 3.4.3, we have $\mathbb{E}[P_\tau] \geq \frac{1}{\beta} \sum_{p \in \mathcal{J}^-_\tau} s^\star_{\mathsf{a},p}$. □

Getting the desired lower bound for $\mathbb{E}[Q_\tau]$ is however more complicated than that of $\mathbb{E}[P_\tau]$. In Lemma 3.4.6 and Lemma 3.4.7 we give two different lower bounds for $\mathbb{E}[Q_\tau]$ which we then merge in Lemma 3.4.8 to obtain an stronger one. The proof of both these lemmas are based on careful analysis of the relations between $\mathbf{q}$ and subsets of $\mathcal{T}_\tau$, and are deferred to Section 3.6 due to being very complicated.

Lemma 3.4.6. *For* $\mathsf{B}_1 = \{\mathsf{b} \in \mathsf{B} : \mathbb{E}[q_\mathsf{b}] = 1\}$, *we have*

$$\mathbb{E}[Q_\tau - |\mathsf{B}_1|] \geq (k - |\mathsf{B}_1| + 1)\delta/\beta.$$

Lemma 3.4.7. *For* $\mathsf{B}_1 = \{\mathsf{b} \in \mathsf{B} : \mathbb{E}[q_\mathsf{b}] = 1\}$ *and* $m = |\mathsf{B}_1|$ *we have*

$$\mathbb{E}[Q_\tau - m] \geq \frac{F(s^\star, \tau) - m\delta}{\beta} + \delta/\beta.$$

Lemma 3.4.8. *For* $\mathsf{B}_1 = \{\mathsf{b} \in \mathsf{B} : q_{\mathsf{b},\tau} = 1\}|$ *and* $m = |\mathsf{B}_1|$*, we have*

$$\mathbb{E}[Q_\tau - m] \geq \frac{k.\max(F(s^\star, \tau)/k, \delta_\tau) - (m-1)\delta_\tau}{\beta}.$$

Proof. This is a direct result of the lower bounds given in Lemma 3.4.6 and Lemma 3.4.7, which are respectively as follows.

$$\mathbb{E}[Q_\tau - m] \geq (k - m + 1)\delta_\tau/\beta.$$

$$\mathbb{E}[Q_\tau - m] \geq \frac{F(s^\star, \tau) - m\delta_\tau}{\beta} + \delta_\tau/\beta.$$

By combining these lower bounds we get

$$\beta \cdot \mathbb{E}[Q_\tau - m] \geq \max(F(s^\star, \tau), k\delta_\tau) - (m-1)\delta_\tau \geq k \cdot \max(F(s^\star, \tau)/k, \delta_\tau) - (m-1)\delta_\tau. \quad \square$$

3.4.2 Revenue of the discounted vector

In this section, we continue our effort to give a lower bound for $\mathbb{E}[W_{\mathsf{a}}(\mathbf{r}, \tau)]$ as a function of δ_τ and $F(s^\star, \tau)$. Recall that by Claim 3.4.1 we have

$$\mathbb{E}[W_{\mathsf{a}}(\mathbf{r}, \tau)] \geq \max(k.\Pr[Q_\tau > k], \mathbb{E}[\min(P_\tau, k)]).$$

In Lemma 3.4.8 and Lemma 3.4.5 in the previous section, we have obtained lower bounds for both $\mathbb{E}[Q_\tau]$ and $\mathbb{E}[P_\tau]$ as functions of δ_τ and $F(\mathbf{s}^\star, \tau)$. Thus, we proceed to find numeric lower bounds for $k \cdot \Pr[Q_\tau > k]$ and $\mathbb{E}[\min(P_\tau, k)]$ by all possible values of these parameters. To be able to do so, we use the fact that both Q_τ and P_τ are sums of Bernoulli random variables. Based on a sequence of observations about Bernoulli random variables that are mostly presented in Section 3.8 we approximate $\Pr[Q_\tau > k]$ by a function on a set of Bernoulli random variables whose expectation is related to $\mathbb{E}[Q_\tau]$. Later, we use the relation between Binomial and Poisson distributions to get a lower bound that can be computed numerically given fixed values of δ_τ and $F(s^\star, \tau)$. We take a similar but simpler approach to find a lower bound for $\mathbb{E}[\min(P_\tau, k)]$.

Let us define function $G(x, \lambda)$ for a real number $\tau > 0$ and any integer $x \geq 0$, as follows:

$$G(x, \lambda) = 1 - \sum_{i=0}^{x} \frac{\lambda^i e^{-\lambda}}{i!}. \tag{3.8}$$

Note that $G(x, \lambda)$ is the probability with which a random variable drawn from $\text{Pois}(\lambda)$ is greater than x. This function later arises in the lower bound for $\Pr[Q_\tau > k]$ due to the special relation between Poisson and Binomial distribution when the number of trials goes to infinity.

We start by the following lemma about Bernoulli random variables (proved in Section 3.8).

Lemma 3.4.9. *Given $m \in \mathbb{N}$ and a random variable X that is sum of a set of independent Bernoulli random variables with $\mathbb{E}[X] = \mu$, if $m + 1 < \mu$, then we have*

$$\Pr[X > m] \geq \min_{0 \leq i \leq m} G(m - i, \mu - i).$$

For any $0 \le i \le k$, let us define λ_i as follows. We have

$$\lambda_0 = \min\left(\frac{2F(s^\star, \tau)}{k\beta} + \delta_\tau/\beta - 2, \frac{F(s^\star, \tau)}{k\beta}\right),$$

$$\lambda_1 = \min\left(\frac{2F(s^\star, \tau)}{k\beta} + \delta_\tau/\beta - 1, \frac{2F(s^\star, \tau)}{k\beta}\right),$$

and for any any $i \ge 2$,

$$\lambda_i = \frac{iF(s^\star, \tau)}{k\beta} + \delta_\tau/\beta.$$

We are not ready to state our lemma about a lower bound for $\frac{F(s^\star,\tau)}{k\beta}$.

Lemma 3.4.10. *If we have* $\frac{F(s^\star,\tau)}{k\beta} > 1$ *and* $\lambda_i \ge i + 1$ *for any* $i \ge 0$*, then*

$$\Pr[Q_\tau > k] \ge \min_{0 \le m \le k} G(m, \lambda_m).$$

Proof. Let us define $\mathsf{B}_1 = \{\mathsf{b} \in \mathsf{B} : q_\mathsf{b} = 1\}$, $m = |\mathsf{B}_1|$, and $Q_{\tau,2} = \sum_{\mathsf{b} \in \mathsf{B} \setminus \mathsf{B}_1} q_\mathsf{b}$. We have $\mathbb{E}[Q_\tau] = \mathbb{E}[Q_{\tau,2}] + m$ which implies $\Pr[Q_\tau > k] = \Pr[Q_{\tau,2} > k - m]$. Using Lemma 3.4.8, we have the following lower bound for the expected value of random variable $Q_{\tau,2}$:

$$\mathbb{E}[Q_{\tau,2}] \ge \frac{k.\max(F(s^\star, \tau)/k, \delta_\tau) - (m-1)\delta_\tau}{\beta}. \tag{3.9}$$

Further, since $Q_{\tau,2}$ is sum of a set of independent Bernoulli random variables, if $\mathbb{E}[Q_{\tau,2}] \ge k - m + 1$ holds, as an application of Lemma 3.4.9, we get

$$\Pr[Q_{\tau,2} > (k - m)] \ge \min_{0 \le i \le k_1} G(k_1 - i, \mathbb{E}[Q_{\tau,2}] - i),$$

where $k_1 = k - m$. We will later prove that $\mathbb{E}[Q_{\tau,2}] \geq k - m + 1$ holds. Given this equation, to complete the proof it suffices to show that for any $0 \leq i \leq k_1$, we have $G(k_1 - i, E[Q_{\tau,2}] - i) \geq G(k_1 - i, \lambda_{k_1-i})$ since $0 \leq k_1 - i \leq k$. We do this by proving that the following equation holds for any $0 \leq i \leq k_1$:

$$\mathbb{E}[Q_{\tau,2}] - i \geq \lambda_{k_1-i}.$$

Note that proving this also gives us $\mathbb{E}[Q_{\tau,2}] \geq k-m+1$ since we get $\mathbb{E}[Q_{\tau,2}] \geq \lambda_{k-m}$ and by the statement of the lemma, we have $\lambda_{k-m} \geq k - m + 1$. Recall that by definition, for any i with $k_1 - i \geq 1$ we have

$$\lambda_{k_1-i} = \frac{(k_1 - i)F(s^\star, \tau)}{k\beta} + \delta_\tau/\beta.$$

Moreover, using the lower bound provided for $\mathbb{E}[Q_{\tau,2}]$ in Equation 3.9, we get

$$\begin{aligned} \mathbb{E}[Q_{\tau,2}] &\geq \frac{k.\max(F(s^\star, \tau)/k, \delta_\tau) - (m-1)\delta_\tau}{\beta} \\ &\geq \frac{(k-m).F(s^\star, \tau)/k + \delta_\tau}{\beta} \geq \frac{k_1 F(s^\star, \tau)}{k\beta} + \delta_\tau/\beta. \end{aligned}$$

We complete the proof for the case of $k_1 - i > 1$ by invoking $\frac{F(s^\star,\tau)}{k\beta} > 1$ from the statement of lemma. Therefore, to complete the proof it suffices to show that $\mathbb{E}[Q_{\tau,2}] -$

$i \geq \lambda_{k_1 - i}$ holds for $k_1 - i \leq 1$. Using Equation 3.9 and by the fact that $k \geq 2$, we have

$$\mathbb{E}[Q_{\tau,2}] \geq \frac{k.\max(F(s^\star,\tau)/k, \delta_\tau) - m\delta_\tau + \delta_\tau}{\beta} \geq \frac{2F(s^\star,\tau)/k + (k-2-m)\max(F(s^\star,\tau)/k, \delta_\tau) + \delta_\tau}{\beta}.$$

If $m = 0$, then we have $k_1 - 2 = k - 2 - m \geq 0$. Moreover, since we have $F(s^\star,\tau)/(k\beta) > 1$, in the case of $m = 0$, we get

$$\begin{aligned}
\mathbb{E}[Q_{\tau,2}] - i &\geq \frac{2F(s^\star,\tau)/k + \delta_\tau + (k_1 - 2)\max(F(s^\star,\tau)/k, \delta_\tau)}{\beta} - i \\
&\geq \frac{2F(s^\star,\tau)/k + \delta_\tau}{\beta} + k_1 - 2 - i.
\end{aligned}$$

This implies $\mathbb{E}[Q_{\tau,2}] - i \geq \lambda_{k_1 - i}$ for $m = 0$ and $k_1 - i \leq 1$. Now, it remains to show this for $m > 0$ and $k_1 - i \leq 1$ as well. If $m > 0$ we can write the followings:

$$\begin{aligned}
\mathbb{E}[Q_{\tau,2}] &\geq \frac{k.\max(F(s^\star,\tau)/k, \delta_\tau) - (m-1)\delta_\tau}{\beta} \geq \frac{(k-m+1)F(s^\star,\tau)/k}{\beta} \\
&= \frac{(k_1+1)F(s^\star,\tau)/k}{\beta},
\end{aligned}$$

$$\mathbb{E}[Q_{\tau,2}] - i \geq \frac{(k_1 - i + 1)F(s^\star,\tau)/k + i \cdot F(s^\star,\tau)/k}{\beta} - i \geq \frac{(k_1 - i + 1)F(s^\star,\tau)/k}{\beta}.$$

As a result of this for the cases of $(k_1 - i) = 0$ and $(k_1 - i) = 1$ we respectively get $\mathbb{E}[Q_{\tau,2}] - i \geq F(s^\star,\tau)/(k\beta)$ and $\mathbb{E}[Q_{\tau,2}] - i \geq 2F(s^\star,\tau)/(k\beta)$. Knowing that by definition, we have $\lambda_0 \leq F(s^\star,\tau)/(k\beta)$ and $\lambda_1 \leq 2F(s^\star,\tau)/(k\beta)$ hold completes the proof. $\square$

Based on a simple application of Chernoff bound, we show that for any $m \geq 2000$, we have $G(m, 1.05m) \geq 0.9$. (We will prove this as Lemma 3.8.5 in Section 3.8.) Since λ_m is an increasing function of $F(s^\star, \tau)/(k\beta)$, and that $\lambda_m \geq mF(s^\star, \tau)$ holds for any $m \geq 2$, this implies that for $F(s^\star, \tau)/(k\beta) \geq 1.05$, and $m \geq 2000$ we have $G(m, \lambda_m) \geq 0.9$. This gives us

$$\Pr[Q_\tau > k] \geq \min(0.9, \min_{0 \leq m < 2000} G(m, \lambda_m)). \tag{3.10}$$

For smaller values of m; however, giving a desired lower bound for $G(m, \lambda_m)$ is unnecessarily complicated. To avoid the complication of that proof, we instead numerically compute $G(m, \lambda_\tau)$ for different values of $F(s^\star, \tau)/(k\beta)$ and δ_τ/β in Table 3.1. Then, using Lemma 3.4.10 find a lower bound for $\Pr[Q_\tau > k]$ given fixed values for these variables. Each element of Table 3.1, contains value of $\min_{0 \leq m < 2000} G(m, \lambda_m)$ for fixed values of $F(s^\star, \tau)/(k\beta)$ and δ_τ/β. Note that, we ignore an entry of the table by inserting an $-$, if the values associated to $F(s^\star, \tau)$ and δ_τ/β in that entry do not satisfy the necessary conditions of Lemma 3.4.10. We later use this table to complete the proof of Lemma 3.3.3.

$\frac{F(s^\star,\tau)}{k\beta}$ \ δ_τ/β	0.6	0.8	0.9	1
1.05	-	-	0.57	0.59
1.1	-	0.57	0.59	0.62
1.2	0.57	0.62	0.64	0.66
1.5	0.697	0.73	0.746	0.76
1.7	0.76	0.789	0.8	0.814
1.8	0.789	0.8	0.826	0.834

Table 3.1: lower bounds for $\Pr[Q_\tau > k]$ given fixed values of $F(s^\star, \tau)$ and δ/β based on Equation 3.10.

To be able to use the information in this table towards giving a numeric lower bound for $\max(\Pr[Q_\tau > k], \mathbb{E}[\min(Q_\tau, k)]/k)$, we need to construct a similar table for $\mathbb{E}[\min(P_\tau, k)]/k$. To provide the desired lower bound for $\mathbb{E}[\min(P_\tau, k)]/k$ in Lemma 3.4.13, we first need some facts about Bernoulli random variables which are stated in Claim 3.4.12 and Lemma 3.4.11 below. To prevent interruptions to the flow of this section, both proofs are deferred to Section 3.8.

Lemma 3.4.11. *For any integer number $m > 2$ and any real number $\theta \in [0, 2]$, we have*

$$\min_{\boldsymbol{\mu} \in M_{2,\theta}} H(2, \boldsymbol{\mu}) \leq \min_{\boldsymbol{\mu} \in M_{m,\theta}} H(m, \boldsymbol{\mu}),$$

where $M_{m,\theta} = \{\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n) \in [0,1]^n | \sum_{i=1}^n \mu_i = m\theta\}$, *and*

$$H(k, (\mu_1, \ldots, \mu_n)) = \frac{\mathbb{E}[\min(\sum_{i\in[n]} x_i, m)]}{m},$$

with $x_i, \ldots, x_n$ *being independent Bernoulli random variables with means* $\mu_i, \ldots, \mu_n$.

Claim 3.4.12. *Given a fixed real number* $\theta \in (0, 2)$, *and a set of independent Bernoulli random variables* $x_1, \ldots, x_n$ *with* $\mathbb{E}[\sum_{i\in[n]} x_i] = 2\theta$ *we have*

$$\tfrac{1}{2}\,\mathbb{E}[\min(\sum_{i\in[n]} x_i, 2)] \geq 1 - (1+\theta)e^{-2\theta}.$$

Lemma 3.4.13. *For any* $k > 1$, *we have*

$$\mathbb{E}[\min(P_\tau, k)]/k > 1 - (1+\alpha)e^{-2\alpha}, \qquad \text{where} \qquad \alpha = F(s^\star, \tau)/(k\beta) - \delta/\beta. \quad (3.11)$$

Proof. For any real number $\theta \in [0, 2]$ we define set

$$M_{k,\theta} = \{\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n) \in [0,1]^n | \sum_{i=1}^n \mu_i \geq k\theta\},$$

and function

$$H(k, (\mu_1, \ldots, \mu_n)) = \frac{\mathbb{E}[\min(\sum_{i\in[n]} x_i, k)]}{k},$$

where $x_i, \ldots, x_n$ are independent Bernoulli random variables with means $\mu_i, \ldots, \mu_n$.

By Lemma 3.4.11, we know that

$$\min_{\boldsymbol{\mu} \in M_{2,\theta}} H(2, \boldsymbol{\mu}) \leq \min_{\boldsymbol{\mu} \in M_{k,\theta}} H(k, \boldsymbol{\mu}).$$

Note that we have $(\mathbb{E}[p_1], \ldots \mathbb{E}[p_n]) \in M_{k,\alpha}$ since based on Lemma 3.4.5

$$\mathbb{E}[P_\tau] \geq \frac{F(\mathbf{s}^\star, \tau) - k\delta}{\beta} = \alpha k.$$

Moreover, observe that $H(k, (\mathbb{E}[p_1], \ldots \mathbb{E}[p_n])) = \mathbb{E}[\min(P_\tau, k)]/k$, which implies

$$\mathbb{E}[\min(P_\tau, k)]/k > \min_{\boldsymbol{\mu} \in M_{2,\alpha}} G(2, \boldsymbol{\mu}).$$

We complete the proof using Lemma 3.4.12 that states $\min_{\boldsymbol{\mu} \in M_{2,\alpha}} H(2, \boldsymbol{\mu}) \geq 1 - (1 + \alpha)e^{-2\alpha}$. □

We now proceed to construct a similar table for $\mathbb{E}[\min(P_\tau, k)]/k$ based on the lower bound provided in Lemma 3.4.13. Each element of Table 3.2 contains a lower bound for $\mathbb{E}[\min(P_\tau, k)]/k$, given fixed values of $m(s^\star, \tau)/(k\beta)$ and δ_τ/β.

$F(s^\star, \tau)/(k\beta) - \delta_\tau/\beta$	0.15	0.2	0.6	0.7	0.8	0.9	1	1.1	1.2
$\mathbb{E}[\min(P_\tau, k)]/k$	0.14	0.19	0.51	0.58	0.63	0.68	0.72	0.76	0.8

Table 3.2: lower bounds for $\mathbb{E}[\min(P_\tau, k)]/k$ based on different values of $\alpha := F(s^\star, \tau)/(\beta k) - \delta/\beta$. We use the lower bound $\mathbb{E}[min(P_\tau, k)]/k > 1 - (1 + \alpha)e^{-2\alpha}$ obtained in Lemma 3.4.13.

The following lemma is the final piece that we need to complete the proof of

Lemma 3.3.3. In this lemma, we use the constructed tables to show that by setting $\beta = 0.55$ in the rounding algorithm we get

$$F(s^\star, \tau) - \beta \mathbb{E}[W_{\mathsf{a}}(\mathbf{r}, \tau)] \leq 1.05k\beta \leq 0.58.$$

Lemma 3.4.14. *For $\beta = 0.55$ we have $F(s^\star, \tau) - \beta \mathbb{E}[W_{\mathsf{a}}(\mathbf{r}, \tau)] \leq 1.05k\beta$.*

Proof. We start by considering different values of $F(s^\star, \tau)$ and finding a lower bound for $\mathbb{E}[W_{\mathsf{a}}(\mathbf{r}, \tau)]$ based on that. Recall that by Claim 3.4.1, we have

$$\mathbb{E}[W_{\mathsf{a}}(\mathbf{r}, \tau)]/k \geq \max\big(\Pr[Q_\tau > k], \mathbb{E}[\min(P_\tau, k)]/k\big)$$

Further, based on Lemma 3.4.10 and Lemma 3.4.13, we have

$$\max\big(\Pr[Q_\tau > k], \mathbb{E}[\min(P_\tau, k)]/k\big) \geq \max\Bigg(\min_{0 \leq m \leq k} G(m, \lambda_m), 1 - (1+\alpha)^{-2\alpha}\Bigg),$$

where $\alpha := F(s^\star, \tau)/(\beta k) - \delta_\tau/\beta$. It is easy to see that $\min_{0 \leq m \leq k} G(m, \lambda_m)$ is an increasing function of δ_τ/β, while $1 - (1+\alpha)^{-2\alpha}$ is a decreasing function of δ_τ/β; therefore, for any $x \in (0,1)$ we have[2]

$$\max(\Pr[Q_\tau > k], \mathbb{E}[\min(Q_\tau, k)]/k) \geq \min\Bigg(\min_{0 \leq m \leq k} G(m, \lambda_m)_{|\frac{\delta_\tau}{\beta} = x}, (1 - (1+\alpha)^{-2\alpha})_{|\frac{\delta_\tau}{\beta} = x}\Bigg). \tag{3.12}$$

We use this fact to construct Table 3.3 based on Table 3.1 and Table 3.2. To

[2]To clarify the notation, when we use $G(a, b)_{|b=x}$, for a function G, we refer to the value of G given that $b = y$.

do so, for any fixed value of $F(s^\star,\tau)/(k\beta) = y$, we consider four possible values $x \in \{0.6, 0.8, 0.9, 1\}$ for δ_τ/β and rewrite Equation 3.12 as

$$\begin{aligned}
&\max(\Pr[Q_\tau > k], \mathbb{E}[\min(Q_\tau, k)]/k)_{|\frac{F(s^\star,\tau)}{k\beta}=y} \\
&\geq \max_{x\in\{0.6,0.8,0.9,1\}} \min\Big(\min_{0\leq m\leq k} G(m,\lambda_m)_{|\frac{F(s^\star,\tau)}{k\beta}=y,\frac{\delta_\tau}{\beta}=x}, (1-(1+\alpha)^{-2\alpha})_{|F(s^\star,\tau)/(k\beta)=y,\frac{\delta_\tau}{\beta}=x}\Big).
\end{aligned}$$

For any $y \in \{1.05, 1.1, 1.2, 1.5, 1.7, 1.8\}$ and $x \in \{0.6, 0.8, 0.9, 1\}$, we refer to Table 3.1 and Table 3.2 for values of $\min_{0\leq m\leq k} G(m,\lambda_m)_{|F(s^\star,\tau)/(k\beta)=y,\delta_\tau/\beta=x}$ and $(1-(1+\alpha)^{-2\alpha})_{|F(s^\star,\tau)/(k\beta)=y,\delta_\tau/\beta=x}$. (For the sake of constructing this table, if for a pair of x and y, values of these function are not precomputed in Table 3.1 and Table 3.2, we assume that they are equal to zero.)

$F(s^\star,\tau)/(k\beta)$	1.05	1.1	1.2	1.5	1.7	1.8
$\max(\Pr[Q_\tau > k], \mathbb{E}[\min(Q_\tau, k)]/k)$	0.14	0.19	0.51	0.68	0.76	0.789

Table 3.3: lower bounds for $\mathbb{E}[W_{\mathsf{a}}(\mathbf{r},\tau)]/k$ using Equation 3.12, given fixed values of $F(s^\star,\tau)/(k\beta)$ achieved from combining Table 3.1 and Table 3.2.

As an instance, the first column of the table is obtained as follows. For $\delta_\tau/\beta = 0.9$, and $F(s^\star,\tau)/(k\beta) = 1.05$ we have the followings respectively based on Table 3.1 and Table 3.2:

$$\min_{0\leq m\leq k} G(m,\lambda_m)_{|\frac{F(s^\star,\tau)}{k\beta}=1.05,\delta_\tau/\beta=0.9} = 0.57,$$

$$\big(1-(1+\alpha)^{-2\alpha}\big)_{|\frac{F(s^\star,\tau)}{k\beta}=1.05,\delta_\tau/\beta=0.9} = 0.14.$$

As a result, we have

$$\max(\Pr[Q_\tau > k], \mathbb{E}[\min(Q_\tau, k)]/k)_{|\frac{F(s^\star,\tau)}{k\beta}=1.05}$$
$$\geq \max_{x\in\{0.8,0.9,1\}} \min\left(\min_{0\leq m\leq k} G(m,\lambda_m)_{|\frac{F(s^\star,\tau)}{k\beta}=1.05,\frac{\delta_\tau}{\beta}=0.9}, (1-(1+\alpha)^{-2\alpha})_{|\frac{F(s^\star,\tau)}{k\beta}=1.05,\frac{\delta_\tau}{\beta}=0.9}\right)$$
$$\geq \min(0.57, 0.14) = 0.14.$$

We now proceed to complete the proof using Table 3.3. Observe that the lower bound in Equation 3.12 is a non-decreasing function of $F(s^\star,\tau)/(k\beta)$. This implies that for any $0 \leq x < y \leq 1$, we have $\mathbb{E}[W_{\mathsf{a}}(\mathbf{r},\tau)|\frac{F(s^\star,\tau)}{k\beta} = x] \leq \mathbb{E}[W_{\mathsf{a}}(\mathbf{r},\tau)|\frac{F(s^\star,\tau)}{k\beta} \in [x,y]]$. Having this, we complete the proof using a case by case analysis and proving that

$$F(s^\star,\tau) - \beta\mathbb{E}[W_{\mathsf{a}}(\mathbf{r},\tau)] \leq 1.05k\beta$$

holds for all possible values of $F(s^\star,\tau)/(k\beta)$.

- For $F(s^\star,\tau)/(k\beta) \in [0, 1.05]$, it is obvious that $F(s^\star,\tau) - \beta\mathbb{E}[W_{\mathsf{a}}(\mathbf{r},\tau)] \leq 1.05k\beta$ holds.

- If $F(s^\star,\tau)/(k\beta) \in [1.05, 1.1]$, then we get $F(s^\star,\tau) - \beta\mathbb{E}[W_{\mathsf{a}}(\mathbf{r},\tau)] \leq (1.1 - 0.14)k\beta = 0.96k\beta$ since in this case we have $\mathbb{E}[W_{\mathsf{a}}(\mathbf{r},\tau)]/k \geq 0.14$ based on Table 3.3.

- If $F(s^\star,\tau)/(k\beta) \in [1.1, 1.2]$, then we get $F(s^\star,\tau) - \beta\mathbb{E}[W_{\mathsf{a}}(\mathbf{r},\tau)] \leq (1.2 - 0.19)k\beta = 1.01k\beta$ since in this case we have $\mathbb{E}[W_{\mathsf{a}}(\mathbf{r},\tau)]/k \geq 0.19$.

- If $F(s^\star,\tau)/(k\beta) \in [1.2, 1.5]$, then we get $F(s^\star,\tau) - \beta\mathbb{E}[W_{\mathsf{a}}(\mathbf{r},\tau)] \leq (1.5 - 0.51)k\beta =$

$0.99k\beta$ since in this case we have $\mathbb{E}[W_{\mathsf{a}}(\mathbf{r},\tau)]/k \geq 0.51$.

- If $F(s^\star,\tau)/(k\beta) \in [1.5, 1.7]$, then we get $F(s^\star,\tau) - \beta\mathbb{E}[W_{\mathsf{a}}(\mathbf{r},\tau)] \leq (1.7-0.68)k\beta = 1.02k\beta$ since in this case we have $\mathbb{E}[W_{\mathsf{a}}(\mathbf{r},\tau)]/k \geq 0.68$.

- If $F(s^\star,\tau)/(k\beta) \in [1.7, 1.8]$, then we get $F(s^\star,\tau) - \beta\mathbb{E}[W_{\mathsf{a}}(\mathbf{r},\tau)] \leq (1.8-0.76)k\beta = 1.04k\beta$ since in this case we have $\mathbb{E}[W_{\mathsf{a}}(\mathbf{r},\tau)]/k \geq 0.76$.

- If $F(s^\star,\tau)/(k\beta) \in [1.8, 1/\beta]$, then we get $F(s^\star,\tau) - \beta\mathbb{E}[W_{\mathsf{a}}(\mathbf{r},\tau)] \leq (1.8 - 0.76)k\beta = 1.04k\beta$ since in this case we have $\mathbb{E}[W_{\mathsf{a}}(\mathbf{r},\tau)]/k \geq 0.76$.

The proof is concluded since we know $F(s^\star,\tau)/(k\beta) \leq 1/\beta$. This is due to the definition of $F(s^\star,\tau)$ which is

$$F(\mathbf{s}^\star,\tau) = \sum_{\substack{p: p\in\mathcal{S}_{\mathsf{a}},\\ \mathsf{Rev}_{\mathsf{a}}(p)\geq\tau}} s^\star_{\mathsf{a},p} - (1-\beta)\mathbb{E}[W_{\mathsf{a}}(\mathbf{r}',\tau)],$$

and the fact that by constraints of the LP, we have $\sum_{p:p\in\mathcal{S}_{\mathsf{a}}} s^\star_{\mathsf{a},p} \leq k$. □

3.5 Upper Bound for the "Simple Rounding" Approach

The algorithm designed for the single unit case of the problem in the previous chapter directly uses the fraction solution of the LP as a probability distribution over reserve prices, and for each buyer independently draws a reserve price from that. This is equivalent to setting $\beta = 0$ in our rounding algorithm. In the rest of the section, we use *simple rounding* to refer to this rounding technique. We show using an example that for a large enough k (number of items) the approximation factor of this algorithm is at most $0.5 + \varepsilon$ for any small constant ε.

Lemma 3.5.1. *Given any constant $\varepsilon > 0$, there exists a dataset of bids for which the simple rounding approach achieves at most $0.5 + \varepsilon$ fraction of the optimal revenue.*

Proof. We use Table 3.4 to represent our dataset of auctions. Let k denote the number of items. In our example, we have $2k + 2$ auctions and $k + 2$ buyers represented by $b_1, \ldots, b_{k+2}$. Each column in the table represents an auction and its weight is the number of times that this auction is repeated in the dataset. To put it differently, we can simply assume that there are only four *weighted* auctions represented by the four columns and each one has a revenue equal to the total payment of buyers in the auction multiplied by its weight. Further, note that buyers $b_3, \ldots, b_{k+2}$ have similar bids in all auctions.

Buyer \ Weight	1	1	k	k
b_1	k^3	0	0	0
b_2	0	0	k	1
$\mathsf{b}_3, \ldots, \mathsf{b}_{k+2}$	0	k^2	k	1

Table 3.4: A Bad Example

We observe that there are two different optimal vectors of reserve prices for this dataset which are $(k^3, k, k^2, \ldots, k^2)$ and $(k^3, 1, 1, \ldots, 1)$. By applying the first reserves, the amounts of revenue that we get from our four auctions (the four columns) are respectively k^3, k^3, k^2, and 0, while the second vector gives us k^3, 0, k^3, and k^2 amounts of revenue respectively. Thus, the optimal revenue adds up to $2k^3 + k^2$. Further, let us mention that the all-zero vector of reserve prices simply gives us a revenue of $k^3 + k^2$, and as a result the approximation factor of this solution is at most $0.5 + \varepsilon$ for a large enough k and any constant $\varepsilon > 0$. Given a small constant $\varepsilon > 0$, we construct a fractional solution of the LP which if rounded using the simple rounding procedure, results in a less than $0.5+\varepsilon$ approximate solution for large values of k. Let $\delta = \varepsilon/4$, and consider a solution of the LP that assigns δ probability to the first optimal solution and $1 - \delta$ probability to the second one. A formal representation of this solution is given below. Note that for simplicity, instead of sub-profiles, we use profiles to represent our solution. (See Definition 5.4.1.) Also, recall that $\hat{b_1}, \ldots, \hat{b_{k+1}}$ are the auxiliary buyers who bid 0 in all the auctions.

- $s_{\mathsf{a}_1,p_1} = 1$ for $p_1 = (b_1, \hat{b_1}, \ldots, \hat{b_k}, k^3, 0, \ldots, 0)$.
- $s_{\mathsf{a}_2,p_2} = \delta$ for $p_2 = (b_3, \ldots, b_{k+2}, \hat{b_1}, , k^2, \ldots, k^2, 0)$.
- $s_{\mathsf{a}_2,p'_2} = 1 - \delta$ for $p'_2 = (b_3, \ldots, b_{k+2}, \hat{b_1}, 1, \ldots, 1, 0)$.
- $s_{\mathsf{a}_3,p_3} = \delta$ for $p_3 = (b_2, \hat{b_1}, \ldots, \hat{b_k}, k, 0, \ldots, 0)$.
- $s_{\mathsf{a}_3,p'_3} = 1 - \delta$ for $p'_3 = (b_2, b_3, \ldots, b_{k+2}, 1, \ldots, 1)$.
- $s_{\mathsf{a}_4,p_4} = 1 - \delta$ for $p_4 = (b_2, b_3, \ldots, b_{k+2}, 1, \ldots, 1)$.
- $s_{\mathsf{a}_4,p'_4} = \delta$ for $p'_4 = (\hat{b}_1, b_3, \ldots, \hat{b}_{k+1}, 0, \ldots, 0)$.
- $x_{b_1,k^3} = 1$.
- $x_{b_2,k} = \delta$, and $x_{b_2,1} = 1 - \delta$.

- $x_{b,k^2} = \delta$, and $x_{b,1} = 1 - \delta$ for any $b \in \{b_3, b_{k+2}\}$.

It is easy to verify that this fractional solution satisfies the constraints of the LP and is an optimal fractional solution; therefore, it suffices to show that the vector of reserve prices obtained from the simple rounding algorithm gives us at most $(0.5 + \varepsilon)(2k^3 + k^2)$ revenue for a large enough k. As mentioned before, the simple rounding algorithm directly uses the fractional solution of the LP as a probability distribution to randomly pick the reserve price of any buyer. More precisely, for any buyer b any reserve price r is chosen with probability $x_{b,r}$ independently from other buyers. Let $\mathbf{s}$ denote the vector of reserve prices obtained from rounding our fractional solution using this simple rounding technique. Below we investigate the revenue obtained from each auction by applying $\mathbf{s}$.

1. From the first auction we simply get revenue of k^3 as reserve price k^3 is chosen for buyer b_1 with probability one.

2. In the second auction there are k buyers with nonzero bids, thus all the buyers get an item and pay their reserve prices. This implies that the expected revenue of this auction is $k(\delta k^2 + (1 - \delta))$ since the simple rounding algorithm chooses reserve prices of k^2 and 1 for buyers $b_3, \ldots, b_{k+2}$ with probabilities δ and $(1 - \delta)$ respectively.

3. In the third auction, however, there are $k+1$ buyers $b_2, \ldots b_{k+2}$ with bid k; therefore, we get a revenue of k^3 if all the bids are cleared. Otherwise, revenue of this auction is the sum of reserve prices of the buyers whose bid is cleared multiplied by the weight of the auction (which is k). For any buyer $b \in \{b_3, \ldots b_{k+2}\}$

the simple rounding technique chooses reserve price k^2 with probability δ and reserve price 1 with probability $1-\delta$. Moreover, buyer b_2 gets reserve prices k and 1 respectively with probabilities δ and $1-\delta$. This implies that the revenue obtained from this auction is k^3 with probability $(1-\delta)^k$ and it is less than $2k^2$ with probability $1-(1-\delta)^k$. Note that for any constant value of δ, we have $\lim_{k\to\infty}(1-\delta)^k = 0$; thus, for a large enough k, the expected revenue obtained from this auction is at most $2k^2+\delta$.

4. Since the maximum bid in the fourth auction is one and its weight is k, the total revenue obtained from this auction is at most k^2.

By summing up the revenue obtained from the four auctions, we conclude that for a large enough k, the expected total revenue is upper bounded by $(1+\delta)k^3+3k^2+k$. Since $\delta = \varepsilon/4$, for a large enough k we have $(1+\delta)k^3+3k^2+k < (1+\varepsilon/2)k^3$. Recall that the optimal vector of reserve prices in this example gives us a revenue of $2k^3+k^2$. This implies that the approximation factor of the simple rounding algorithm is upper bounded by $0.5+\varepsilon$ for a large enough k. □

3.6 Omitted Proofs of Section 3.4.1

In this section, our goal is to prove Lemma 3.4.4, Lemma 3.4.6, and Lemma 3.4.7 which are stated in Section 3.4.1. Before going into their proofs, however, we start by a series of claims that are needed for completing the proofs.

Claim 3.6.1. *If the following inequality holds, then* $\mathbb{E}[W_{\mathsf{a}}(\mathbf{r}, \tau)] = k$*.*

$$\mathbb{E}[W_{\mathsf{a}}(\mathbf{r}', \tau)] < \sum_{\mathsf{b} \in \mathsf{B}} \Pr[\beta_{\mathsf{a},\mathsf{b}} \geq r'_{\mathsf{b}} \geq \tau] \tag{3.13}$$

Proof. We have

$$\mathbb{E}[W_{\mathsf{a}}(\mathbf{r}', \tau)] = \min\left(\sum_{\mathsf{b} \in \mathsf{B}} \Pr[\beta_{\mathsf{a},\mathsf{b}} \geq r'_{\mathsf{b}} \geq \tau], k\right).$$

Define $\mathsf{B}' = \{\mathsf{b} \in \mathsf{B} : \Pr[\beta_{\mathsf{a},\mathsf{b}} \geq r'_{\mathsf{b}} \geq \tau] \neq 0\}$. Observe that if $|\mathsf{B}'| \leq k$ then

$$\sum_{\mathsf{b} \in \mathsf{B}} \Pr[\beta_{\mathsf{a},\mathsf{b}} \geq r'_{\mathsf{b}} \geq \tau] \leq k,$$

which implies

$$\mathbb{E}[W_{\mathsf{a}}(\mathbf{r}', \tau)] = \sum_{\mathsf{b} \in \mathsf{B}} \Pr[\beta_{\mathsf{a},\mathsf{b}} \geq r'_{\mathsf{b}} \geq \tau].$$

Thus, if Equation 3.13 holds then $|\mathsf{B}'| > k$. By Claim 3.6.5, provided in the appendix, for any $\mathsf{b} \in \mathsf{B}'$ we have $\mathbb{E}[q_{\mathsf{b}}] = 1$. This gives us $\Pr[\sum_{\mathsf{b} \in \mathsf{B}'} q_{\mathsf{b}} > k] = 1$ and as a result $\Pr[Q_\tau > k] = 1$. Finally, Claim 3.4.1 gives us $\mathbb{E}[W_{\mathsf{a}}(\mathbf{r}, \tau)] \geq k \Pr[Q_\tau > k]$, and concludes the proof. □

Assumption 3.6.2. *The following equation holds for any auction* $\mathsf{a} \in \mathsf{A}$*.*

$$\mathbb{E}[W_{\mathsf{a}}(\mathbf{r}', \tau)] = \sum_{\mathsf{b} \in \mathsf{B}} \Pr[\beta_{\mathsf{a},\mathsf{b}} \geq r'_{\mathsf{b}} \geq \tau].$$

Proof. As a corollary of Claim 3.6.1, if

$$\mathbb{E}[W_{\mathsf{a}}(\mathbf{r}', \tau)] \leq \sum_{\mathsf{b} \in \mathsf{B}} \Pr[\beta_{\mathsf{a},\mathsf{b}} \geq r'_{\mathsf{b}} \geq \tau],$$

then Equation 3.5 is simply satisfied for $\beta = 0.55$ and $c = 0.58$ since in this case $\beta\mathbb{E}[W_{\mathsf{a}}(\mathbf{r}, \tau)] = \beta k$. Moreover, based on LP we have

$$\sum_{\substack{p: p \in \mathcal{P}_{\mathsf{a}} \\ \mathsf{Rev}_{\mathsf{a}}(p) \geq \tau}} s^{\star}_{\mathsf{a},p} \leq k,$$

which results in

$$\sum_{\substack{p: p \in \mathcal{P}_{\mathsf{a}} \\ \mathsf{Rev}_{\mathsf{a}}(p) \geq \tau}} s^{\star}_{\mathsf{a},p} - (1-\beta)\mathbb{E}[W_{\mathsf{a}}(\mathbf{r}', \tau)] - \beta\mathbb{E}[W_{\mathsf{a}}(\mathbf{r}, \tau)] \leq (1-\beta)k = 0.45k < 0.58k.$$

Therefore, to complete the proof of Lemma 3.3.3, we make the following assumption and only focus on proving Equation 3.5 for auctions that do not satisfy the mentioned condition. □

Claim 3.6.3. *For any buyer* $\mathsf{b} \in \mathsf{B}_{\tau}$ *the following holds.*

$$\bigcup_{\mathsf{b}_1 \in \mathsf{B}_{\tau} \setminus \{\mathsf{b}\}} (\mathcal{Q}_{\mathsf{b},\mathsf{a}} \cap \mathcal{Q}'_{\mathsf{b}_1,\mathsf{a}}) = \mathcal{Q}_{\mathsf{b},\mathsf{a}} \cap \mathcal{L}_{\tau}.$$

Proof. Consider a valid profile a valid sub-profile $p = (\mathsf{b}', \mathsf{b}'', r', r'') \in \mathcal{S}_{\mathsf{a}}$. By definition, for any buyer $\mathsf{b}_1 \in \mathsf{B}_{\tau}$ we have $p \in (\mathcal{Q}_{\mathsf{b},\mathsf{a}} \cap \mathcal{Q}'_{\mathsf{b}_1,\mathsf{a}})$ iff $\mathsf{b}' = \mathsf{b}$ and $\mathsf{b}'' = \mathsf{b}_1$. Moreover, due to p being a valid sub-profile, it satisfies $\mathsf{b}' \neq \mathsf{b}''$. As a result we have $p \in$

$\bigcup_{\mathsf{b}_1 \in \mathsf{B}_\tau \setminus \{\mathsf{b}\}} (\mathcal{Q}_{\mathsf{b},\mathsf{a}} \cap \mathcal{Q}'_{\mathsf{b}_1,\mathsf{a}})$ iff $\mathsf{b}' = \mathsf{b}$ and $\mathsf{b}'' \in \mathsf{B}_\tau$. This completes our proof since this is equal to definition of $\mathcal{Q}_{\mathsf{b},\mathsf{a}} \cap \mathcal{L}_\tau$. □

Claim 3.6.4. *For any buyer $\mathsf{b} \in \mathsf{B}_\tau$ we have*

$$\sum_{p \in \mathcal{L}_\tau \cap \mathcal{Q}_{\mathsf{b}}} s^\star_{\mathsf{a},p} + \sum_{p \in \mathcal{L}_\tau \cap \mathcal{Q}'_{\mathsf{b}}} s^\star_{\mathsf{a},p}/k \leq \delta_\tau.$$

Proof. As a constraint of the LP we have the following for b and any other buyer $\mathsf{b}_1 \in \mathsf{B}$.

$$\sum_{p \in \mathcal{Q}_{\mathsf{b},\mathsf{a}} \cap \mathcal{Q}'_{\mathsf{b}_1,\mathsf{a}}} s_{\mathsf{a},p} \leq \sum_{r \in \mathsf{R}} y'_{\mathsf{b}_1,r,\mathsf{a}}.$$

By summing both sides over all buyers in $\mathsf{B}_\tau \setminus \{\mathsf{b}\}$ we get

$$\sum_{\mathsf{b}_1 \in \mathsf{B}_\tau \setminus \{\mathsf{b}\}} \sum_{\substack{p \in \mathcal{Q}_{\mathsf{b},\mathsf{a}} \\ \cap \mathcal{Q}'_{\mathsf{b}_1,\mathsf{a}}}} s_{\mathsf{a},p} \leq \sum_{\mathsf{b}_1 \in \mathsf{B}_\tau \setminus \{\mathsf{b}\}} \sum_{r \in \mathsf{R}} y'_{\mathsf{b}_1,r,\mathsf{a}}.$$

Observe that by Claim 3.6.3 we have

$$\bigcup_{\mathsf{b}_1 \in \mathsf{B}_\tau \setminus \{\mathsf{b}\}} (\mathcal{Q}_{\mathsf{b},\mathsf{a}} \cap \mathcal{Q}'_{\mathsf{b}_1,\mathsf{a}}) = \mathcal{Q}_{\mathsf{b},\mathsf{a}} \cap \mathcal{L}_\tau,$$

which results in

$$\sum_{p \in \mathcal{L}_\tau \cap \mathcal{Q}_{\mathsf{b},\mathsf{a}}} s_{\mathsf{a},p} \leq \sum_{\mathsf{b}_1 \in \mathsf{B}_\tau \setminus \{\mathsf{b}\}} \sum_{r \in \mathsf{R}} y'_{\mathsf{b}_1,r,\mathsf{a}}. \tag{3.14}$$

Moreover, as the second constraint of the LP we have

$$y'_{\mathsf{b},r,\mathsf{a}} = \sum_{p \in \mathcal{Q}'_{\mathsf{b},r,\mathsf{a}}} s_{\mathsf{a},p}/k.$$

By summing up both sides of this equation over all reserve prices in R, we have

$$\sum_{r \in \mathsf{R}} y'_{\mathsf{b},r,\mathsf{a}} = \sum_{r \in \mathsf{R}} \sum_{p \in \mathcal{Q}'_{b,r,\mathsf{a}}} s_{\mathsf{a},p}/k - \sum_{p \in \mathcal{Q}'_{\mathsf{b},\mathsf{a}}} s_{\mathsf{a},p}/k. \tag{3.15}$$

Combining Equation 3.14 and Equation 3.15 yields

$$\sum_{p \in \mathcal{L}_\tau \cap \mathcal{Q}_{\mathsf{b},\mathsf{a}}} s_{\mathsf{a},p} \leq \sum_{\mathsf{b}_1 \in \mathsf{B}_\tau} \sum_{p \in \mathcal{Q}'_{\mathsf{b}_1,\mathsf{a}}} s_{\mathsf{a},p}/k - \sum_{p \in \mathcal{Q}'_{\mathsf{b},\mathsf{a}}} s_{\mathsf{a},p}/k. \tag{3.16}$$

We further use the fact that for any buyer $\mathsf{b}_2 \in \mathsf{B}_\tau$ and any sub-profile $p = (\mathsf{b}', \mathsf{b}_2, r', r) \in \mathcal{Q}'_{\mathsf{b}_2,\mathsf{a}}$ we have $p \in \mathcal{L}_\tau$. Recall that $\mathcal{L}_\tau$ is defined as

$$\mathcal{L}_\tau := \{(\mathsf{b}', \mathsf{b}, r', r) \in \mathcal{T}_\tau |\, \mathsf{b} \in \mathsf{B}_\tau\}.$$

We complete the proof using the definition of δ_τ which is $\delta_\tau := \sum_{\mathsf{b} \in \mathsf{B}_\tau} \sum_{p \in \mathcal{L}_\tau \cap \mathcal{Q}'_{\mathsf{b}}} s^\star_{\mathsf{a},p}/k$. This gives us the followings.

$$\sum_{p \in \mathcal{L}_\tau \cap \mathcal{Q}_{\mathsf{b},\mathsf{a}}} s_{\mathsf{a},p} + \sum_{p \in \mathcal{L}_\tau \cap \mathcal{Q}'_{\mathsf{b},\mathsf{a}}} s_{\mathsf{a},p}/k \leq \sum_{\mathsf{b}_1 \in \mathsf{B}_\tau} \sum_{p \in \mathcal{L}_\tau \cap \mathcal{Q}'_{\mathsf{b}_1,\mathsf{a}}} s_{\mathsf{a},p}/k \leq \delta_\tau. \tag{3.17}$$

□

Claim 3.6.5. *For any buyer* $\mathsf{b} \in \mathsf{B}_\tau$ *with* $\Pr[r'_\mathsf{b} \leq \beta_{\mathsf{a},\mathsf{b}}] \neq 0$ *we have* $\mathbb{E}[q_\mathsf{b}] = 1$.

Proof. By construction of $\mathbf{r}'$ we have

$$\Pr[\beta_{\mathsf{a},\mathsf{b}} \geq r'_{\mathsf{b}}] = \sum_{\substack{r: r \leq \beta_{\mathsf{a},\mathsf{b}}, \\ r \geq t_b}} x_{\mathsf{b},r}/(1-\beta).$$

As a result, if $\Pr[\beta_{\mathsf{a},\mathsf{b}} \geq r'_{\mathsf{b}}] \neq 0$ then $t_{\mathsf{b}} \leq \beta_{\mathsf{a},\mathsf{b}}$. Recall that by definition of t_{b}, we have

$$\sum_{r < t_b} x_{\mathsf{b},r} = \beta.$$

Further by construction of $\mathbf{r}$ we have

$$\mathbb{E}[q_b] = \Pr[r_{\mathsf{b}} \leq \beta_{\mathsf{a},\mathsf{b}}] = \sum_{\substack{r: r \leq \beta_{\mathsf{a},\mathsf{b}}, \\ r < t_b}} x_{\mathsf{b},r}/\beta.$$

Given that $t_{\mathsf{b}} \leq \beta_{\mathsf{a},\mathsf{b}}$ we get

$$\mathbb{E}[q_b] = \Pr[r_{\mathsf{b}} \leq \beta_{\mathsf{a},\mathsf{b}}] = \sum_{r < t_b} x_{\mathsf{b},r}/\beta = 1.$$

This concludes our proof. $\square$

Claim 3.6.6.

$$(1-\beta) \sum_{\mathsf{b} \in \mathsf{B}} \Pr[\beta_{\mathsf{a},\mathsf{b}} \geq r'_b \geq \tau] \geq \sum_{p \in \mathcal{J}_\tau^+} s^\star_{\mathsf{a},p}.$$

Proof. By definition of $\mathbf{r}'$

$$(1-\beta)\sum_{\mathsf{b}\in\mathsf{B}}\Pr[\beta_{\mathsf{a},\mathsf{b}}\geq r'_b\geq\tau]=(1-\beta)\sum_{\mathsf{b}\in\mathsf{B}}\sum_{\substack{r:\beta_{\mathsf{a},\mathsf{b}}\leq r,\\ r\leq\tau}}f_{\mathsf{b},r}$$
$$=(1-\beta)\sum_{\mathsf{b}\in\mathsf{B}}\sum_{\substack{r:r\geq t_{\mathsf{b}},\\ r\in[\tau,\beta_{\mathsf{a},\mathsf{b}}]}}\frac{x_{\mathsf{b},r}}{1-\beta}$$
$$=\sum_{\mathsf{b}\in\mathsf{B}}\sum_{\substack{r:r\geq t_{\mathsf{b}},\\ r\in[\tau,\beta_{\mathsf{a},\mathsf{b}}]}}x_{\mathsf{b},r}. \tag{3.18}$$

Recall definition $\mathcal{J}_\tau^+ := \{p=(\mathsf{b},\mathsf{b}',r,r')\in\mathcal{T}_\tau|\,\mathsf{b}'\notin\mathsf{B}_\tau,\ r\geq t_{\mathsf{b}'}\}$, for which we have

$$\mathcal{J}_\tau^+\subset\{p=(\mathsf{b},\mathsf{b}',r,r')\in\mathcal{S}_{\mathsf{a}}|\,r\geq\tau,\ r\geq t_{\mathsf{b}}\}.$$

This is because combination of $p\in\mathcal{T}_\tau$ and $\mathsf{b}'\notin\mathsf{B}_\tau$ implies that revenue of the sub-profile is greater than or equal to τ while bid of the supporting buyer is smaller than τ which yields $r\geq\tau$. Moreover, due to validity of any sub-profile in $\mathcal{J}_\tau^+$ we have $\beta_{\mathsf{a},\mathsf{b}}\geq r$. Using the first constraint of the LP, we get

$$\sum_{p\in\mathcal{J}_\tau^+}s^\star_{\mathsf{a},p}=\sum_{\mathsf{b}\in\mathsf{B}}\sum_{\substack{p\in(\mathcal{J}_\tau^+\\ \cap\mathcal{Q}_{\mathsf{b},r})}}s^\star_{\mathsf{a},p}\leq\sum_{\mathsf{b}\in\mathsf{B}}\sum_{\substack{r:r\geq t_{\mathsf{b}},\\ r\in[\tau,\beta_{\mathsf{a},\mathsf{b}}]}}y_{\mathsf{b},r,\mathsf{a}}.$$

Moreover, by the third constraint, we have

$$\sum_{\substack{r:r\geq t_{\mathsf{b}},\\ r\in[\tau,\beta_{\mathsf{a},\mathsf{b}}]}}y_{\mathsf{b},r,\mathsf{a}}\leq\sum_{\substack{r:r\geq t_{\mathsf{b}},\\ r\in[\tau,\beta_{\mathsf{a},\mathsf{b}}]}}x_{\mathsf{b},r}.$$

Evoking Equation 3.18, we obtain

$$\sum_{p\in\mathcal{J}_\tau^+} s^\star_{\mathsf{a},p} \le \sum_{\mathsf{b}\in\mathsf{B}} \sum_{\substack{r:r\ge t_\mathsf{b},\\ r\in[\tau,\beta_{\mathsf{a},\mathsf{b}}]}} x_{\mathsf{b},r} = (1-\beta)\sum_{\mathsf{b}\in\mathsf{B}} \Pr[\beta_{\mathsf{a},\mathsf{b}} \ge r'_b \ge \tau];$$

thus, the proof is completed. □

Claim 3.6.7. *The following equation holds for any buyer* b*.*

$$\sum_{p\in\mathcal{T}_\tau\cap\mathcal{Q}_{\mathsf{b},\mathsf{a}}} s^\star_{\mathsf{a},p} + \sum_{p\in\mathcal{L}_\tau\cap\mathcal{Q}'_{\mathsf{b},\mathsf{a}}} s^\star_{\mathsf{a},p}/k - (1-\beta)\Pr[\beta_{\mathsf{a},\mathsf{b}} \ge r'_\mathsf{b} \ge \tau] \le \max(\beta,\delta_\tau). \tag{3.19}$$

Proof. We consider two cases of $t_b \ge \tau$ and $t_b < \tau$ and prove the lemma for them seperately. We show that if $t_b \ge \tau$ then the left hand side of Equation 3.19 is uppe bounded by β, and for the second case we show that it is upper bounded by δ_τ. We have

$$\sum_{p\in\mathcal{T}_\tau\cap\mathcal{Q}_{\mathsf{b},\mathsf{a}}} s^\star_{\mathsf{a},p} + \sum_{p\in\mathcal{L}_\tau\cap\mathcal{Q}'_{\mathsf{b},\mathsf{a}}} s^\star_{\mathsf{a},p}/k \le \sum_{p\in\mathcal{Q}_{\mathsf{b},\mathsf{a}}} s^\star_{\mathsf{a},p} + \sum_{p\in\mathcal{Q}'_{\mathsf{b},\mathsf{a}}} s^\star_{\mathsf{a},p}/k = \sum_{r\le\beta_{\mathsf{a},\mathsf{b}}} \sum_{p\in\mathcal{Q}_{\mathsf{b},r,\mathsf{a}}} s^\star_{\mathsf{a},p} + \sum_{r\le\beta_{\mathsf{a},\mathsf{b}}} \sum_{p\in\mathcal{Q}'_{\mathsf{b},r,\mathsf{a}}} s^\star_{\mathsf{a},p}/k.$$

The right hand side is due to the fact that any sub-profile in $\mathcal{Q}_{\mathsf{b},\mathsf{a}}$ or $\mathcal{Q}'_{\mathsf{b},\mathsf{a}}$ is a valid profile of auction a which implies $\mathcal{Q}_{\mathsf{b},\mathsf{a}} = \bigcup_{r\le\beta_{\mathsf{a},\mathsf{b}}} \mathcal{Q}'_{\mathsf{b},r,\mathsf{a}}$ and $\mathcal{Q}_{\mathsf{b},\mathsf{a}} = \bigcup_{r\le\beta_{\mathsf{a},\mathsf{b}}} \mathcal{Q}'_{\mathsf{b},r,\mathsf{a}}$. Moreover, based on the first three constraints of the LP for any $r \le \beta_{\mathsf{a},\mathsf{b}}$ we have

$$y_{\mathsf{b},r,\mathsf{a}} = \sum_{p\in\mathcal{Q}_{\mathsf{b},r,\mathsf{a}}} s^\star_{\mathsf{a},p}, \quad y'_{\mathsf{b},r,\mathsf{a}} = \sum_{p\in\mathcal{Q}'_{\mathsf{b},r,\mathsf{a}}} s^\star_{\mathsf{a},p,\mathsf{a}}/k, \text{and} \quad y'_{\mathsf{b},r,\mathsf{a}} + y_{\mathsf{b},r,\mathsf{a}} \le x_{\mathsf{b},r}.$$

which implies

$$\sum_{r\le\beta_{\mathsf{a},\mathsf{b}}}\sum_{p\in\mathcal{Q}_{\mathsf{b},r,\mathsf{a}}} s^{\star}_{\mathsf{a},p} + \sum_{r\le\beta_{\mathsf{a},\mathsf{b}}}\sum_{p\in\mathcal{Q'}_{\mathsf{b},r,\mathsf{a}}} s^{\star}_{\mathsf{a},p}/k \le \sum_{r\le\beta_{\mathsf{a},\mathsf{b}}} y_{\mathsf{b},r,\mathsf{a}} + \sum_{r\le\beta_{\mathsf{a},\mathsf{b}}} y'_{\mathsf{b},r,\mathsf{a}} \le \sum_{r\le\beta_{\mathsf{a},\mathsf{b}}} x_{\mathsf{b},r}. \tag{3.20}$$

Moreover, based on the construction of $\mathbf{r}'$ we have

$$(1-\beta)\Pr[\beta_{\mathsf{a},\mathsf{b}} \ge r'_{\mathsf{b}} \ge \tau] = (1-\beta)\sum_{r\in[\tau,\beta_{\mathsf{a},\mathsf{b}}]} f'_{\mathsf{b},r} = (1-\beta)\sum_{\substack{r:r\in[\tau,\beta_{\mathsf{a},\mathsf{b}}],\\ r\ge t_{\mathsf{b}}}} x_{\mathsf{b},r}/(1-\beta). \tag{3.21}$$

If $t_{\mathsf{b}} \ge \tau$, combining Equation 3.20 and Equation 3.21, gives us

$$\sum_{p\in\mathcal{T}_\tau\cap\mathcal{Q}_{\mathsf{b},\mathsf{a}}} s^{\star}_{\mathsf{a},p} + \sum_{p\in\mathcal{L}_\tau\cap\mathcal{Q'}_{\mathsf{b},\mathsf{a}}} s^{\star}_{\mathsf{a},p}/k - (1-\beta)\Pr[\beta_{\mathsf{a},\mathsf{b}} \ge r'_{\mathsf{b}} \ge \tau] \le \sum_{r\le\beta_{\mathsf{a},\mathsf{b}}} x_{\mathsf{b},r} - \sum_{r\in\left[t_{\mathsf{b}},\beta_{\mathsf{a},\mathsf{b}}\right]} x_{\mathsf{b},r} = \sum_{r<t_{\mathsf{b}}} x_{\mathsf{b},r}.$$

By definition of t_{b}, we have $\sum_{r<t_{\mathsf{b}}} x_{\mathsf{b},r} = \beta$, therefore our proof for the case of $t_{\mathsf{b}} \ge \tau$ is completed and in the rest of the proof we assume $t_{\mathsf{b}} < \tau$. Recall definition

$$\mathcal{J}^{-}_{\tau} := \{p = (\mathsf{b},\mathsf{b}',r,r') \in \mathcal{T}_\tau \,|\, \beta_{\mathsf{a},\mathsf{b}'} < \tau \text{ and } r < t_{\mathsf{b}}\}.$$

It is easy to verify that if $t_{\mathsf{b}} < \tau$ then $\mathcal{J}^{-}_{\tau} \cap \mathcal{Q}_{\mathsf{b},\mathsf{a}} = \emptyset$. Since $\mathcal{T}_\tau$ is partitioned to disjoint sets of $\mathcal{J}^{+}_{\tau}$, $\mathcal{J}^{-}_{\tau}$, and $\mathcal{L}_\tau$ we obtain.

$$\sum_{p\in\mathcal{T}_\tau\cap\mathcal{Q}_{\mathsf{b}}} s^{\star}_{\mathsf{a},p} = \sum_{p\in\mathcal{J}^{+}_{\tau}\cap\mathcal{Q}_{\mathsf{b}}} s^{\star}_{\mathsf{a},p} + \sum_{p\in\mathcal{L}_\tau\cap\mathcal{Q}_{\mathsf{b}}} s^{\star}_{\mathsf{a},p}.$$

In addition, by Claim 3.6.6, we have

$$(1-\beta)\Pr[\beta_{\mathsf{a},\mathsf{b}} \geq r'_{\mathsf{b}} \geq \tau] \geq \sum_{p\in\mathcal{J}_\tau^+\cap\mathcal{Q}_{\mathsf{b}}} s^\star_{\mathsf{a},p},$$

which gives us

$$\sum_{p\in\mathcal{T}_\tau\cap\mathcal{Q}_{\mathsf{b}}} s^\star_{\mathsf{a},p} - (1-\beta)\Pr[\beta_{\mathsf{a},\mathsf{b}} \geq r'_{\mathsf{b}} \geq \tau] \leq \sum_{p\in\mathcal{L}_\tau\cap\mathcal{Q}_{\mathsf{b}}} s^\star_{\mathsf{a},p}.$$

The proof is then completed using Claim 3.6.4 that shows

$$\sum_{p\in\mathcal{L}_\tau\cap\mathcal{Q}_{\mathsf{b}}} s^\star_{\mathsf{a},p} + \sum_{p\in\mathcal{L}_\tau\cap\mathcal{Q}'_{\mathsf{b}}} s^\star_{\mathsf{a},p}/k \leq \delta_\tau$$

for any buyer b. □

Claim 3.6.8. *For any buyer* b *with* $\beta_{\mathsf{a},\mathsf{b}} \geq \tau$ *we have*

$$\mathbb{E}[q_{\mathsf{b}}] \geq \min\left(\frac{1}{\beta}\left(\sum_{p\in\mathcal{T}_\tau\cap\mathcal{Q}_{\mathsf{b}}} s^\star_{\mathsf{a},p} + \sum_{p\in\mathcal{L}_\tau\cap\mathcal{Q}'_{\mathsf{b}}} s^\star_{\mathsf{a},p}/k - (1-\beta)\Pr[\beta_{\mathsf{a},\mathsf{b}} \geq r'_{\mathsf{b}} \geq \tau]\right), 1\right)$$

Proof. We provide two different proofs for cases of $t_b \leq \beta_{\mathsf{a},\mathsf{b}}$ and $t_b > \beta_{\mathsf{a},\mathsf{b}}$. We claim that in the first case, we have $\mathbb{E}[q_b] = 1$ and in the second case,

$$\mathbb{E}[q_{\mathsf{b}}] \geq \frac{1}{\beta}\left(\sum_{p\in\mathcal{T}_\tau\cap\mathcal{Q}_{\mathsf{b}}} s^\star_{\mathsf{a},p} + \sum_{p\in\mathcal{L}_\tau\cap\mathcal{Q}'_{\mathsf{b}}} s^\star_{\mathsf{a},p}/k - \Pr[\beta_{\mathsf{a},\mathsf{b}} \geq r'_b \geq \tau]\right).$$

By construction, for vector of reserve prices $\mathbf{r}$ and any buyer b we have

$$\mathbb{E}[q_{\mathsf{b}}] = \Pr\left[r_{\mathsf{b}} \leq \beta_{\mathsf{a},\mathsf{b}}\right] = \sum_{r\leq\beta_{\mathsf{a},\mathsf{b}}} f_{\mathsf{b},r},$$

where $f_{\mathsf{b},r} = x_{b,r}/\beta$ for any $r < t_{\mathsf{b}}$ and $f_{\mathsf{b},r} = 0$ for any $r \geq t_b$ as defined in the algorithm. Note that t_b is chosen in a way that $\sum_{r<t_{\mathsf{b}}} x_{b,r} = \beta$. Thus, if $t_b \leq \beta_{\mathsf{a},\mathsf{b}}$ we get

$$\mathbb{E}[q_{\mathsf{b}}] = \sum_{r \leq \beta_{\mathsf{a},\mathsf{b}}} f_{\mathsf{b},r} = \sum_{\substack{r: r \leq \beta_{\mathsf{a},\mathsf{b}}, \\ r < t_{\mathsf{b}}}} x_{\mathsf{b},r}/\beta = \sum_{r<t_{\mathsf{b}}} x_{\mathsf{b},r}/\beta = 1.$$

This completes the proof for the first case. Therefore, in the rest of the proof we focus on the case of $t_b > \beta_{\mathsf{a},\mathsf{b}}$. This gives us

$$\mathbb{E}[q_{\mathsf{b}}] = \sum_{r \leq \beta_{\mathsf{a},\mathsf{b}}} f_{\mathsf{b},r} = \sum_{\substack{r: r \leq \beta_{\mathsf{a},\mathsf{b}}, \\ r < t_{\mathsf{b}}}} x_{\mathsf{b},r}/\beta = \sum_{r<\beta_{\mathsf{a},\mathsf{b}}} x_{\mathsf{b},r}/\beta \geq \sum_{r<\beta_{\mathsf{a},\mathsf{b}}} \left(y_{\mathsf{b},r} + y'_{\mathsf{b},r}\right)/\beta,$$

where the right hand side is by constraint $y_{\mathsf{b},r} + y'_{\mathsf{b},r} \leq x_{\mathsf{b},r}$ in the LP. Further by the first two constraints of the LP we obtain

$$\sum_{r<\beta_{\mathsf{a},\mathsf{b}}} \left(y_{\mathsf{b},r} + y'_{\mathsf{b},r}\right) = \sum_{r<\beta_{\mathsf{a},\mathsf{b}}} \left(\sum_{p \in \mathcal{Q}_{\mathsf{b},r,\mathsf{a}}} s_{\mathsf{a},p} + \sum_{p \in \mathcal{Q}'_{\mathsf{b},r,\mathsf{a}}} s_{\mathsf{a},p}/k \right).$$

Note that we can drop the constraint $r < \beta_{\mathsf{a},\mathsf{b}}$ from the right hand side of the equation since by definition of valid sub-profiles it holds for any p in $\mathcal{Q}_{\mathsf{b},\mathsf{a}}$ or $\mathcal{Q}'_{\mathsf{b},\mathsf{a}}$. This gives us

$$\sum_{r<\beta_{\mathsf{a},\mathsf{b}}} \left(\sum_{p \in \mathcal{Q}_{\mathsf{b},r,\mathsf{a}}} s_{\mathsf{a},p} + \sum_{p \in \mathcal{Q}'_{\mathsf{b},r,\mathsf{a}}} s_{\mathsf{a},p}/k \right) = \sum_{p \in \mathcal{Q}_{\mathsf{b},\mathsf{a}}} s_{\mathsf{a},p} + \sum_{p \in \mathcal{Q}'_{\mathsf{b},\mathsf{a}}} s_{\mathsf{a},p}/k \geq \sum_{p \in \mathcal{T}_\tau \cap \mathcal{Q}_{\mathsf{b},\mathsf{a}}} s^\star_{\mathsf{a},p} + \sum_{p \in \mathcal{L}_\tau \cap \mathcal{Q}'_{\mathsf{b},\mathsf{a}}} s^\star_{\mathsf{a},p}/k,$$

which completes our proof. □

Lemma 3.4.4. (restated) The following inequality holds.

$$F(\mathbf{s}^\star, \tau) \leq \sum_{p \in \mathcal{J}_\tau^-} s^\star_{\mathsf{a},p} + \sum_{p \in \mathcal{L}_\tau} s^\star_{\mathsf{a},p}.$$

Proof. Recall that by definition

$$F(\mathbf{s}^\star, \tau) = \sum_{p \in \mathcal{T}_{\mathsf{a},\tau}} s^\star_{\mathsf{a},p} - (1-\beta)\mathbb{E}[W_{\mathsf{a}}(\mathbf{r}', \tau)].$$

Moreover, by Assumption 3.6.2,

$$\mathbb{E}[W_{\mathsf{a}}(\mathbf{r}', \tau)] = \sum_{\mathsf{b} \in \mathsf{B}} \Pr[\beta_{\mathsf{a},\mathsf{b}} \geq r'_b \geq \tau].$$

In addition based on Claim 3.6.6, we know

$$(1-\beta) \sum_{\mathsf{b} \in \mathsf{B}} \Pr[\beta_{\mathsf{a},\mathsf{b}} \geq r'_b \geq \tau] \geq \sum_{p \in \mathcal{J}_\tau^+} s^\star_{\mathsf{a},p}.$$

Considering that $\mathcal{T}_\tau$ is partitioned to three disjoint sets of $\mathcal{J}_\tau^+, \mathcal{J}_\tau^-$, and $\mathcal{L}_\tau$, by putting the mentioned equations together, we get

$$F(\mathbf{s}^\star, \tau) = \sum_{p \in \mathcal{T}_{\mathsf{a},\tau}} s^\star_{\mathsf{a},p} - (1-\beta)\mathbb{E}[W_{\mathsf{a}}(\mathbf{r}', \tau)] \leq \sum_{p \in \mathcal{T}_{\mathsf{a},\tau}} s^\star_{\mathsf{a},p} - \sum_{p \in \mathcal{J}_\tau^+} s^\star_{\mathsf{a},p} \leq \sum_{p \in \mathcal{J}_\tau^-} s^\star_{\mathsf{a},p} + \sum_{p \in \mathcal{L}_\tau} s^\star_{\mathsf{a},p}.$$

□

Lemma 3.4.6. (restated) For $\mathsf{B}_1 = \{\mathsf{b} \in \mathsf{B} : \mathbb{E}[q_\mathsf{b}] = 1\}$, we have

$$\mathbb{E}[Q_\tau - |\mathsf{B}_1|] \geq (k - |\mathsf{B}_1| + 1)\delta/\beta.$$

Proof. By construction of vector of reserve prices $\mathbf{r}$ and $\mathbf{r}'$, for any buyer b we have

$$\sum_{r \leq \beta_{\mathsf{a},\mathsf{b}}} x_{r,\mathsf{b}} = \beta \Pr[r \leq \beta_{\mathsf{a},\mathsf{b}}] + (1-\beta)\Pr[r \leq \beta_{\mathsf{a},\mathsf{b}}].$$

Moreover, by the third constraint of the LP for any buyer b and reserve price $r \in \mathsf{R}$, we have $y_{\mathsf{b},r,\mathsf{a}} + y'_{\mathsf{b},r,\mathsf{a}} \leq x_{\mathsf{b},r}$, which implies

$$\sum_{r \leq \beta_{\mathsf{a},\mathsf{b}}} \left(y_{\mathsf{b},r,\mathsf{a}} + y'_{\mathsf{b},r,\mathsf{a}}\right) = \beta \Pr[r \leq \beta_{\mathsf{a},\mathsf{b}}] + (1-\beta)\Pr[r \leq \beta_{\mathsf{a},\mathsf{b}}].$$

Combining this with the first two constraints of the LP, $y_{\mathsf{b},r,\mathsf{a}} = \sum_{p \in \mathcal{Q}_{\mathsf{b},r,\mathsf{a}}} s_{\mathsf{a},p}$ and $y'_{\mathsf{b},r,\mathsf{a}} = \sum_{p \in \mathcal{Q'}_{\mathsf{b},r,\mathsf{a}}} s_{\mathsf{a},p}$ we get

$$\sum_{r \leq \beta_{\mathsf{a},\mathsf{b}}} \left(\sum_{p \in \mathcal{Q}_{\mathsf{b},r,\mathsf{a}}} s^\star_{\mathsf{a},p} + \sum_{p \in \mathcal{Q'}_{\mathsf{b},r,\mathsf{a}}} s^\star_{\mathsf{a},p}/k \right) = \beta \Pr[r \leq \beta_{\mathsf{a},\mathsf{b}}] + (1-\beta)\Pr[r \leq \beta_{\mathsf{a},\mathsf{b}}].$$

By definition of $\mathcal{Q}_{\mathsf{b},r}$ and $\mathcal{Q'}_{\mathsf{b},\mathsf{a}}$ we can write

$$\sum_{p \in \mathcal{Q}_{\mathsf{b},\mathsf{a}}} s^\star_{\mathsf{a},p} + \sum_{p \in \mathcal{Q'}_{\mathsf{b},\mathsf{a}}} s^\star_{\mathsf{a},p}/k = \beta \Pr[r \leq \beta_{\mathsf{a},\mathsf{b}}] + (1-\beta)\Pr[r \leq \beta_{\mathsf{a},\mathsf{b}}]$$

and

$$\sum_{p \in \mathcal{Q}_{\mathsf{b},\mathsf{a}} \cap \mathcal{L}_\tau} s^\star_{\mathsf{a},p} + \sum_{p \in \mathcal{Q'}_{\mathsf{b},\mathsf{a}} \cap \mathcal{L}_\tau} s^\star_{\mathsf{a},p}/k \leq \beta \Pr[r \leq \beta_{\mathsf{a},\mathsf{b}}] + (1-\beta)\Pr[r \leq \beta_{\mathsf{a},\mathsf{b}}].$$

Further, by Claim 3.6.4, for any buyer b we have

$$\sum_{p \in \mathcal{L}_\tau \cap \mathcal{Q}_{\mathsf{b}}} s^\star_{\mathsf{a},p} + \sum_{p \in \mathcal{L}_\tau \cap \mathcal{Q'}_{\mathsf{b}}} s^\star_{\mathsf{a},p}/k \leq \delta_\tau.$$

which means

$$\sum_{\mathsf{b} \in \mathsf{B}} \left(\sum_{p \in \mathcal{Q}_{\mathsf{b},\mathsf{a}} \cap \mathcal{L}_\tau} s_{\mathsf{a},p} + \sum_{p \in \mathcal{Q'}_{\mathsf{b},\mathsf{a}} \cap \mathcal{L}_\tau} s'_{\mathsf{a},p}/k \right) - \delta_\tau |\mathsf{B}_1| \leq \sum_{\mathsf{b} \in \mathsf{B} \setminus \mathsf{B}_1} \beta \Pr[r \leq \beta_{\mathsf{a},\mathsf{b}}] + (1-\beta) \Pr[r \leq \beta_{\mathsf{a},\mathsf{b}}].$$

Further, by 3.6.5, we know $\Pr[r \leq \beta_{\mathsf{a},\mathsf{b}}] = 0$ holds for any $\mathsf{b} \notin \mathsf{B}_1$. This implies

$$\sum_{\mathsf{b} \in \mathsf{B}} \left(\sum_{p \in \mathcal{Q}_{\mathsf{b},\mathsf{a}} \cap \mathcal{L}_\tau} s_{\mathsf{a},p} + \sum_{p \in \mathcal{Q'}_{\mathsf{b},\mathsf{a}} \cap \mathcal{L}_\tau} s'_{\mathsf{a},p}/k \right) - \delta_\tau |\mathsf{B}_1| \leq \sum_{\mathsf{b} \in \mathsf{B} \setminus \mathsf{B}_1} \beta \Pr[r \leq \beta_{\mathsf{a},\mathsf{b}}] = \sum_{\mathsf{b} \in \mathsf{B} \setminus \mathsf{B}_1} \beta \mathbb{E}[q_{\mathsf{b}}].$$

To complete the proof, recall that we have defined $\delta_\tau = \sum_{b \in \mathsf{B}_\tau} \sum_{p \in \mathcal{Q'}_{\mathsf{b},\mathsf{a}} \cap \mathcal{L}_\tau} s^\star_{\mathsf{a},p}/k$, and by Claim 3.4.2 we have

$$\sum_{\mathsf{b} \in \mathsf{B}_\tau} \sum_{p \in \mathcal{Q}_{\mathsf{b},\mathsf{a}} \cap \mathcal{L}_\tau} s^\star_{\mathsf{a},p} = \sum_{p \in \mathcal{L}_\tau} s^\star_{\mathsf{a},p} = k\delta_\tau.$$

This implies

$$\sum_{p \in \mathcal{Q}_{\mathsf{b},\mathsf{a}} \cap \mathcal{L}_\tau} s^\star_{\mathsf{a},p} + \sum_{p \in \mathcal{Q'}_{\mathsf{b},\mathsf{a}} \cap \mathcal{L}_\tau} s^\star_{\mathsf{a},p}/k = (k+1)\delta_\tau,$$

$$(k + 1 - |\mathsf{B}_1|)\delta_\tau \leq \sum_{\mathsf{b} \in \mathsf{B} \setminus \mathsf{B}_1} \beta \mathbb{E}[q_{\mathsf{b}}] = \beta(\mathbb{E}[Q_\tau] - |\mathsf{B}_1|),$$

$$(k + 1 - |\mathsf{B}_1|)\delta_\tau/\beta \leq (\mathbb{E}[Q_\tau] - |\mathsf{B}_1|),$$

and concludes the proof. □

Lemma 3.4.7. (restated) For $\mathsf{B}_1 = \{\mathsf{b} \in \mathsf{B} : \mathbb{E}[q_\mathsf{b}] = 1\}$ and $m = |\mathsf{B}_1|$ we have

$$\mathbb{E}[Q_\tau - m] \geq \frac{F(s^\star, \tau) - m\delta}{\beta} + \delta/\beta.$$

Proof. Recall that by definition

$$F(\mathbf{s}^\star, \tau) = \sum_{p \in \mathcal{T}_{\mathsf{a},\tau}} s^\star_{\mathsf{a},p} - (1-\beta)\mathbb{E}[W_\mathsf{a}(\mathbf{r}', \tau)].$$

Combining this with Asspmtion 3.6.2 gives us

$$F(s^\star, \tau) = \sum_{\mathsf{b} \in \mathsf{B}} \left(\sum_{p \in \mathcal{T}_\tau \cap \mathcal{Q}_\mathsf{b}} s^\star_{\mathsf{a},p} - (1-\beta)\Pr[\beta_{\mathsf{a},\mathsf{b}} \geq r'_\mathsf{b} \geq \tau] \right),$$

which results in

$$\frac{F(s^\star, \tau) + \delta}{\beta} = \frac{1}{\beta} \sum_{\mathsf{b} \in \mathsf{B}} \left(\sum_{p \in \mathcal{T}_\tau \cap \mathcal{Q}_\mathsf{b}} s^\star_{\mathsf{a},p} + \sum_{p \in \mathcal{L}_\tau \cap \mathcal{Q}'_\mathsf{b}} s^\star_{\mathsf{a},p}/k - (1-\beta)\Pr[\beta_{\mathsf{a},\mathsf{b}} \geq r'_\mathsf{b} \geq \tau] \right). \quad (3.22)$$

Moreover, by Claim 3.6.8, for any buyer b, we have

$$\mathbb{E}[q_\mathsf{b}] \geq \min\left(\frac{1}{\beta} \left(\sum_{p \in \mathcal{T}_\tau \cap \mathcal{Q}_\mathsf{b}} s^\star_{\mathsf{a},p} + \sum_{p \in \mathcal{L}_\tau \cap \mathcal{Q}'_\mathsf{b}} s^\star_{\mathsf{a},p}/k - (1-\beta)\Pr[\beta_{\mathsf{a},\mathsf{b}} \geq r'_\mathsf{b} \geq \tau] \right), 1 \right),$$

and by Claim 3.6.7 for any buyer b, we have

$$\sum_{p \in \mathcal{T}_\tau \cap \mathcal{Q}_\mathsf{b}} s^\star_{\mathsf{a},p} + \sum_{p \in \mathcal{L}_\tau \cap \mathcal{Q}'_\mathsf{b}} s^\star_{\mathsf{a},p}/k - (1-\beta)\Pr[\beta_{\mathsf{a},\mathsf{b}} \geq r'_\mathsf{b} \geq \tau] \leq \max(\beta, \delta_\tau).$$

This implies that if $\delta_\tau \leq \beta$, then for any buyer b we have

$$\mathbb{E}[q_{\mathsf{b}}] \geq \frac{1}{\beta}\left(\sum_{p\in\mathcal{T}_\tau\cap\mathcal{Q}_{\mathsf{b}}} s^\star_{\mathsf{a},p} + \sum_{p\in\mathcal{L}_\tau\cap\mathcal{Q}'_{\mathsf{b}}} s^\star_{\mathsf{a},p}/k - (1-\beta)\Pr[\beta_{\mathsf{a},\mathsf{b}} \geq r'_{\mathsf{b}} \geq \tau]\right),$$

and

$$\mathbb{E}[Q_\tau] \geq \frac{F(s^\star,\tau)+\delta}{\beta}.$$

This completes the proof for the case of $\delta_\tau \leq \beta$; therefore, in the rest of the proof we assume $\delta_\tau > \beta$ which means for any buyer b,

$$\sum_{p\in\mathcal{T}_\tau\cap\mathcal{Q}_{\mathsf{b}}} s^\star_{\mathsf{a},p} + \sum_{p\in\mathcal{L}_\tau\cap\mathcal{Q}'_{\mathsf{b}}} s^\star_{\mathsf{a},p}/k - (1-\beta)\Pr[\beta_{\mathsf{a},\mathsf{b}} \geq r'_{\mathsf{b}} \geq \tau] \leq \delta_\tau. \tag{3.23}$$

Note that for any $\mathsf{b} \in \mathsf{B}/\mathsf{B}_1$ we have $q_{\mathsf{b}} < 1$ which means

$$\mathbb{E}[q_{\mathsf{b}}] = \frac{1}{\beta}\left(\sum_{p\in\mathcal{T}_\tau\cap\mathcal{Q}_{\mathsf{b}}} s^\star_{\mathsf{a},p} + \sum_{p\in\mathcal{L}_\tau\cap\mathcal{Q}'_{\mathsf{b}}} s^\star_{\mathsf{a},p}/k - (1-\beta)\Pr[\beta_{\mathsf{a},\mathsf{b}} \geq r'_{\mathsf{b}} \geq \tau]\right).$$

Combining this with Equation 3.22, we obtain

$$\begin{aligned}\mathbb{E}[Q_\tau - m] = \sum_{\mathsf{b}\in\mathsf{B}/\mathsf{B}_1}\mathbb{E}[q_t] = \frac{F(s^\star,\tau)+\delta}{\beta} \\ - \frac{1}{\beta}\sum_{\mathsf{b}\in\mathsf{B}_1}\left(\sum_{p\in\mathcal{T}_\tau\cap\mathcal{Q}_{\mathsf{b}}} s^\star_{\mathsf{a},p} + \sum_{p\in\mathcal{L}_\tau\cap\mathcal{Q}'_{\mathsf{b}}} s^\star_{\mathsf{a},p}/k - (1-\beta)\Pr[\beta_{\mathsf{a},\mathsf{b}} \geq r'_{\mathsf{b}} \geq \tau]\right).\end{aligned}$$

Using Equation 3.23, we get the following which completes the proof

$$\mathbb{E}[Q_\tau - m] = \sum_{\mathsf{b}\in\mathsf{B}/\mathsf{B}_1}\mathbb{E}[q_t] \geq \frac{F(s^\star,\tau)+\delta_\tau}{\beta} - \frac{m\delta_\tau}{\beta} = \frac{F(s^\star,\tau)-m\delta_\tau}{\beta} + \delta_\tau/\beta.$$

□

3.7 Proof of Lemma 3.2.3

Lemma 3.2.3. (restated) The optimal revenue is upper bounded by $\sum_{\mathsf{a}\in\mathsf{A}}\sum_{p\in\mathcal{S}_\mathsf{a}} s^\star_{\mathsf{a},p}\cdot \mathsf{Rev}_\mathsf{a}(p)$.

Proof. Consider OPT, an optimal solution of the problem. To prove this claim, it suffices to construct vectors $\mathbf{s}^{\mathrm{o}}$, $\mathbf{x}^o$, and $\mathbf{y}^o$ in a way that setting $\mathbf{s}=\mathbf{s}^{\mathrm{o}}$, $\mathbf{y}=\mathbf{y}^o$ and $\mathbf{x}=\mathbf{x}^o$ in the LP satisfies all the LP constraints and that the objective function of the LP equals to the revenue of OPT, or in the other words

$$\mathsf{Rev}(\textsc{opt}) = \max_{\mathbf{x},\mathbf{s}} \sum_{\mathsf{a}\in\mathsf{A}}\sum_{p\in\mathcal{P}_\mathsf{a}} s^o_{\mathsf{a},p}\cdot\mathsf{Rev}_\mathsf{a}(p). \tag{3.24}$$

Roughly speaking, we construct $\mathbf{s}^{\mathrm{o}}$ to be the representation of OPT in the profile space. For any sub-profile $p=(\mathsf{b}_1,\mathsf{b}_2,r_1,r_2)$ we have $s^{\mathrm{o}}_{a,p}=1$ iff using OPT, in auction a buyer b_1 is one of the winners, buyer b_2 is the supporting buyer, and their reserve prices are respectively r_1 and r_2.

We first show that Equation 3.24 holds for $\mathbf{s}^o$. Let $\mathbf{r}$ denote the vector of reserve prices in OPT. For any auction $\mathsf{a}\in\mathsf{A}$, let $\mathsf{b}_{s,\mathsf{a}}$ be the supporting buyer, and let B_a denote the set of winners in auction a using vector of reserve prices $\mathbf{r}$. Payment of any winner $\mathsf{b}\in\mathsf{B}_\mathsf{a}$ in auction a is $\max(r_\mathsf{b},\beta_{\mathsf{a},\mathsf{b}_{s,\mathsf{a}}})$, which means revenue obtained from auction a using the vector of reserve prices $\mathbf{r}$ is $\sum_{\mathsf{b}\in\mathsf{B}_\mathsf{a}}\max(r_\mathsf{b},\beta_{\mathsf{a},\mathsf{b}_{s,\mathsf{a}}})$. To prove Equation 3.24, it suffices to show that for any auction a we have

$$\sum_{\mathsf{b}\in\mathsf{B}_\mathsf{a}} \max(r_\mathsf{b}, \beta_{\mathsf{a},\mathsf{b}_{s,\mathsf{a}}}) = \sum_{p\in\mathcal{P}_\mathsf{a}} \mathsf{Rev}_\mathsf{a}(p).$$

Note that for any profile $p = (\mathsf{b}, r'_1, \mathsf{b}_2, r'_2) \in \mathcal{P}_\mathsf{a}$ we have $s_{\mathsf{a},p} = 1$ iff $\mathsf{b} \in \mathsf{B}_\mathsf{a}$, $\mathsf{b}_2 = \mathsf{b}_{s,\mathsf{a}}$, $r_\mathsf{b} = r'_1$ and $r_{\mathsf{b}_2} = r'_2$. Moreover, by defintion, we have $\mathsf{Rev}_\mathsf{a}(p) = \max(r'_1, \beta_{\mathsf{a},\mathsf{b}_2}) = \max(r_\mathsf{b}, \beta_{\mathsf{a},\mathsf{b}_{s,\mathsf{a}}})$. This implies that

$$\sum_{p\in\mathcal{P}_\mathsf{a}} \mathsf{Rev}_\mathsf{a}(p) = \max(r_\mathsf{b}, \beta_{\mathsf{a},\mathsf{b}_{s,\mathsf{a}}}),$$

which results in Equation 3.24.

To complete the proof we construct $\mathbf{x}^o$, $\mathbf{y}^o$, and $\mathbf{y}'^o$ in a way that setting $\mathbf{x} = \mathbf{x}^o$, $\mathbf{y}' = \mathbf{y}'^o$, $\mathbf{y} = \mathbf{y}'^o$ and $\mathbf{s} = \mathbf{s}^\text{o}$, satisfied all the constraints of the LP.

- For any buyer $\mathsf{b} \in \mathsf{B}$ and $r \in \mathsf{R}$ we set $x^o_{\mathsf{b},r} = 1$ iff reserve price r is assigned to buyer b in OPT and set $x^o_{\mathsf{b},r} = 0$ otherwise.

- For any buyer $\mathsf{b} \in \mathsf{B}$, auction $\mathsf{a} \in \mathsf{A}$, and reserve price $r \in \mathsf{R}$ we set $y^o_{\mathsf{b},r,\mathsf{a}} = 1$ iff using solution OPT, buyer b is a winner in auction a and he is assigned a reserve price r. Otherwise we set $y^o_{\mathsf{b},r,\mathsf{a}} = 0$.

- For any buyer $\mathsf{b} \in \mathsf{B}$, auction $\mathsf{a} \in \mathsf{A}$, and reserve price $r \in \mathsf{R}$ we set $y'^o_{\mathsf{b},r,\mathsf{a}} = 1$ if using solution OPT, buyer b is the supporting buyer in auction a and he is assigned a reserve price r. Otherwise we set $y'^o_{\mathsf{b},r,\mathsf{a}} = 0$.

We now start investigating the constraints of the LP one by one and verify that all seven constraints hold for $\mathbf{x} = \mathbf{x}^o$, $\mathbf{y}' = \mathbf{y}'^o$, $\mathbf{y} = \mathbf{y}'^o$ and $\mathbf{s} = \mathbf{s}^o$.

1. For the first constraint we need to show $y^o_{\mathsf{b},r,\mathsf{a}} = \sum_{p \in \mathcal{Q}_{\mathsf{b},r,\mathsf{a}}} s^o_{\mathsf{a},p}$ for any $\mathsf{b} \in \mathsf{B}, r \in \mathsf{R}$, and $\mathsf{a} \in \mathsf{A}$. It is easy to see that if b is not a winner in auction a then both sides of this equation are equal to zero. To complete the proof we assume that b is a winner. Let b_2 denote the supporting buyer in auction a. Moreover, let r_1 and r_2 respectively denote the reserve prices assigned to buyers b and b_2 in OPT. We have $y^o_{\mathsf{b},r,\mathsf{a}} = 1$ iff $r = r_1$ and b is a winner in auction a. We also show that $\sum_{p \in \mathcal{Q}_{\mathsf{b},r,\mathsf{a}}} s^o_{\mathsf{a},p} = 1$ iff $r = r_1$. Note that $p = (\mathsf{b}, r_1, \mathsf{b}_2, r_2)$ is the only sub-profiles in $\mathcal{S}_\mathsf{a}$ with $s^o_{\mathsf{a},p} = 1$. Further, by definition $\mathcal{Q}_{\mathsf{b},r,\mathsf{a}}$ is a subset of $\mathcal{S}_\mathsf{a}$ and contains the sub-profiles in which buyer b is a winner in auction a, thus it contains p iff $r = r_1$. This means that both sides of the equation are equal to one if $r = r_1$ and both are equal to zero otherwise.

2. For the second constraint, we need to show that $y'^o_{\mathsf{b},r,\mathsf{a}} = \sum_{p \in \mathcal{Q}'_{\mathsf{b},r,\mathsf{a}}} s^o_{\mathsf{a},p}/k$ holds for any $\mathsf{b} \in \mathsf{B}, r \in \mathsf{R}$, and $\mathsf{a} \in \mathsf{A}$. Let b_1 be the supporting buyer in auction a and let r_1 be the reserve price assigned to this buyer. By definition we have $y'^o_{\mathsf{b},r,\mathsf{a}} = 1$ iff $\mathsf{b} = \mathsf{b}_1$ and $r = r_1$, and we have $y'^o_{\mathsf{b},r,\mathsf{a}} = 1$ otherwise. Therefore, for this constraint to be satisfied we need to show that $\sum_{p \in \mathcal{Q}'_{\mathsf{b},r,\mathsf{a}}} s^o_{\mathsf{a},p} = k$ holds iff $\mathsf{b} = \mathsf{b}_1$ and $r = r_1$, and we have $\sum_{p \in \mathcal{Q}'_{\mathsf{b},r,\mathsf{a}}} s^o_{\mathsf{a},p} = 0$ otherwise. Let b_2 denote one of the winners in auction a and let r_2 be the reserve price assigned to this buyer. By definition, for any profile $p = (\mathsf{b}_2, r_2, \mathsf{b}_1, r_1) \in \mathcal{S}_\mathsf{a}$ we have $s^o_{\mathsf{a},p} = 1$ iff b_2 is a winner in auction a and r_1 is the reserve price assigned to him. Since we have assumed that each auction has exactly k winners then we have $\sum_{p \in \mathcal{Q}'_{\mathsf{b},r,\mathsf{a}}} s^o_{\mathsf{a},p} = k$

iff $\mathsf{b} = \mathsf{b}_1$ and $r = r_2$, and we have $\sum_{p \in \mathcal{Q'}_{\mathsf{b},r,\mathsf{a}}} s^o_{\mathsf{a},p} = 0$ otherwise.

3. To prove that our constructed solution satisfies the third constraint of the LP, we need to show $y^o_{\mathsf{b},r,\mathsf{a}} + y'^o_{\mathsf{b},r,\mathsf{a}} \leq x^o_{\mathsf{b},r}$ for any $\mathsf{b} \in \mathsf{B}$, any reserve price $r \in \mathsf{R}$ and any auction a. Consider a buyer b and let r_1 be the reserve price assigned to this buyer in solution OPT. For any $r \neq r_1$ both sides of the equation are obviously zero. However, for $r = r_1$ we have $x^o_{\mathsf{b},r} = 1$. Observe that in any auction a, we also have $y^o_{\mathsf{b},r,\mathsf{a}} + y'^o_{\mathsf{b},r,\mathsf{a}} \leq 1$ since b cannot be both a winner and the supporting buyer in an auction.

4. For the fourth constraint, we need to show $\sum_{p \in \mathcal{Q}_{\mathsf{b}_2,\mathsf{a}} \cap \mathcal{Q'}_{\mathsf{b}_1,\mathsf{a}}} s^o_{\mathsf{a},p} \leq \sum_{r \in \mathsf{R}} y'^o_{\mathsf{b}_1,r,\mathsf{a}}$. For any sub-profile $p = (\mathsf{b}_3, r_1, \mathsf{b}_4, r_2)$ in $\mathcal{Q}_{\mathsf{b}_2,\mathsf{a}} \cap \mathcal{Q'}_{\mathsf{b}_1,\mathsf{a}}$ we have $\mathsf{b}_4 = \mathsf{b}_2$ and $\mathsf{b}_3 = \mathsf{b}_1$. Moreover, we have $s_{\mathsf{a},p} = 1$ iff r_1 and r_2 are respectively the reserve prices assigned to buyers b_1 and b_2, buyer b_1 is a winner in auction a and buyer b_2 is the supporting buyer in this auction. This implies that the left hand side is equal to one iff b_1 and b_2 are respectively a winner and the supporting buyer in auction a. Further, the right hand side is equal to one iff b_1 is the supporting buyer in auction a. This concludes that the fourth constraint is satisfied for $\mathbf{s}^o$ and $\mathbf{y}'^o$.

5. The condition $\sum_{p \in \mathcal{P}_\mathsf{a}} s^o_{\mathsf{a},p} \leq k$ is satisfied due to the assumption that we have k winners in all auctions. Let b' be the supporting buyer in auction a. For a sub-profile $(\mathsf{b}_1, r_1, \mathsf{b}_2, r_2) \in \mathcal{P}_\mathsf{a}$ we have $s^o_{\mathsf{b},r} = 1$ iff $\mathsf{b}_2 = \mathsf{b}'$, b_1 is a winner in auction a, and r_1 and r_2 are respectively the reserve prices assigned to buyers b_1 and b_2. Given that we have exactly k winner and that each buyer has a unique

reserve price then, $\sum_{p \in \mathcal{P}_{\mathsf{a}}} s^o_{\mathsf{a},p} \leq k$.

6. For this constraint, we need to show $\sum_{r \in \mathsf{R}} x^o_{\mathsf{b},r} = 1$. This is true since for a reserve r we have $x^o_{\mathsf{b},r} = 1$ if reserve r is assigned to buyer b, and in OPT there is exactly one reserve price assigned to each buyer.

7. The last constraint is simply satisfied since $s^o_{\mathsf{a},p}$ is either zero or one.

□

3.8 Useful Facts about Bernoulli Random Variables

Lemma 3.8.1. *If $Y \sim$ Binomial(n, p), then for any $m \leq n$ we have* $\Pr(Y \geq m) = G(p)$, *where*

$$G(p) = \frac{n!}{(m-1)!(n-m)!} \int_{1-p}^{1} t^{n-m}(1-t)^{m-1} dt.$$

Proof. Let us define $H(p) := \Pr(Y \geq m)$. We have

$$H(p) = \sum_{j=m}^{n} \binom{n}{j} p^j (1-p)^{n-j}.$$

By taking derivative of this function we get:

$$\begin{aligned}
H'(p) &= \sum_{j=m}^{n} \binom{n}{j} j p^{j-1}(1-p)^{n-j} - \sum_{j=m}^{n} \binom{n}{j} (n-j) p^{j}(1-p)^{n-j-1} \\
&= n \sum_{j=m}^{n} \binom{n-1}{j-1} p^{j-1}(1-p)^{n-j} - n \sum_{j=m}^{n-1} \binom{n-1}{j} p^{j}(1-p)^{n-j-1} \\
&= n \sum_{i=m-1}^{n-1} \binom{n-1}{i} p^{i}(1-p)^{n-1-i} - n \sum_{i=m}^{n-1} \binom{n-1}{i} p^{i}(1-q)^{n-1-i} \\
&= n \binom{n-1}{m-1} p^{m-1}(1-p)^{n-m} = G'(p).
\end{aligned}$$

The proof is completed as we also have $G(0) = H(0) = 0$. □

Lemma 3.8.2. *For any $\alpha > 1$ and $m > 0$, $\Pr[X \geq m]$ is minimized subject to $X \sim$ Binomial$(n, m\alpha/n)$ when $n \to \infty$.*

Proof. Let us define $X_n \sim \text{Binomial}(n, m\alpha/n)$ and $X_{n+1} \sim \text{Binomial}(n, m\alpha/(n+1))$ We need to show that $\Pr[X_n \geq m] \leq \Pr[X_{n+1} \geq m]$. decreasing function of n. Using Lemma 3.8.1, we have

$\Pr[X_n \geq m] = \frac{n!}{(m-1)!(n-m)!} G(n)$ where

$$G(n, \alpha) = \frac{n!}{(m-1)!(n-m)!} \int_{1-m\alpha/n}^{1} t^{n-m}(1-t)^{m-1} dt.$$

We have

$$\frac{\Pr[X_{n+1} \geq m]}{\Pr[X_n \geq m]} = \frac{(n-m+1)G(n+1)}{(n+1)G(n, \alpha)}.$$

Define $D_{n,\alpha} = (n-m+1)G(n+1) - (n+1)G(n)$. To complete the proof it suffices to show $D_{n,\alpha} \leq 1$.

$$\frac{\partial D_{n,\alpha}}{\partial \alpha}\alpha(m\alpha)^{-m}(n+1)^{m-1}\Big(1-\frac{m\alpha}{n}\Big)^{m-n}$$
$$= D_{n,\alpha;1} := \Big(1-\frac{m\alpha}{n}\Big)^{m-n}\Big(1-\frac{m\alpha}{n+1}\Big)^{n-m+1} - \frac{n-m+1}{n}\Big(\frac{n+1}{n}\Big)^{m-1}.$$

Moreover, for any $\alpha \in (1, n/m)$,

$$\frac{\partial D_{n,\alpha;1}}{\partial \alpha} = \frac{(\alpha-1)m^2 n^{n-m}(n+1-m\alpha)^{n-m}}{(n+1)^{n-m+1}(n-m\alpha)^{n-m+1}} > 0.$$

As a result, sign of $D_{n,\alpha;1}$ can only change from $-$ to $+$ as α increases from 1 to n/m. So, $\frac{\partial D_{n,\alpha}}{\partial \alpha}$ has the same sign pattern. To get $D_{n,\alpha} \le 1$, it suffices to show that $D_{n,0} \le 0$ and $D_{n,n/m} \le 0$.

Since for $\alpha = 0$, we have $G(n,0) = 0$ for all n, we obviously have $D_{n,0} = 0 \le 0$. We conclude the proof by noting the following.

$$D_{n,n/m} = (n+1)G(n+1, n/m) - (n-m+1)G(n, n/m)$$
$$\le (n+1)G(n+1, (n+1)/m) - (n-m+1)G(n, n/m)$$
$$= 1\Big/\binom{n}{m-1} - 1\Big/\binom{n}{m-1} = 0.$$

□

Lemma 3.8.3. *Let* $\mathbf{p} = (p_0, \ldots, p_n) \in [0,1]^n$ *be an extreme point of function* $F_m(\mathbf{p})$

defined below subject to $\sum_{i\in[n]} p_i = t$ *for a fixed* t.

$$F_m(\mathbf{p}) = \Pr\Big[\sum_{i\in[n]} x_i > m\Big],$$

where $x_1, \ldots, x_n$ *are independent Bernoulli random variables with* $\mathbb{E}[x_i] = p_i$ *for any* $i \in [n]$. *The following holds for any* $i, j \in [n]$. *If* $p_i \notin \{0,1\}$ *and* $p_j \notin \{0,1\}$, *then* $p_i = p_j$.

Proof. We use proof by contradiction. If there does not exist such an extreme point, then let $\mathbf{p}$ be an arbitrary extreme point with maximum $u_{\mathbf{p}}$ defined as below.

$$u_{\mathbf{p}} = \min_{i\in U}(p_i)$$

where $U = \{i : p_i \neq 0\}$. W.l.o.g, let $p_i = u_{\mathbf{p}}$ and pick a $j \in [n]$ where $p_i < p_j$. To obtain a contradiction, we show that if $\mathbf{p}$ is an extreme point, then it is possible to modify p_i and p_j without changing other elements of $\mathbf{p}$ in a way that the values of $p_i + p_j$ and $F_m(\mathbf{p})$ are unchanged but p_i is increased. This gives us a contradiction since by repeating this process one can increase $u_{\mathbf{p}}$.

Let $X = \sum_{l\in n} x_l$ and $X' = X - x_i - x_j$. We have

$$F_m(\mathbf{p}) = \Pr[X' > m] + \Pr[X' = m]\Pr[x_i + x_j > 0] + \Pr[X' = m-1]\Pr[x_i + x_j = 2].$$

Define $d = (p_j - p_i)/2$ and $s = (p_i + p_j)/2$. We have $\Pr[x_i + x_j = 2] = (s+d)(s-d) =$

$s^2 - d^2$ and $\Pr[x_i + x_j > 0] = d^2 + 2s - s^2$. Let

$$G(s,d) = \Pr[X' = m](d^2 + 2s - s^2) + \Pr[X' = m-1](s^2 - d^2).$$

Therefore,

$$F_m(\mathbf{p}) = \Pr[X' > m] + G(s,d).$$

Note that $\frac{\partial G}{\partial d} = 2d(\Pr[X' = m] + \Pr[X' = m-1])$. By the assumption that $\mathbf{p}$ is an extreme point, $p_i \notin \{0,1\}$ and $p_j \notin \{0,1\}$ we obtain that $\Pr[X' = m] + \Pr[X' = m-1] = 0$. This gives us the freedom to change values of p_i and p_j and set $p_i = p_j = s$ as it does not change the value of $F_m(\mathbf{p})$. Thus, we obtain a contradiction and the proof is completed.

□

Fact 3.8.4. *Given n iid Bernoulli random variables $x_1, \ldots, x_n$ with $\mathbb{E}[x_i] = p$, if $n \to \infty$, then for any $0 \le j \le n$ we have*

$$\Pr\left[\sum_{i=1}^{n} x_i = j\right] = \frac{e^{-np}(np)^j}{j!}.$$

Proof. This is based on the relation between Poisson and Binomial distribution when $n \to \infty$. □

Lemma 3.4.9. (restated) Given $m \in \mathbb{N}$ and a random variable X that is sum of a set of independent Bernoulli random variables with $\mathbb{E}[X] = \mu$, if $m + 1 < \mu$, then we have

$$\Pr[X > m] \geq \min_{0 \leq i \leq m} G(m - i, \mu - i).$$

Proof. Let $x_1 \ldots x_n$ denote a set of independent Bernoulli random that minimize $\Pr[\sum_{i=1}^{n} x_i > m]$ subject to $\sum_{i=1}^{n} x_i = \mathbb{E}[X]$. By Lemma 3.8.3, we know that any two variables x_i and x_j that are not deterministically zero or one are identical. Let $I = \{i \in [n] : \mathbb{E}[x_i] = 1\}$. Moreover, w.l.o.g., assume none of the variables are deterministically zero. We have $\Pr[\sum_{i=1}^{n} x_i > m] = \Pr[\sum_{i \notin M} x_i > m - |I|]$. Further by Lemma 3.8.2, $\Pr[\sum_{i=1}^{n} x_i > m]$ is minimized when $n \to \infty$, which using Fact 3.8.4, leads to

$$\Pr\left[\sum_{i \notin M} x_i \leq m - |I|\right] \leq \sum_{j=0}^{m-|I|} \frac{(\mu - |I|)^j e^{-(\mu - |I|)}}{j!}.$$

Recall that we have

$$G(x, \lambda) = 1 - \sum_{i=0}^{x} \frac{\lambda^i e^{-\lambda}}{i!}.$$

Note that if $|I| > m$, then $\Pr[\sum_{i=1}^{n} x_i > m] = 1$, thus by considering all possible values of $0 \leq |I| \leq m$ we get

$$\Pr\left[\sum_{i=0}^{n} x_i \geq m\right] \geq \min_{0 \leq i \leq m} \left(1 - \sum_{i=0}^{m-i} \frac{(\mu - i)^j e^{-(\mu - i)}}{j!}\right) = \min_{0 \leq i \leq m} G(m - i, \mu - i).$$

□

Lemma 3.4.11. (restated) For any integer number $m > 2$ and any real number

$\theta \in [0, 2]$, we have

$$\min_{\boldsymbol{\mu} \in M_{2,\theta}} H(2, \boldsymbol{\mu}) \leq \min_{\boldsymbol{\mu} \in M_{m,\theta}} H(m, \boldsymbol{\mu}),$$

where $M_{m,\theta} = \{\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n) \in [0, 1]^n | \sum_{i=1}^n \mu_i = m\theta\}$, and

$$H(k, (\mu_1, \ldots, \mu_n)) = \frac{\mathbb{E}[\min(\sum_{i \in [n]} x_i, m)]}{m},$$

with $x_i, \ldots, x_n$ being independent Bernoulli random variables with means $\mu_i, \ldots, \mu_n$.

Proof. Given a $\theta \in [0, 1]$ and an arbitrary $m \geq 2$, let $\mathbf{x} = (x_1, \ldots x_n)$ be a vector of independent Bernoulli random variables with means $\mu_1, \ldots, \mu_n$ summing to $m\theta$ and let $\mathbf{y} = (y_1, \ldots y_n)$ be a vector of independent Bernoulli random variables with expectations $\frac{m-1}{m}\mu_1, \ldots, \frac{m-1}{m}\mu_n$ summing up to $(m-1)\theta$. We will prove that for any such $\mathbf{x}$ and $\mathbf{y}$ we have

$$\mathbb{E}\left[\frac{\min(\sum_{i=1}^n x_i, m)}{m}\right] \geq \mathbb{E}\left[\frac{\min(\sum_{i=1}^n y_i, m-1)}{m-1}\right]. \tag{3.25}$$

This shows that for a given m and θ,

$$\min_{\mathbf{p} \in M_{m,\theta}} H(m, \mathbf{p})$$

is an increasing function of m and as a result for any m we have

$$\min_{\mathbf{p} \in M_{2,\theta}} H(2, \mathbf{p}) \leq \min_{\mathbf{p} \in M_{m,\theta}} H(m, \mathbf{p}).$$

To achieve this, for any i, we couple x_i with y_i so that $\mathbb{E}[y_i | x_i = 0] = 0$ and $\mathbb{E}[y_i | x_i =$

$1] = \frac{m-1}{m}$, and show that

$$\mathbb{E}\left[K(\frac{m-1}{m}x_i, \ldots, \frac{m-1}{m}x_n)\right] \geq \mathbb{E}[K(y_1, \ldots, y_n)], \tag{3.26}$$

where $K(z_1, \ldots, z_n) = \min(\sum_{i=1}^{n} \frac{z_i}{m-1}, 1)$. Note that the left hand side of Equation 3.26, is equal to that of Equation 3.25 and similarly the right hand sides are equal too, thus proving the correctness of Equation 3.26 would complete the proof. Let $\bar{\mathbf{x}} =$ be an arbitrary realization of the vector of random variables $\mathbf{x}$. We show that for any value of $\bar{\mathbf{x}}$ we have

$$\mathbb{E}\left[K(\frac{m-1}{m}x_i, \ldots, \frac{m-1}{m}x_n)|\mathbf{x} = \bar{\mathbf{x}}\right] \geq \mathbb{E}[K(y_1, \ldots, y_n)|\mathbf{x} = \bar{\mathbf{x}}]. \tag{3.27}$$

We know that if at least m elements of $\bar{\mathbf{x}}$ are equal to one, then the left hand side is equal to one as well. Also, function $K(.)$ is upper bounded by one thus, if $\sum_{i=1}^{n} \bar{x}_i > m-1$, the Equation 3.27 holds. Otherwise, if at most $m-1$ elements of $\bar{\mathbf{x}}$ are equal to one then we have

$$\mathbb{E}\left[K(\frac{m-1}{m}x_i, \ldots, \frac{m-1}{m}x_n)|\mathbf{x} = \bar{\mathbf{x}}, \sum_{i=1}^{n} \bar{x}_i \leq m-1\right] = \min\left(\sum_{i=1}^{n} \frac{\bar{x}_i}{m}, 1\right) = \frac{\sum_{i=1}^{n} \bar{x}_i}{m}.$$

Recall that for any y_i we have $\Pr[y_i|x_i = 0] = 0$, which results in

$$\Pr\left[\sum_{i=1}^{n} y_i \leq m-1|\mathbf{x} = \bar{\mathbf{x}}, \sum_{i=1}^{n} \bar{x}_i \leq m-1\right] = 1.$$

This implies that

$$\mathbb{E}\left[K(y_i,\ldots,y_n)|\mathbf{x}=\bar{\mathbf{x}},\sum_{i=1}^{n}\bar{x}_i\leq m-1\right]=\mathbb{E}\left[\min\left(\sum_{i=1}^{n}\frac{y_i}{m-1},1\right)\right]$$
$$=\mathbb{E}\left[\sum_{i=1}^{n}\frac{y_i}{m-1}\right]=\frac{\sum_{i=1}^{n}\bar{x}_i}{m},$$

which completes the proof as we showed that Equation 3.27 holds for all possible realizations of $\mathbf{x}$.

□

Claim 3.4.12. (restated) Given a fixed real number $\theta\in(0,2)$, and a set of independent Bernoulli random variables $x_1,\ldots,x_n$ with $\mathbb{E}[\sum_{i\in[n]}x_i]=2\theta$ we have

$$\tfrac{1}{2}\,\mathbb{E}[\min(\sum_{i\in[n]}x_i,2)]\geq 1-(1+\theta)e^{-2\theta}.$$

Proof. Let $X=\sum_{i\in[n]}x_i$. We have

$$\mathbb{E}[\min(X,2)]=\Pr[X=1]+2\Pr[X\geq 2]=2-\Pr[X=1]-2\Pr[X=0]$$

It is easy to verify that this function is minimized when variables $x_1,\ldots,x_n$ are iid and $n\to\infty$ in which case, X is a Poisson random variable with $\lambda=2\theta$. For $X\sim$ Poisson(2θ) we have

$$\Pr[X=1]+2\Pr[X=0]=2e^{-2\theta}+2\theta^{-2\theta}=(2+2\theta)e^{-2\theta},$$

which results in

$$\frac{\mathbb{E}[\min(X,2)]}{2} = 1 - (1+\theta)e^{-2\theta}.$$

□

Lemma 3.8.5. *Given a real number* $\alpha > 1.05$*, an integer* $m \geq 2000$*, and a set of independent Bernoulli random variables* $x_i, \ldots, x_n$ *with* $\mathbb{E}[X] \geq \alpha m$ *we have*

$$\Pr[X \geq m+1] \geq 0.9,$$

where $X = \sum_{i=1}^{n} x_i$*.*

Proof. Note that for any $m \geq 2000$, we have $1.05m < 1.049(m+1)$; therefore, given that $\alpha \geq 1.05$ and $m \geq 2000$ we obtain $\mathbb{E}[X] \geq 1.049(m+1)$. By Chernoff bound, for any $\delta > 0$, we have

$$\Pr(X < (1-\delta)\mathbb{E}[X]) < \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{E[X]}.$$

In our case, we are interested in giving an upper bound for $\Pr[X < m+1]$, which is

$$\Pr[X < m+1] \leq \Pr\left[X < \frac{1}{1.049}\mathbb{E}[X]\right] < \left(\frac{e^{-0.047}}{(1-0.047)^{1-0.047}}\right)^{2000*1.05} < 0.1.$$

This gives us $\Pr[X \geq m+1] > 0.9$ and completes the proof. □

Chapter 4: Stochastic Matching

In this chapter, we consider the following *stochastic matching* problem. An arbitrary graph $G = (V, E)$ is given, then each edge $e \in E$ is retained (or to be consistent with the literature *realized*) independently with some given probability $p \in (0, 1]$. The goal is to pick a subgraph Q of G without knowing the edge realizations such that:

1. The expected size of the maximum matching among the realized edges of Q approximates the expected size of the maximum matching among the realized edges in G.
2. The maximum degree in Q is bounded by a function that may depend on p^{-1} but must be independent of the size of G.[1]

It would be useful to think of p as some constant whereas $n := |V| \to \infty$. Then the second condition translates to Q having $O(1)$ maximum degree. In other words, the subgraph Q should provide a good approximation while having $O(n)$ edges, in contrast to G which may have up to $\Omega(n^2)$ edges.

Applications. The setting is mainly motivated by applications in which the process of determining an edge realization (referred to as *querying* the edge) is considered time

[1]In this chapter, we solve a generalization of this problem where each edge e has its own realization probability p_e and the degree of Q can be proportional to $p = \min_e p_e$. See Section 5.2.1 for the formal setting.

consuming or expensive. For such applications, one can instead of querying every edge of G, only query the edges of its much sparser subgraph Q and still find a large realized matching in G. Kidney exchange and online labor markets are major examples of such applications. For more details on the role of the stochastic matching problem in these applications, see [5, 6, 12, 17, 17] (particularly [17, Section 1.2]) for kidney exchange and [9, 11, 12] for online labor markets. Another natural application of the model is that this subgraph Q can be used as a *matching sparsifier* for G which approximately preserves its maximum matching size under random edge failures [4].

Related work. The problem has received significant attention [4, 5, 6, 9, 11, 12, 17, 41] after the pioneering work of Blum *et al.* [17] who proved that it admits a $(\frac{1}{2} - \varepsilon)$-approximation. Earlier follow-up works revolved around the prevalent half-approximation barrier until it was first broken by Assadi *et al.* [5]. This was followed by a 0.6568-approximation by Behnezhad *et al.* [11] and eventually a $(\frac{2}{3} - \varepsilon)$-approximation by Assadi and Bernstein [4] which is the state-of-the-art. See also [11, 12, 35, 41] for various natural generalizations of the problem.

Our result. Below we state the main contribution of this chapter:

Theorem 5. *For any $\varepsilon > 0$, there is an algorithm that picks an $O_{\varepsilon,p}(1)$-degree subgraph Q of G such that the expected size of the maximum realized matching in Q is at least $(1 - \varepsilon)$ times the expected size of the maximum realized matching in G.*

To get a $(1 - \varepsilon)$-approximation, the dependence of the maximum degree of Q on both ε and p is necessary. Particularly, a simple lower bound shows that even when G is a clique, to avoid too many singleton vertices in a realization of Q, the maximum degree in Q must be $\Omega(\frac{\ln \varepsilon^{-1}}{p})$ [5]. The same lower bound also shows that a $(1 - o(1))$ approximation is not achievable unless the maximum degree of Q is $\omega(1)$, meaning that our approximation-factor is essentially the best one can hope for.

Remark 4.0.1. *In Theorem 6, $O_{\varepsilon,p}(1)$ is in the order*

$$\exp\Big(\exp\Big(\exp\big(O\big(\varepsilon^{-1}\big)\big) \times \log\log p^{-1}\Big)\Big).$$

We do not believe this dependence is optimal and leave it as an open problem to improve it. Particularly, we conjecture that the same algorithm that is analyzed in this chapter (see Algorithm 2) should obtain up to $(1 - \varepsilon)$-approximation even by picking only a $\text{poly}(1/\varepsilon p)$*-degree subgraph.*

The algorithm. Many different constructions of Q have been studied in the literature. A well-studied algorithm first considered by Blum *et al.* [17] which was further analyzed (module minor differences and generalizations) in the subsequent works of [5, 6, 12, 35, 41] is as follows: Iteratively pick a maximum matching M_i from G, remove its edges, and finally let $Q = M_1 \cup \ldots \cup M_R$ for some parameter R that controls the maximum degree in Q. Despite the positive results proved for this algorithm, it was already shown in [17] that its approximation-factor is not better than 5/6. Thus to obtain $(1 - \varepsilon)$-approximation, one has to use a different algorithm.

We focus on an algorithm proposed previously by Behnezhad *et al.* [11], which they proved obtains at least a 0.6568-approximation. The algorithm is equally simple, but subtly different: Draw R independent realizations $\mathcal{G}_1, \ldots, \mathcal{G}_R$ of G and let $Q =$

$\mathsf{MM}(\mathcal{G}_1) \cup \ldots \cup \mathsf{MM}(\mathcal{G}_R)$ where $\mathsf{MM}(\mathcal{G}_i)$ is a maximum matching of $\mathcal{G}_i$. Our main result is obtained via providing a different analysis of this algorithm. Within the next two paragraphs, we discuss how our analysis differs substantially from the previous approaches and in particular from the analysis of [11].

The analysis and the Ruzsa-Szemerédi barrier. A major barrier to overcome in order to prove existence of a $(1-\varepsilon)$-approximate subgraph was already discussed in the work of Assadi, Khanna, and Li [5, Section 6] based on Ruzsa-Szemerédi graphs [2, 26, 28, 39] which we henceforth call the "RS-barrier". Consider an extension of the stochastic matching setting where the realization of edges in a single a-priori known matching M of G can be correlated while other edges are still realized independently. An implication of the RS-barrier is that in this extended model, no algorithm can obtain $(1-\varepsilon)$-approximation (or even beat $\frac{2}{3}$-approximation[2]) unless Q has maximum degree $n^{\Omega(1/\log\log n)} = \omega(\operatorname{polylog} n)$. Put differently, this proves that in order to beat $\frac{2}{3}$-approximation, the analysis has to use the fact that *every* edge around a vertex is realized independently. This explains why the previous arguments were short of bypassing $\frac{2}{3}$-approximation: They can all (to our knowledge) be adapted to tolerate adversarial realization of one edge per vertex.

"Vertex-independent matchings" to the rescue. We overview our analysis soon in Section 5.1. However, here we briefly mention our key analytical tool in bypassing the RS-barrier. It is an algorithm (Lemma 4.3.8) for constructing a matching Z on the realized *crucial* edges (roughly, an edge is crucial if it has a sufficiently high probabil-

[2]The original proof of [5] rules out $> \frac{6}{7}$-approximation. A similar instance can rule out $\frac{2}{3}$-approximation using a more efficient construction of RS-graphs [28] and allowing a subset of edges of G to have realization probability 1.

ity of being part of an optimal realized matching). The algorithm constructs Z such that among some other useful properties, it guarantees that each vertex is matched independently from all but $O(1)$ other vertices. Here the independence is with regards to both the randomization of the algorithm in constructing Z, and also importantly the edge realizations of G. This independence property is the key that separates the stochastic matching model from the extended model of the RS-barrier: Due to the added correlations in the edge realizations, such vertex-independent matchings essentially do not exist in the model of the RS-barrier. Using this independence, we show that Z can be well-augmented by the rest of the realized edges in Q. See Section 5.1 for a more detailed overview of our analysis and how the independence property helps.

Our method of bypassing the RS-barrier via vertex-independent matchings sheds more light on the limitations imposed by Ruzsa-Szemerédi type graphs. These graphs are known to be notoriously hard examples in various other areas such as property testing, streaming algorithms, communication complexity, and additive combinatorics among others [2, 26, 28, 32, 39]. As such, we believe that this method may find applications beyond the stochastic matching problem.

Organization of the chapter. In Section 5.1 we provide an informal overview of our analysis. In Section 5.2.1 we formally state the problem and the notations used throughout the chapter. In Section 5.2 we describe the algorithm and basic definitions that we will use throughout the analysis. In Section 4.4 we prove how the vertex-independent matching lemma leads to a $(1-\varepsilon)$-approximation and in Section 4.5, we prove the vertex-independent matching lemma. Finally, Section 4.9 contains the proofs of (less important) statements that are deferred.

4.1 Technical Overview

As previously described, we consider the following algorithm for constructing subgraph Q (see also Algorithm 2): Draw R realizations $\mathcal{G}_1, \ldots, \mathcal{G}_R$ of graph G, then pick a matching $\mathsf{MM}(\mathcal{G}_i)$ from each realization, and finally set $Q = \mathsf{MM}(\mathcal{G}_1) \cup \ldots \cup \mathsf{MM}(\mathcal{G}_R)$. In this section, we give an informal overview of our analysis for this algorithm.

Note that these realizations $\mathcal{G}_i$ are part of the randomization of the algorithm and may be very different from the actual realization $\mathcal{G}$ of G. In fact, in expectation, only p fraction of the edges of each matching $\mathsf{MM}(\mathcal{G}_i)$ are realized in $\mathcal{G}$. Thus, we have to argue that the realized edges of these matchings can be used to augment each other and form a large matching in the realized subgraph $\mathcal{Q}$ of Q. In order to do this, we will give a "procedure" to construct a matching in $\mathcal{Q}$. To get a handle on the dependencies involved, the procedure carefully decides how the realization of edges in Q are revealed and which are chosen to be in the matching. We emphasize that this procedure is merely an analytical tool for analyzing the approximation-factor. Thus, no matter how intricate it is, the algorithm for constructing Q remains to be the simple Algorithm 2 described above.

A crucial/non-crucial decomposition. Similar to [11] (and also implicitly [6]), we consider a partitioning of the edges of G into what we call *crucial* and *non-crucial* edges. For each edge e, define $q_e := \Pr[e \in \mathsf{MM}(\mathcal{G})]$ where $\mathsf{MM}(\cdot)$ is the same matching algorithm used to construct Q. We further assume that $\mathsf{MM}(\cdot)$ is deterministic, so the probability is taken only over the realization $\mathcal{G}$. For two thresholds $0 < \tau_- < \tau_+ < 1$ that we fix later, we define:

- The crucial edges as $C := \{e \in E \mid q_e \geq \tau_+\}$.
- The non-crucial edges as $N := \{e \in E \mid q_e \leq \tau_-\}$.

Note that in the decomposition above edges e with $q_e \in (\tau_-, \tau_+)$ are neither crucial nor non-crucial. We will essentially "ignore" these edges in the analysis but ensure that we choose τ_- and τ_+ such that there are few ignored edges.

In our procedure to construct a matching on $\mathcal{Q}$, we treat crucial and non-crucial edges differently. We start with the crucial edges and (in Lemma 4.3.8) construct a matching Z on them whose expected size is (almost) as large as the expected number of crucial edges in the optimal maximum realized matching of G. We then show that this matching Z can be augmented via the non-crucial edges to eventually form a matching whose expected size is arbitrarily close to $\textsc{opt} := \mathbb{E}[|\mathsf{MM}(\mathcal{G})|]$.

The procedure for crucial edges. In addition to the lower bound on the expected size of Z, we make sure that no vertex tends to be "over-matched" in Z. More formally, the probability of any vertex v being matched in Z should not be larger than the probability that v is matched via a crucial edge in $\mathsf{MM}(\mathcal{G})$. Both of these conditions can actually be satisfied by a very simple randomized procedure: Reveal the whole realization $\mathcal{C}$ of C, also draw a random realization $\mathcal{N}'$ of the non-crucial edges, and let Z be the crucial edges in matching $\mathsf{MM}(\mathcal{C} \cup \mathcal{N}')$.

Unfortunately, the matching constructed via the above-mentioned procedure is hard to augment via the non-crucial edges as we have no control over the correlations.

To get around this, we need an extra "independence" property. Let X_v be the indicator of the event that vertex v is matched in Z. The independence property requires random variables $X_{v_1}, X_{v_2}, \ldots, X_{v_n}$ to be (almost) independent where $\{v_1, \ldots, v_n\}$ is the vertex-set of G. Clearly, perfect independence cannot be achieved: Given the event that a vertex v is matched in Z, we derive that at least one of its neighbors in C is also matched. What we prove can be achieved, though, is that each X_v is independent from X_u of vertices u outside a small local neighborhood of v in graph C. (See Lemma 4.3.8 part 4 for the formal statement.)

In order to satisfy the independence property described above, we will not reveal the whole realization $\mathcal{C}$ outright and then construct Z based on it as it was done in the simple procedure described above. Instead, we present a different algorithm (Algorithm 2) for constructing this matching Z. To prove the independence property, we show that this algorithm can be simulated locally. In other words, for each vertex v, the value of X_v can be determined uniquely by having the realization of edges in a small local neighborhood of v. Thus, if two vertices u and v are sufficiently far from each other in graph C, then X_v and X_u would be independent.

Augmenting Z via non-crucial edges. We noted above that $\mathbb{E}[|Z|]$ is (almost) as large as the expected number of crucial edges in $\mathsf{MM}(\mathcal{G})$. Therefore, in order to construct a matching of $\mathcal{Q}$ with expected size arbitrarily close to OPT, we have to augment Z via the non-crucial edges. To do this, we only use non-crucial edges $\{u, v\}$ in Q such that X_u and X_v are independent. Describing how exactly we construct the matching on these non-crucial edges requires a number of definitions which we give in

Section 4.4.1. However, to convey the key intuition, here we only mention how and why the independence of X_u and X_v plays an important role in using a non-crucial edge $e = \{u, v\}$ to augment Z. Suppose that $\Pr[X_u] = \Pr[X_v] = 1/2$. Note that it is only when *both* u and v are unmatched in Z that we can use edge e to augment Z. If X_u and X_v are independent, there is a relatively large probability $(1-\Pr[X_u])(1-\Pr[X_v]) = \frac{1}{4}$ that this occurs. However, if X_u and X_v can be correlated, it may be the case that with probability half $X_u = 1$ and $X_v = 0$, and with probability half $X_u = 0$ and $X_v = 1$. In this case, the probability of both u and v being unmatched in Z would be zero and thus we would never be able to use e to augment Z. We remark that this is precisely the type of correlation introduced in the RS-barrier of [5] which the independence property allows us to bypass.

4.2 Preliminaries

General notations. We denote the maximum matching size of any graph G by $\mu(G)$. For a matching M, we use $V(M)$ to denote the set of vertices matched in M. For any two nodes u and v in a graph G, we use $d_G(u, v)$ to denote their distance, i.e. the number of edges in their shortest path. Furthermore, the distance $d_G(u, e)$ between an edge e and a node u is the minimum distance between an endpoint of e and u. We use $\mathbb{1}(A)$ as the *indicator* of an event A, i.e. $\mathbb{1}(A) = 1$ if event A occurs and $\mathbb{1}(A) = 0$ otherwise. Also, we may use $[k] := \{1, 2, \ldots, k\}$ for any integer $k \geq 1$.

Throughout the thesis, we define various functions of form $x : E \to [0, 1]$ that map each edge $e \in E$ to a real number in $[0, 1]$. Having such function x, for any vertex

v we define $x_v := \sum_{e \ni v} x_e$, for any edge subset F we define $x(F) := \sum_{e \in F} q_e$, and for any vertex subset U we define $x(U) := \sum_{e=\{u,v\}:u,v \in U} x_e$. We also denote $|x| = \sum_e x_e$.

The setting. We consider a generalized variant of the standard stochastic matching problem studied in the literature where each edge e has a realization probability p_e that may be different from that of other edges. We then let $p = \min_e p_e$, which is the parameter the degree of subgraph Q can depend on. This generalization will actually help in solving the original model of the literature which coincides with the case where $p_e = p$ for every edge e.

We denote realizations by script font; for instance, we use $\mathcal{G} = (V, \mathcal{E})$ to denote the realized subgraph of the input graph G, which includes each edge e independently with probability p_e. Similarly, we use $\mathcal{Q}$ to denote the realized subgraph of Q. The same notation also naturally extends to denote realization of other subgraphs of G that we may later define.

As we discussed previously, the goal is to pick a sparse subgraph Q of G such that the ratio $\mathbb{E}[\mu(\mathcal{Q})]/\mathbb{E}[\mu(\mathcal{G})]$, known as the approximation-factor, is large. Here the expectations are taken over the realizations $\mathcal{Q}$ and $\mathcal{G}$, and possibly the randomization of the algorithm in constructing subgraph Q. For brevity, we use OPT to denote $\mathbb{E}[\mu(\mathcal{G})]$. Note that OPT is just a number.

We note that the *expected* approximation-factor defined above can automatically be turned into *high-probability* due to a simple concentration bound. See Appendix 4.6.

4.3 Basic Definitions and The Algorithm

The algorithm that we analyze is formally stated as Algorithm 2.

Algorithm 1 ([11]). A sampling-based non-adaptive algorithm for stochastic matching.

Parameter: R, which controls the maximum degree of Q.

Take R realizations $\mathcal{G}_1, \ldots, \mathcal{G}_R$ of G independently where each realization $\mathcal{G}_i$ includes each edge e independently with probability p_e. Return subgraph $Q = \mathsf{MM}(\mathcal{G}_1) \cup \ldots \cup \mathsf{MM}(\mathcal{G}_R)$.

In the algorithm above, $\mathsf{MM}(\mathcal{G}_i)$ returns a maximum matching of $\mathcal{G}_i$. It will be convenient for the analysis to assume $\mathsf{MM}(\cdot)$ is a deterministic maximum matching algorithm.

In order to analyze Algorithm 2, we will make the following assumption which will simplify many of our arguments.

Assumption 4.3.1. $\textsc{opt} \geq 0.1\varepsilon n$.

Assumption 4.3.1 comes w.l.o.g. due to a reduction of Assadi *et al.* [5]. The reduction is roughly as follows: If $n \gg \textsc{opt}$, randomly put nodes of G into $O(\frac{\textsc{opt}}{\varepsilon})$ buckets and contract the nodes within each bucket. The resulting graph will have only $O(\frac{\textsc{opt}}{\varepsilon})$ nodes but its expected maximum realized matching will be as large as $(1-O(\varepsilon))\textsc{opt}$. Solving this modified graph will then solve the original graph G as well. We provide further details in Appendix 4.7 and note that for the reduction to work, it is important that our algorithm can handle different edge realization probabilities.

4.3.1 A Crucial/Non-crucial Decomposition

For each edge e define $q_e := \Pr[e \in \mathsf{MM}(\mathcal{G})]$ where $\mathsf{MM}(\cdot)$ is the same matching algorithm used in Algorithm 2. Since we assumed $\mathsf{MM}(\cdot)$ is deterministic, the probability is taken only over the randomization of the realization $\mathcal{G}$. Having this definition, for any vertex v we denote $q_v := \sum_{e \ni v} q_e$ and for any subset $E' \subseteq E$ denote $q(E') := \sum_{e \in E'} q_e$. The following statements immediately follow from the definition:

Observation 4.3.2. $q(E) = \text{OPT}$.

Observation 4.3.3. *For any vertex v, q_v denotes the probability that v is matched in $\mathsf{MM}(\mathcal{G})$.*

We will fix two thresholds $0 < \tau_- < \tau_+ < 1$ that both depend only on ε and p. Next, for any edge e, we say e is *crucial* if $q_e \geq \tau_+$, *non-crucial* if $q_e \leq \tau_-$, and *ignored* if $q_e \in (\tau_-, \tau_+)$. We denote the crucial edges by $C := \{e \in E \mid e \text{ is crucial}\}$, and the non-crucial edges by $N := \{e \in E \mid e \text{ is non-crucial}\}$. Furthermore, we denote their realizations by $\mathcal{C} := C \cap \mathcal{E}$ and $\mathcal{N} := N \cap \mathcal{E}$. When confusion is impossible, we may use C to denote graph (V, C) instead of merely the edge-subset. The same also naturally generalizes to N, $\mathcal{C}$, and $\mathcal{N}$. We will further use Δ_C to denote the maximum degree in graph C. Moreover, for any vertex v we use c_v (resp. n_v) to denote the probability that v is matched via a crucial (resp. non-crucial) edge in $\mathsf{MM}(\mathcal{G})$.

Observation 4.3.4. $\Delta_C \leq 1/\tau_+$.

Proof. Each edge $e \in C$ has $q_e \geq \tau_+$ by definition. Thus, if there is a vertex v of degree larger than $1/\tau_+$ in C, then it should hold that $q_v > 1/\tau_+ \times \tau_+ = 1$ which contradicts

Observation 4.3.3. □

4.3.2 Setting the Thresholds τ_- and τ_+

To describe how we set the values of τ_- and τ_+, we state a lemma that we prove in Section 4.9.

Lemma 4.3.5. *Fix any arbitrary function* $f(x)$ *such that* $0 < f(x) < x$ *for any* $0 < x < 1$. *There is a choice of* $0 < \tau_- < \tau_+ < 1$ *such that: (1)* $\tau_- = f(\tau_+)$. *(2)* $q(N) + q(C) \geq (1 - \varepsilon)\textsc{opt}$. *(3) Both* τ_- *and* τ_+ *depend only on* ε *and* p. *And finally, (4)* $\tau_+ \leq (\varepsilon p)^{50}$.

The lemma above essentially shows that we can have any desirably large gap between τ_+ and τ_- and still ensure that $q(N) + q(C) \geq (1 - \varepsilon)\textsc{opt}$. That is, the ignored edges in expectation constitute at most $\varepsilon\textsc{opt}$ edges of $\mathsf{MM}(\mathcal{G})$. While this may sound counter-intuitive, it follows roughly speaking from the fact that by iteratively reducing the threshold τ_+ by a sufficient amount, all the previously ignored edges become crucial. Thus it cannot continue to hold that there are still a significant mass of the matching on the ignored edges after sufficiently many iterations. See Section 4.9 for the proof.

Having Lemma 4.3.5, we set our thresholds and the parameter R of Algorithm 2 as follows:

Setting τ_-, τ_+, and R:

Define function $f(x) := x^{10g(x)}$ where $g(x) := \varepsilon^{-20} \log \frac{1}{x}$.

We plug this function f into Lemma 4.3.5 and define τ_- and τ_+ accordingly. We also set $R = \frac{1}{2\tau_-}$.

Note that function f as defined above satisfies $0 < f(x) < x$ for any $0 < x < 1$ since clearly $g(x) \geq 1$ so long as $0 < x < 1$. Therefore, we can indeed plug f into Lemma 4.3.5. This results in the following properties:

Corollary 4.3.6. *It holds that: (1) $\tau_- = (\tau_+)^{10g}$ where $g = \varepsilon^{-20} \log \frac{1}{\tau_+}$. (2) $q(N) + q(C) \geq (1-\varepsilon)\text{OPT}$. (3) Both τ_- and τ_+ depend only on ε and p and thus $R = O_{\varepsilon,p}(1)$. (4) $\tau_- < \tau_+ \leq (\varepsilon p)^{50}$.*

The next lemma shows that R is set such that Algorithm 2 samples (almost) all crucial edges.

Observation 4.3.7. *For every edge $e \in C$, $\Pr[e \in Q] \geq 1 - \varepsilon$.*

Proof. Note that $e \in Q$ if there is at least one $i \in [R]$ where $e \in \mathsf{MM}(\mathcal{G}_i)$. The probability that $e \in \mathsf{MM}(\mathcal{G}_i)$ for any fixed i is precisely q_e. Since realizations $\mathcal{G}_1, \ldots, \mathcal{G}_R$ are independent, it holds that $\Pr[e \notin Q] = (1 - q_e)^R$. On the other hand $q_e \geq \tau_+$ since e is crucial. Also $R = \frac{1}{2\tau_-} > \ln \varepsilon^{-1}/\tau_+$ where the latter inequality follows easily from Corrolary 4.3.6 part (1). Combining all of these gives:

$$\Pr[e \notin Q] = (1 - q_e)^R < (1 - \tau_+)^{\ln \varepsilon^{-1}/\tau_+} < e^{-\ln \varepsilon^{-1}} = \varepsilon.$$

Therefore indeed $\Pr[e \in Q] \geq 1 - \varepsilon$. □

4.3.3 The Vertex-Independent Matching Lemma

As discussed before, a key technical contribution of this chapter that allows getting an arbitrary good approximation-factor is a "vertex-independent matching" lemma that we state here. The proof of this lemma is involved and thus we defer it to Section 4.5. In Section 4.4, we show how Lemma 4.3.8 can be used to analyze Algorithm 2 and prove Theorem 6.

Lemma 4.3.8 (Vertex-Independent Matching Lemma)**.** *There is a randomized algorithm that constructs an integral matching Z of $\mathcal{C}$ (the realized subgraph of C) such that defining X_v as the indicator random variable for $v \in V(Z)$, we get:*

1. $\mathbb{E}[|Z|] \geq q(C) - 30\varepsilon\text{OPT}$.
2. *For every vertex v, $\Pr[X_v] \leq \max\{c_v - \varepsilon^2, 0\}$, where recall that c_v is the probability that vertex v is matched via a crucial edge in $\mathsf{MM}(\mathcal{G})$.*
3. *The matching Z is independent of the realization of non-crucial edges in G.*
4. *Let $\lambda := \varepsilon^{-20} \log \Delta_C$. For every k and every $\{v_1, v_2, \ldots, v_k\} \subseteq V$ such that $d_C(v_i, v_j) \geq \lambda$ for all $v_i \neq v_j$, random variables $X_{v_1}, \ldots, X_{v_k}$ are independent.*

We emphasize that $\mathbb{E}[|Z|]$ and X_v are both defined with respect to the randomizations in both the realization of C, and the randomization of the algorithm in constructing Z.

Observation 4.3.9. *Let g be as defined in Corollary 4.3.6 and λ be as defined in Lemma 4.3.8. Then it holds that $g \geq \lambda$.*

Proof. Since $\lambda = \varepsilon^{-20} \log \Delta_C$ by definition and $\Delta_C \leq 1/\tau_+$ by Observation 4.3.4, we get that $\lambda \leq \varepsilon^{-20} \log \frac{1}{\tau_+}$. On the other hand $g = \varepsilon^{-20} \log \frac{1}{\tau_+}$. Therefore, $g \geq \lambda$. □

4.4 The Analysis via the Vertex-Independent Matching Lemma

In this section, given correctness of Lemma 4.3.8, we prove Theorem 6. In what follows we give the outline of the proof by referring to the needed lemmas that will be proved in subsequent Sections 4.4.1, 4.4.2, 4.4.3, and 4.4.4.

Proof Outline for Theorem 6. Let Q be the output of by Algorithm 2 where parameter R is set as described above. We show that one can construct a matching of expected size at least $(1 - 56\varepsilon)\text{OPT}$ on the realized subgraph $\mathcal{Q}$ of Q. This implies that $\mathbb{E}[\mu(\mathcal{Q})] \geq (1 - 56\varepsilon)\text{OPT} = (1 - 56\varepsilon)\mathbb{E}[\mu(\mathcal{G})]$. In other words, this proves that the approximation-factor of the algorithm is at least $(1 - 56\varepsilon)$. (Note this is equivalent to $(1 - \varepsilon)$ approximation since one can choose ε to be any desirably small constant.)

In order to construct a matching of expected size at least $(1 - 56\varepsilon)\text{OPT}$ on $\mathcal{Q}$, we first describe how to construct an "expected fractional matching" (see Definition 4.4.1) x on $\mathcal{Q}$ in Sections 4.4.1, 4.4.2, and 4.4.3. Later on, we show in Section 4.4.4 how to turn x into a fractional matching y on $\mathcal{Q}$ such that $\mathbb{E}[|y|] \geq (1-55\varepsilon)\text{OPT}$ (see Lemma 4.4.11). Finally, to turn y into an *integral* matching, we show (Observation 4.4.10) that the so called "blossom inequalities" of size up to $1/\varepsilon$ also hold for y. That is, we show that for all vertex subsets $U \subseteq V$ with $|U| \leq 1/\varepsilon$, we have $y(U) \leq \lfloor \frac{|U|}{2} \rfloor$. By Edmond's celebrated theorem [25, 40] on the matching polytope, this means that there is an integral matching of size at least $\frac{1}{1+\varepsilon}|y| \geq (1 - \varepsilon)|y|$ in $\mathcal{Q}$. As described, $\mathbb{E}[|y|] \geq$

$(1-55\varepsilon)$OPT, thus indeed $\mathbb{E}[\mu(\mathcal{Q})] \geq (1-\varepsilon)(1-55\varepsilon)$OPT $\geq (1-56\varepsilon)$OPT as desired.

4.4.1 Construction of an Expected Fractional Matching x on $\mathcal{Q}$

In this section, we describe an algorithm that constructs an "expected fractional matching" x on $\mathcal{Q}$ as defined below.

Definition 4.4.1. *Let $\mathcal{A}$ be a random process that assigns a fractional value $x_e \in [0,1]$ to each edge e of a graph $G(V,E)$. We say x is an expected fractional matching if:*

1. *For each vertex v, defining $x_v := \sum_{e \ni v} x_e$ we have $\mathbb{E}[x_v] \leq 1$.*
2. *For all subsets $U \subseteq V$ with $|U| \leq 1/\varepsilon$, $x(U) \leq \lfloor \frac{|U|}{2} \rfloor$ with probability 1.*

We emphasize that the definition only requires $\mathbb{E}[x_v] \leq 1$, thus depending on the coin tosses of the process, it may occur that $x_v > 1$, violating the constraints of a normal fractional matching. We will later argue that in our construction, the values of x_v's are sufficiently concentrated around their mean and thus we can turn our expected fractional matching to an actual fractional matching of (almost) the same size.

As described before, we construct an expected fractional matching x on the edges of graph $\mathcal{Q}$. Note that here the graph $\mathcal{Q}$ itself is also stochastic. In the construction, we treat crucial and non-crucial edges completely differently.

Crucial edges. On the crucial edges, we first construct an integral matching Z using the algorithm of Lemma 4.3.8. Once we have Z, we define x on crucial edges as follows.

$$\text{For every crucial edge } e, \qquad x_e := \begin{cases} 1, & \text{if } e \in Z \text{ and } e \in Q, \\ 0, & \text{otherwise.} \end{cases} \tag{4.1}$$

Note from Observation 4.3.7 that each crucial edges belong to Q with probability at least $1 - \varepsilon$. Therefore the construction above (roughly speaking) sets $x_e = 1$ for most of the edges e in Z.

Non-crucial edges. For defining x on the non-crucial edges, we start with a number of useful definitions. For any edge e, define t_e to be the number of matchings $\mathsf{MM}(\mathcal{G}_1), \ldots, \mathsf{MM}(\mathcal{G}_R)$ that include e. Then based on that, define

$$f_e := \begin{cases} \frac{t_e}{R}, & \text{if } \frac{t_e}{R} \leq \frac{1}{\sqrt{\varepsilon R}} \text{ and } e \text{ is non-crucial,} \\ 0, & \text{otherwise.} \end{cases} \tag{4.2}$$

Note that f_e is a random variable of only the randomization of Algorithm 2, i.e. it is independent of the realization. Also note that f_e is desirably non-zero only on the edges that belong to graph Q. Having defined f_e, we define x_e on the non-crucial edges as follows.

For every non-crucial edge e, define

$$x_e = \begin{cases} \frac{f_e}{p_e(1-\Pr[X_v])(1-\Pr[X_u])}, & \text{if } e \text{ is realized, } u, v \notin V(Z)\text{, and } d_C(u,v) \geq \lambda, \\ 0, & \text{otherwise.} \end{cases} \tag{4.3}$$

We note that λ in the definition above is the number defined in Lemma 4.3.8 and that X_v is the indicator random variable for the event $v \in V(Z)$.

Before concluding this section, let $f_v := \sum_{e \in N : v \in e} f_e$ for each vertex v. We note the following properties of f, which can be derived directly from the definition above. The proof is given in Section 4.9.

Claim 4.4.2. *It holds that:*

1. *For every non-crucial edge e, $\mathbb{E}[f_e] \leq q_e$.*
2. *For every non-crucial edge e, $\mathbb{E}[f_e] \geq (1-\varepsilon)q_e$.*
3. *For every vertex v, it always holds that $\sum_{e \ni v} f_e \leq 1$.*
4. *For every vertex v, $\Pr[f_v > n_v + 0.1\varepsilon] \leq (\varepsilon p)^{10}$, where recall that n_v is the probability that v is matched via a non-crucial edge in $\mathsf{MM}(\mathcal{G})$.*

Consider a non-crucial edge $\{u, v\}$ between two nodes u and v with $d_C(u, v) \geq \lambda$. The probability that x_e is non-zero is $p_e(1-\Pr[X_v])(1-\Pr[X_u])$: Both u and v should be unmatched in Z and e should be realized, and further all these events are independent. This intuitively explains why we set $x_e = \frac{f_e}{p_e(1-\Pr[X_v])(1-\Pr[X_u])}$ if all these conditions hold: We want the denominator to cancel out with this probability so that we get $\mathbb{E}[x_e] = f_e$. We will formalize this intuition in Section 4.4.3 where we prove the expected size of x is large.

4.4.2 Validity of x

In this section, we prove that x is indeed an expected fractional matching of $\mathcal{Q}$.

First, we prove that x is non-zero only on the edges of $\mathcal{Q}$. This simply follows from the construction of x.

Claim 4.4.3. *Any edge e with $x_e > 0$ belongs to $\mathcal{Q}$. That is, x is only non-zero on the set of edges queried by Algorithm 2 that are also realized.*

Proof. For any crucial edge e, we either have $x_e = 1$ or $x_e = 0$. By definition, if $x_e = 1$ then $e \in Z \cap Q$. By Lemma 4.3.8, Z is a matching of *realized* crucial edges, i.e. $e \in Z$ implies $e \in \mathcal{E}$. Therefore, $e \in Z \cap Q$ implies $e \in \mathcal{E} \cap Q = \mathcal{Q}$ as desired.

For any non-crucial edge e, if $e \notin Q$, then $f_e = 0$ by definition of f_e. Therefore, if $x_e > 0$, then $f_e > 0$ which implies $e \in Q$. Moreover, by (4.3), $x_e > 0$ implies e is realized. Combining these two, we get that if $x_e > 0$ then $e \in \mathcal{Q}$. □

Next, we prove condition (1) of Definition 4.4.1.

Claim 4.4.4. *For every vertex v, $\mathbb{E}[x_v] \leq 1$.*

Proof. Suppose at first that there is an edge e incident to v that belongs to matching Z. Then we either have $x_e = 1$ or $x_e = 0$ (depending on whether $e \in Q$ or not). For all other edges e' connected to v (crucial or non-crucial) we have $x_{e'} = 0$ by (4.1) and (4.3). Therefore if such edge e exists, we indeed have $x_v \leq 1$. For the rest of the proof, we condition on the event that no such edge e exists, i.e. $v \notin V(Z)$ and prove the claim.

Let $u_1, u_2, \ldots, u_r$ be neighbors of v in graph G such that for all $i \in [r]$: (1) edge $\{v, u_i\}$ is non-crucial, (2) $d_C(v, u_i) \geq \lambda$. Let $e_i := \{v, u_i\}$; we claim that conditioned on $v \notin V(Z)$, we have

$$x_v = x_{e_1} + x_{e_2} + \ldots + x_{e_r}. \tag{4.4}$$

To see this, fix an edge $e = \{v, u\}$ for some $u \notin \{u_1, \ldots, u_r\}$. We show that $x_e = 0$, which suffices to prove (4.4). First if e is crucial, then $e \notin Z$ given that $v \notin V(Z)$; thus according to (4.1) we set $x_e = 0$. Moreover, if e is non-crucial, the assumption $u \notin \{u_1, \ldots, u_r\}$ implies $d_C(v, u) < \lambda$ by definition of the set. In this case also, we set $x_e = 0$ according to (4.3); concluding the proof of (4.4).

By linearity of expectation applied to (4.4), we get

$$\mathbb{E}[x_v \mid v \notin V(Z)] = \sum_{i=1}^{r} \mathbb{E}[x_{e_i} \mid v \notin V(Z)]. \tag{4.5}$$

Moreover, for any arbitrary $i \in [r]$ we have

$$\begin{aligned}\mathbb{E}[x_{e_i} \mid v \notin V(Z)] &= \Pr[u_i \notin V(Z), e_i \text{ realized} \mid v \notin V(Z)] \times \frac{\mathbb{E}[f_{e_i}]}{p_{e_i}(1 - \Pr[X_v])(1 - \Pr[X_{u_i}])} \\ &= p_{e_i}(1 - \Pr[X_{u_i}]) \times \frac{\mathbb{E}[f_{e_i}]}{p_{e_i}(1 - \Pr[X_v])(1 - \Pr[X_{u_i}])} = \frac{\mathbb{E}[f_{e_i}]}{1 - \Pr[X_v]}.\end{aligned} \tag{4.6}$$

The second equality above follows from the fact that the event of e_i being realized is independent of u_i or v being in $V(Z)$, as indicated by Lemma 4.3.8 part 3; and also the fact that $u_i \notin V(Z)$ and $v \notin V(Z)$ are also independent from each other due to Lemma 4.3.8 part 4 combined with the assumption that $d_C(u_i, v) \geq \lambda$. We also note that we have used $\mathbb{E}[f_{e_i}]$ instead of $\mathbb{E}[f_{e_i} \mid v \notin V(Z)]$ in the equation above since f_{e_i} is only a random variable of the randomization used in Algorithm 2 whereas the matching

Z is constructed in Lemma 4.3.8 independent of the outcome of Algorithm 2.

Combining (4.5) and (4.6) we get

$$\mathbb{E}[x_v \mid v \notin V(Z)] = \sum_{i=1}^{r} \frac{\mathbb{E}[f_{e_i}]}{1 - \Pr[X_v]} = \frac{1}{1 - \Pr[X_v]} \sum_{i=1}^{r} \mathbb{E}[f_{e_i}]. \tag{4.7}$$

From Claim 4.4.2 part 1, we know $\mathbb{E}[f_{e_i}] \leq q_{e_i}$. Replacing this into the equality above, we get

$$\mathbb{E}[x_v \mid v \notin V(Z)] \leq \frac{1}{1 - \Pr[X_v]} \sum_{i=1}^{r} q_{e_i} \leq \frac{n_v}{1 - \Pr[X_v]}.$$

Lemma 4.3.8 part (2) guarantees that $\Pr[X_v] < c_v$ which implies $1 - \Pr[X_v] > 1 - c_v$. On the other hand, $c_v + n_v$ is upper bounded by the probability that v is matched in OPT, thus $c_v + n_v \leq 1$, implying $n_v \leq 1 - c_v$. These, combined with the equation above, gives

$$\mathbb{E}[x_v \mid v \notin V(Z)] \leq \frac{n_v}{1 - \Pr[X_v]} \leq \frac{1 - c_v}{1 - c_v} = 1.$$

Recalling also that $\mathbb{E}[x_v \mid v \in V(Z)] \leq 1$ as described at the start of the proof, this concludes the proof of the claim that $\mathbb{E}[x_v] \leq 1$. $\square$

Next, we show that condition (2) of Definition 4.4.1 also holds for our construction.

Claim 4.4.5. *For all subsets $U \subseteq V$ with $|U| \leq 1/\varepsilon$, $x(U) \leq \lfloor \frac{|U|}{2} \rfloor$ with probability 1.*

Proof. By definition of x, the value of x_e on crucial edges is either 1 or 0. Moreover, the definition also implies that if a vertex v is incident to a crucial edge e with $x_e = 1$, for all other edges e' incident to v we have $x_{e'} = 0$. Call all such vertices *integrally matched*. Fix a subset U and let U' be the subset of U excluding its integrally matched vertices. One can easily confirm that if $x(U) > \lfloor |U|/2 \rfloor$, then also $x(U') > \lfloor |U'|/2 \rfloor$. Therefore, either the claim holds, or there should exist a subset with no integrally matched vertices that violates it. Let U be the smallest such subset and observe that $|U| \leq 1/\varepsilon$ (otherwise U does not contradict the claim's statement).

Since U has no integrally matched vertex, for every crucial edge e inside U we have $x_e = 0$ and for every non-crucial edge e inside U by definition (4.3) we have $x_e \leq \frac{f_e}{p_e(1-\Pr[X_u])(1-\Pr[X_v])}$. By definition of f_e, it holds that $f_e \leq 1/\sqrt{\varepsilon R}$ and by Lemma 4.3.8 part 2, $\Pr[X_u], \Pr[X_v] \leq 1 - \varepsilon^2$. Replacing these into the bound above, we get $x_e \leq \frac{1}{p \times \varepsilon^2 \times \varepsilon^2 \sqrt{\varepsilon R}}$. Noting from Corollary 4.3.6 part 4 that $\tau_- < (\varepsilon p)^{50}$ and that $R = 2/\tau_-$, we get $R > 2/(\varepsilon p)^{50}$. Replacing this into the previous upper bound on x_e, we get that x_e is much smaller than say ε^3.

Now since $|U| \leq 1/\varepsilon$ there are at most $\binom{|U|}{2} < 1/\varepsilon^2$ edges e inside U that can have non-zero x_e. For each of these, as discussed above $x_e < \varepsilon^3$. Thus we have $x(U) < \varepsilon^3 \times 1/\varepsilon^2 < 1$ which cannot be larger than $\lfloor |U|/2 \rfloor$ if $|U| \geq 2$ (if $|U| \leq 1$, then there are no edges with both endpoints in U and thus clearly $x(U) = 0$). This contradicts the assumption that $x(U) > \lfloor |U|/2 \rfloor$, implying that there is no such subset. □

4.4.3 The Expected Size of x

In this section we prove the following.

Lemma 4.4.6. *It holds that* $\mathbb{E}[|x|] \geq (1 - 34\varepsilon)\textsc{opt}$.

We start by analyzing the size of x on the crucial edges. This is a simple consequence of Lemma 4.3.8 part 1 which guarantees $\mathbb{E}[Z] \geq q(C) - 30\varepsilon\textsc{opt}$ and Observation 4.3.4 which guarantees each crucial edge belongs to Q with probability at least $1 - \varepsilon$.

Claim 4.4.7. *It holds that* $\mathbb{E}\big[\sum_{e \in C} x_e\big] \geq q(C) - 31\varepsilon\textsc{opt}$.

Proof. Denoting $x(C) = \sum_{e \in C} x_e$, we have

$$\mathbb{E}[x(C)] = \mathbb{E}\Big[\sum_{e \in C} x_e\Big] = \sum_{e \in C} \mathbb{E}[x_e] = \sum_{e \in C} \Pr[e \in Q \text{ and } e \in Z].$$

Observe that Z and Q are picked independently as Lemma 4.3.8 is essentially unaware of Q. Therefore, for any crucial edge e we get

$$\Pr[e \in Q \text{ and } e \in Z] = \Pr[e \in Q] \times \Pr[e \in Z] \geq (1 - \varepsilon) \Pr[e \in Z],$$

where the latter inequality comes from Observation 4.3.7. Replacing this to the equality above gives

$$\begin{aligned} \mathbb{E}[x(C)] &\geq (1 - \varepsilon) \sum_{e \in C} \Pr[e \in Z] = (1 - \varepsilon)\mathbb{E}[|Z|] \\ &\overset{\text{Lemma 4.3.8 part 1}}{\geq} (1 - \varepsilon)(q(C) - 30\varepsilon\textsc{opt}) \geq q(C) - 31\varepsilon\textsc{opt}, \end{aligned}$$

completing the proof of the claim. □

To analyze the size of x on the non-crucial edges, we first define N' to be the subset of non-crucial edges $\{u, v\}$ such that $d_C(u, v) \geq \lambda$ and define $q(N') := \sum_{e \in N'} q_e$ and $x(N') := \sum_{e \in N'} x(N')$. Definition of N' is useful since recall from (4.3) that for any $\{u, v\} \in N$ with $d_C(u, v) < \lambda$ (i.e. $\{u, v\} \notin N'$) we set $x_e = 0$. Therefore only the edges in N that also belong to N' have non-zero x_e, implying $x(N) = x(N')$.

Claim 4.4.8. *It holds that $q(N') \geq q(N) - \varepsilon q(C)$.*

Proof. For any edge $e = \{u, v\}$ in $N \setminus N'$, we choose an arbitrary shortest path P between u and v in graph C and charge the edges of this path. Note that by definition of N', such path between u and v exists and has size less than λ. Now, take a crucial edge f. We denote by $\Phi(f)$ the set of edges in $N \setminus N'$ for which we charge a path containing f. Below, we argue that

$$|\Phi(f)| \leq 4(1/\tau_+)^{2\lambda} \qquad \forall f \in C. \tag{4.8}$$

Fix a crucial edge f and an edge $\{u, v\} \in \Phi(f)$. As discussed above, there should be a path of length less than λ between u and v in graph C that passes through f. This means that $d_C(u, f) < \lambda$ and $d_C(v, f) < \lambda$. Therefore, both u and v are at distance at most λ from f in graph C.

Observe that there are at most $2(\Delta_C)^\lambda$ vertices in the λ-neighborhood of f in graph C. Thus, there are at most $2(\Delta_C)^\lambda \times 2(\Delta_C)^\lambda = 4(\Delta_C)^{2\lambda}$ pairs of vertices that can potentially charge f, proving $|\Phi(f)| \leq 4(\Delta_C)^{2\lambda} \leq 4(1/\tau_+)^{2\lambda}$ where the latter inequality comes from Observation 4.3.4 that $\Delta_C \leq 1/\tau_+$. This concludes the proof of (4.8).

As discussed above, each edge $e \in N \setminus N'$ charges a path in C, thus belongs to

$\Phi(f)$ of at least one crucial edge f. Therefore, we get

$$|N \setminus N'| \leq \sum_{f \in C} \Phi(f). \tag{4.9}$$

Every edge e in $N \setminus N'$ is non-crucial, i.e. $q_e \leq \tau_-$. Thus:

$$\sum_{e \in N \setminus N'} q_e \leq \tau_- |N \setminus N'| \overset{(4.9)}{\leq} \tau_- \sum_{f \in C} \Phi(f) \overset{(4.8)}{\leq} 4\tau_- |C| (1/\tau_+)^{2\lambda} \leq 4\tau_- q(C)(1/\tau_+)^{2\lambda+1}, \tag{4.10}$$

where the last inequality comes from the fact that $q(C) \geq |C|\tau_+$ as for every edge $e \in C$, $q_e \geq \tau_+$.

From Corollary 4.3.6 we have $\tau_- = (\tau_+)^{10g}$ and we have $g \geq \lambda > 1$ by Observation 4.3.9. Thus:

$$4\tau_-(1/\tau_+)^{2\lambda+1} = 4(\tau_+)^{10g}(1/\tau_+)^{2\lambda+1} = 4(\tau_+)^{10g-(2\lambda-1)} < 4\tau_+ < \varepsilon.$$

Replacing it into inequality (4.10), we get

$$\sum_{e \in N \setminus N'} q_e \leq \varepsilon q(C).$$

This concludes the proof since

$$q(N') = \sum_{e \in N'} q_e = \sum_{e \in N \setminus (N \setminus N')} q_e \geq \sum_{e \in N} q_e - \sum_{e \in N \setminus N'} q_e \geq q(N) - \varepsilon q(C)$$

as it is desired. □

Claim 4.4.9. *It holds that* $\mathbb{E}[x(N')] \geq (1-\varepsilon)q(N')$.

Proof. By linearity of expectation, we have

$$\mathbb{E}[x(N')] = \mathbb{E}\Big[\sum_{e \in N'} x_e\Big] = \sum_{e \in N'} \mathbb{E}[x_e]. \tag{4.11}$$

We emphasize that the expectation here is taken over the randomization in Algorithm 2, the randomization in matching Z, and the randomization in realization of non-crucial edges. Specifically, we write $\mathbb{E}_{\mathsf{ALG}2, Z, \mathcal{N}}[x_e]$ to emphasize on this.

The randomization of Algorithm 2 determines the value of f_e which is used in defining x_e. Let us first condition on f_e and compute $\mathbb{E}_{Z,\mathcal{N}}[x_e \mid f_e]$. We have

$$\mathbb{E}_{Z,\mathcal{N}}[x_e \mid f_e] = \Pr[e \in \mathcal{E} \text{ and } u, v \notin V(Z) \mid f_e] \times \frac{f_e}{p_e(1 - \Pr[X_u])(1 - \Pr[X_v])}. \tag{4.12}$$

We claim that

$$\Pr[e \in \mathcal{E} \text{ and } u, v \notin V(Z) \mid f_e] = p_e(1 - \Pr[X_u])(1 - \Pr[X_v]). \tag{4.13}$$

To see this, first observe that the value of f_e is determined solely by the random realizations taken by Algorithm 2. In particular, the events $e \in \mathcal{E}$, and $u, v \notin V(Z)$ are completely independent of the outcome of Algorithm 2. This allows us to remove the condition on f_e from the left hand side of (4.13). Moreover, by Lemma 4.3.8 part 3, the matching Z is chosen independently from the realization of non-crucial edges, thus events $e \in \mathcal{E}$ and $u, v \notin V(Z)$ are independent. Finally, the assumption that $e \in N'$, by definition of N', implies that $d_C(u, v) \geq \lambda$. Therefore, by Lemma 4.3.8 part 4, events $v \in V(Z)$ and $u \in V(Z)$ (and for that matter their complements) are independent. Thus, indeed:

$$\begin{aligned}\Pr[e \in \mathcal{E} \text{ and } u, v \notin V(Z) \mid f_e] &= \Pr[e \in \mathcal{E}] \times \Pr[v \notin V(Z)] \times \Pr[u \notin V(Z)] \\ &= p_e(1 - \Pr[X_u])(1 - \Pr[X_v]).\end{aligned}$$

Replacing (4.13) into (4.12) we get

$$\mathbb{E}_{Z,\mathcal{N}}[x_e \mid f_e] = p_e(1 - \Pr[X_u])(1 - \Pr[X_v]) \times \frac{f_e}{p_e(1 - \Pr[X_u])(1 - \Pr[X_v])} = f_e.$$

Taking expectation over $\mathsf{ALG2}$ from both sides, we get

$$\mathbb{E}_{\mathsf{ALG2}}[\mathbb{E}_{Z,\mathcal{N}}[x_e \mid f_e]] = \mathbb{E}_{\mathsf{ALG2}}[f_e]. \tag{4.14}$$

The left hand side equals $\mathbb{E}_{\mathsf{ALG2},Z,\mathcal{N}}[x_e]$. For the right hand side, by Claim 4.4.2 we have $\mathbb{E}[f_e] \geq (1 - \varepsilon)q_e$. Replacing both the left hand side and right hand side of (4.14) by these bounds, we get

$$\mathbb{E}_{\mathsf{ALG2},Z,\mathcal{N}}[x_e] \geq (1 - \varepsilon)q_e. \tag{4.15}$$

Combining this with (4.11) we get

$$\mathbb{E}[x(N')] = \sum_{e \in N'} \mathbb{E}[x_e] \geq (1 - \varepsilon) \sum_{e \in N'} q_e = (1 - \varepsilon)q(N'),$$

completing the proof. ☐

We are now ready to prove Lemma 4.4.6.

Proof of Lemma 4.4.6. We have

$$\mathbb{E}\Big[\sum_{e} x_e\Big] = \mathbb{E}\Big[\sum_{e\in C} x_e\Big] + \mathbb{E}\Big[\sum_{e\in N} x_e\Big] \overset{\text{Claim 4.4.7}}{\geq} q(C) - 31\varepsilon\text{OPT} + \mathbb{E}\Big[\sum_{e\in N} x_e\Big].$$

Also note that for $e \in N$, $x_e \neq 0$ iff $e \in N'$ by construction of x. Thus,

$$\mathbb{E}\Big[\sum_{e\in N} x_e\Big] = \mathbb{E}\Big[\sum_{e\in N'} x_e\Big] = \mathbb{E}[x(N')] \overset{\text{Claim 4.4.9}}{\geq} (1-\varepsilon)q(N') \overset{\text{Claim 4.4.8}}{\geq} (1-\varepsilon)(q(N)-\varepsilon q(C)).$$

Combining the two equations above, we get

$$\begin{aligned}\mathbb{E}\Big[\sum_{e} x_e\Big] &\geq q(C) - 31\varepsilon\text{OPT} + (1-\varepsilon)(q(N) - \varepsilon q(C)) > q(C) + q(N) - 33\varepsilon\text{OPT} \\ &\overset{\text{Lemma 4.3.5 part (2)}}{\geq} (1-\varepsilon)\text{OPT} - 33\varepsilon\text{OPT} \geq (1-34\varepsilon)\text{OPT},\end{aligned}$$

concluding the proof. ☐

4.4.4 From the Expected Fractional Matching to an Actual Fractional Matching

We showed that x is an expected fractional matching satisfying $\mathbb{E}[x_v] \leq 1$ for every vertex v. However, as mentioned before, there is still a possibility that $x_v > 1$ depending on the coin tosses of the algorithms and the realization. This should never occur in a valid fractional matching. Thus, we define the following scaled fractional matching y based on x which decreases the fractional matching around vertices that deviate significantly from their expectation to 0.

$$\text{For any edge } e = \{u, v\}, \qquad y_e = \begin{cases} x_e/(1+\varepsilon) & \text{if } x_v, x_u \leq 1+\varepsilon, \\ 0 & \text{otherwise.} \end{cases} \tag{4.16}$$

Observation 4.4.10. *By definition above, y is a valid fractional matching, i.e. $y_v \leq 1$ for all $v \in V$. In addition, since $y_e \leq x_e$ for all edges e, Claim 4.4.5 implies that for all $U \subseteq V$ with $|U| \leq 1/\varepsilon$, $y(X) \leq \lfloor \frac{|U|}{2} \rfloor$. That is, y also satisfies all blossom inequalities of size up to $1/\varepsilon$.*

It remains to prove that while turning the expected fractional matching x into an actual fractional matching y, we don't significantly hurt the matching's size. We address this in the lemma below.

Lemma 4.4.11. $\mathbb{E}[|y|] \geq (1 - 55\varepsilon)\text{OPT}$.

The main ingredient in proving Lemma 4.4.11 is the following claim.

Claim 4.4.12. *For every vertex v, $\Pr[x_v > 1 + \varepsilon] \leq \varepsilon^6 p$.*

Let us first see how Claim 4.4.12 suffices to prove Lemma 4.4.11 and then prove it.

Proof of Lemma 4.4.11. By definition of y_e in (4.16), we have

$$\begin{aligned}\sum_e y_e &= \sum_{e=\{u,v\}} \mathbb{1}(x_u \le 1+\varepsilon \text{ and } x_v \le 1+\varepsilon)\frac{x_e}{1+\varepsilon} \\ &\ge \sum_{e=\{u,v\}} (1 - \mathbb{1}(x_u > 1+\varepsilon) - \mathbb{1}(x_v > 1+\varepsilon))\frac{x_e}{1+\varepsilon} && \text{Union bound.} \\ &= \sum_e \frac{x_e}{1+\varepsilon} - 2\sum_{v:x_v>1+\varepsilon}\sum_{e\ni v}\frac{x_e}{1+\varepsilon} = \sum_e \frac{x_e}{1+\varepsilon} - 2\sum_{v:x_v>1+\varepsilon}\frac{x_v}{1+\varepsilon}.\end{aligned}$$

Taking expectation from both sides, we get

$$\begin{aligned}\mathbb{E}\Big[\sum_e y_e\Big] &\ge \mathbb{E}\Big[\sum_e \frac{x_e}{1+\varepsilon} - 2\sum_{v:x_v>1+\varepsilon}\frac{x_v}{1+\varepsilon}\Big] = \frac{1}{1+\varepsilon}\left(\mathbb{E}\Big[\sum_e x_e\Big] - 2\mathbb{E}\Big[\sum_{v:x_v>1+\varepsilon} x_v\Big]\right) \\ &\ge \frac{1}{1+\varepsilon}\left((1-34\varepsilon)\text{OPT} - 2\mathbb{E}\Big[\sum_{v:x_v>1+\varepsilon} x_v\Big]\right) && \text{By Lemma 4.4.6.} \\ &\ge (1-35\varepsilon)\text{OPT} - 2\sum_v \Pr[x_v > 1+\varepsilon]\mathbb{E}[x_v \mid x_v > 1+\varepsilon] \\ &\ge (1-35\varepsilon)\text{OPT} - 2\sum_v \varepsilon^6 p\mathbb{E}[x_v \mid x_v > 1+\varepsilon] && \text{By Claim 4.4.12.}\end{aligned} \tag{4.17}$$

We will soon prove that for every vertex v, it <u>deterministically</u> holds that $x_v \le \frac{1}{p\varepsilon^4}$. Replacing this into the last inequality above, gives the desired bound that

$$\begin{aligned}\mathbb{E}\Big[\sum_e y_e\Big] &\ge (1-35\varepsilon)\text{OPT} - 2\sum_v \varepsilon^6 p\frac{1}{p\varepsilon^4} \\ &\ge (1-35\varepsilon)\text{OPT} - 2\varepsilon^2 n \overset{\text{Assumption 4.3.1}}{\ge} (1-35\varepsilon)\text{OPT} - 20\varepsilon\text{OPT} \\ &= (1-55\varepsilon)\text{OPT}.\end{aligned}$$

Now let's see why $x_v \leq \frac{1}{p\varepsilon^4}$. Observe from the definition of x that if $v \in V(Z)$ then $x_v \leq 1$ and otherwise

$$x_v = \sum_{e=\{v,u\}} x_e \leq \sum_{e=\{v,u\}} \frac{f_e}{p(1-\Pr[X_u])(1-\Pr[X_v])} \leq \frac{1}{p\varepsilon^4} \sum_{e=\{v,u\}} f_e.$$

The last inequality above comes from the fact that for every vertex w, $\Pr[X_w] \leq 1-\varepsilon^2$ due to Lemma 4.3.8 part 2, which means $1 - \Pr[X_w] \geq \varepsilon^2$.

Now recall from Claim 4.4.2 part 3 that $\sum_{e \ni v} f_e \leq 1$. Thus we get our desired upper bound that $x_v \leq \frac{1}{p\varepsilon^4}$. As described above, this completes the proof that $\mathbb{E}[\sum_e y_e] \geq (1-55\varepsilon)\text{OPT}$. □

We now turn to prove Claim 4.4.12 that $\Pr[x_v > 1+\varepsilon] \leq \varepsilon^6 p$ for all v.

Proof of Claim 4.4.12. If an edge incident to v belongs to matching Z, i.e. if $X_v = 1$ (as defined in Lemma 4.3.8), then one can confirm easily from the definition of x in (4.1) and (4.3) that either $x_v = 1$ or $x_v = 0$, implying that $\Pr[x_v > 1+\varepsilon \mid X_v = 1] = 0$. As such, for the rest of the proof, we simply condition on the event that $X_v = 0$.

Similar to the proof of Claim 4.4.4 let $u_1, u_2, \ldots, u_r$ be the neighbors of v such that for each $i \in [r]$, (1) edge $e_i = \{v, u_i\}$ is non-crucial, and (2) $d_C(v, u_i) \geq \lambda$. Recall from (4.4) that given event $X_v = 0$, it holds that

$$x_v = x_{e_1} + x_{e_2} + \ldots + x_{e_r}.$$

Let $f'_v := \sum_{i=1}^{r} f_{e_i}$ and note that $f'_v \leq f_v$ since f_v is sum of f_e of all non-crucial edges e connected to v. Claim 4.4.2 part 4 proves that $\Pr[f_v \geq n_v + 0.1\varepsilon] \leq (\varepsilon p)^{10}$.

Therefore, it also holds that $\Pr[f'_v \geq n_v + 0.1\varepsilon] \leq (\varepsilon p)^{10}$ since $f'_v \leq f_v$. For the rest of the proof, we regard f_{e_i}'s as (adversarially) fixed with the only assumption that $f'_v < n_v + 0.1\varepsilon$ which happens with probability at least $1 - (\varepsilon p)^{10}$. We denote this event, as well as the event that $X_v = 0$, by A and prove

$$\Pr[x_v > 1 + \varepsilon \mid A] \leq 0.5\varepsilon^6 p, \tag{4.18}$$

which clearly is sufficient for proving the claim.

We do this by proving a concentration bound using the second moment method. Consider the variance of x_v conditioned on A:

$$\mathrm{Var}[x_v \mid A] = \sum_{i=1}^{r} \sum_{j=1}^{r} \mathrm{Cov}(x_{e_i}, x_{e_j} \mid A).$$

Now that f_e's are fixed, x_v is only a random variable of (1) the randomization used in Lemma 4.3.8 for obtaining matching Z, and (2) the realization of non-crucial edges.

In what follows we identify a condition under which covariance of x_{e_i} and x_{e_j} becomes 0. We will use this later to upper bound $\mathrm{Var}[x_v \mid A]$.

Observation 4.4.13. *Let $i, j \in [r]$ be such that $d_C(u_i, u_j) \geq \lambda$. Then* $\mathrm{Cov}(x_{e_i}, x_{e_j} \mid A) = 0$.

Proof. We already had $d_C(v, u_i) \geq \lambda$ and $d_C(v, u_j) \geq \lambda$ by definition of u_i, u_j. Combined with assumption $d_C(u_i, u_j) \geq \lambda$ and using Lemma 4.3.8 part 4, we get that X_v, X_{u_i}, X_{u_j} are independent. Realization of e_i and e_j are also independent even given A. This is because these are non-crucial edges and thus are realized independently from Z (according to Lemma 4.3.8 part 3) or the values of f which are derived from Algorithm 2.

By definition (4.3), the value of x_{e_i} conditioned on A is fully determined once we know X_{u_i} and whether e_i is realized. Similarly, the value of x_{e_j} conditioned on A is fully determined once we know X_{u_j} and whether e_j is realized. These, as discussed above, are independent. Hence x_{e_i} and x_{e_j}, conditioned on A, are independent and thus their covariance is 0. $\square$

Now consider two vertices u_i and u_j (possibly $u_i = u_j$) where $d_C(u_i, u_j) < \lambda$. Here, the covariance may not be 0. But we still can upper bound it as follows:

$$\begin{aligned}\mathrm{Cov}(x_{e_i} x_{e_j} \mid A) &= \mathbb{E}[x_{e_i} x_{e_j} \mid A] - \mathbb{E}[x_{e_i} \mid A]\mathbb{E}[x_{e_j} \mid A] \leq \mathbb{E}[x_{e_i} x_{e_j} \mid A] \\ &\leq \frac{f_{e_i}}{p(1-\Pr[X_v])(1-\Pr[X_{u_i}])} \times \frac{f_{e_j}}{p(1-\Pr[X_v])(1-\Pr[X_{u_j}])} \\ &\leq \frac{f_{e_i} f_{e_j}}{p^2 \varepsilon^8}, \end{aligned} \tag{4.19}$$

where the last inequality follows from Lemma 4.3.8 part 2 that states for all vertices w, $\Pr[X_w] < 1 - \varepsilon^2$ and thus $1 - \Pr[X_w] \geq \varepsilon^2$.

Now, for each $i \in [r]$, let $D_i := \{j : d_C(u_i, u_j) < \lambda\}$. Since C is a graph of max degree Δ_C, the $\lambda - 1$ neighborhood of each vertex u_i in C includes $\leq (\Delta_C)^{\lambda-1}$ vertices. Thus:

$$|D_i| \leq (\Delta_C)^{\lambda-1} \qquad \text{for every } i \in [r]. \tag{4.20}$$

Having these, we obtain that

$$\begin{aligned}
\mathrm{Var}[x_v \mid A] &= \sum_{i=1}^{r}\sum_{i=1}^{r} \mathrm{Cov}(x_{e_i}, x_{e_j} \mid A) \overset{\text{Obs } 4.4.13}{=} \sum_{i=1}^{r}\sum_{j\in D_i} \mathrm{Cov}(x_{e_i}, x_{e_j} \mid A) \\
&\overset{(4.19)}{\leq} \sum_{i=1}^{r}\sum_{j\in D_i} \frac{f_{e_i} f_{e_j}}{p^2\varepsilon^8} = \frac{1}{p^2\varepsilon^8}\sum_{i=1}^{r}\left(f_{e_i}\sum_{j\in D_i} f_{e_j}\right) \\
&\overset{f_{e_j}\leq\frac{1}{\sqrt{\varepsilon R}} \text{ by } (4.2)}{\leq} \frac{1}{p^2\varepsilon^8}\sum_{i=1}^{r}\left(f_{e_i}|D_i|\frac{1}{\sqrt{\varepsilon R}}\right) \overset{(4.20)}{\leq} \frac{(\Delta_C)^{\lambda-1}}{p^2\varepsilon^8\sqrt{\varepsilon R}}\sum_{i=1}^{r} f_{e_i} \\
&\overset{\text{Claim } 4.4.2 \text{ part } 3}{\leq} \frac{(\Delta_C)^{\lambda-1}}{p^2\varepsilon^8\sqrt{\varepsilon R}} \overset{\text{Obs } 4.3.4}{\leq} \frac{(1/\tau_+)^{\lambda-1}}{p^2\varepsilon^{8.5}\sqrt{R}}.
\end{aligned}$$

Replacing R with $\frac{1}{2\tau_-}$ and noting that $\tau_- = \tau_+^{10g}$ by Corollary 4.3.6, we get that

$$\begin{aligned}
\mathrm{Var}[x_v \mid A] &\leq \frac{(1/\tau_+)^{\lambda-1}}{p^2\varepsilon^{8.5}\sqrt{\frac{1}{2\tau_+^{10g}}}} \\
&= \frac{\sqrt{2}\cdot\tau_+^{-\lambda+1}}{p^2\varepsilon^{8.5}\tau_+^{-5g}} \\
&= \frac{\sqrt{2}}{p^2\varepsilon^{8.5}}\cdot\tau_+^{5g-\lambda+1} \\
&< \frac{\sqrt{2}\tau_+}{p^2\varepsilon^{8.5}} && \text{By Observation 4.3.9 } g \geq \lambda > 1 \text{ and } \tau_+ < 1. \\
&< \frac{\sqrt{2}(\varepsilon p)^{50}}{p^2\varepsilon^{8.5}} && \text{Corrolary 4.3.6 part 4.} \\
&= \sqrt{2}\varepsilon^{41.5}p^{48} < 0.1\varepsilon^8 p && \text{For } \varepsilon \text{ sufficiently small.}
\end{aligned}$$

With this upper bound on the variance, we can use Chebyshev's inequality to get

$$\Pr\left[|x_v - \mathbb{E}[x_v \mid A]| > 0.5\varepsilon \,\middle|\, A\right] \leq \frac{\mathrm{Var}[x_v \mid A]}{(0.5\varepsilon)^2} \leq \frac{0.1\varepsilon^8 p}{0.25\varepsilon^2} < 0.5\varepsilon^6 p. \tag{4.21}$$

Next, recall from (4.7) in the proof of Claim 4.4.4 that $\mathbb{E}[x_v \mid v \notin V(Z)] \leq \frac{\sum_{i=1}^{r}\mathbb{E}[f_{e_i}]}{1-\Pr[X_v]} =$

$\frac{f'_v}{1-\Pr[X_v]}$. Event A in addition to $v \notin V(Z)$ also fixes the value of f'_v. But recall that event A (as we defined it) guarantees $f'_v \leq n_v + 0.5\varepsilon$. Therefore, we get

$$\mathbb{E}[x_v \mid A] \leq \frac{n_v + 0.5\varepsilon}{1 - \Pr[X_v]} \overset{\Pr[X_v]<c_v}{\leq} \frac{n_v + 0.5\varepsilon}{1 - c_v} \overset{n_v \leq 1-c_v}{\leq} \frac{1 - c_v + 0.5\varepsilon}{1 - c_v} \leq 1 + 0.5\varepsilon. \quad (4.22)$$

Combining (4.21) and (4.22) we get the claimed inequality of (4.18) that

$$\Pr[x_v > 1 + \varepsilon \mid A] \leq \Pr[|x_v - \mathbb{E}[x_v \mid A]| > 0.5\varepsilon \mid A] \leq 0.5\varepsilon^6 p,$$

which as described before suffices to prove $\Pr[x_v > 1 + \varepsilon] \leq \varepsilon^6 p$. □

4.5 Proof of the Vertex-Independent Matching Lemma

In this section we turn to prove Lemma 4.3.8 restated below.

Lemma 4.3.8 (restated)**.** *There is a randomized algorithm that constructs an integral matching Z of $\mathcal{C}$ (the realized subgraph of C) such that defining X_v as the indicator random variable for $v \in V(Z)$, we get:*

1. $\mathbb{E}[|Z|] \geq q(C) - 30\varepsilon\text{OPT}$.
2. *For every vertex v, $\Pr[X_v] \leq \max\{c_v - \varepsilon^2, 0\}$, where recall that c_v is the probability that vertex v is matched via a crucial edge in $\mathsf{MM}(\mathcal{G})$.*
3. *The matching Z is independent of the realization of non-crucial edges in G.*
4. *Let $\lambda := \varepsilon^{-20} \log \Delta_C$. For every k and every $\{v_1, v_2, \ldots, v_k\} \subseteq V$ such that $d_C(v_i, v_j) \geq \lambda$ for all $v_i \neq v_j$, random variables $X_{v_1}, \ldots, X_{v_k}$ are independent.*

We emphasize that $\mathbb{E}[|Z|]$ and X_v are both defined with respect to the randomizations in both the realization of C, and the randomization of the algorithm in constructing Z.

4.5.1 Overview of the Algorithm

In this section, we give an overview of our algorithm for proving Lemma 4.3.8. We emphasize that the overview given here is deliberately informal to describe the main intuitions, with the hope that it makes the algorithm and its analysis more accessible.

Satisfying property 3 required by Lemma 4.3.8 turns out to be easy. Recall that we are constructing matching Z on the realized *crucial* edges, thus we can simply ignore realization of non-crucial edges and automatically satisfy property 3. Among the other 3, let us first focus on property 4. How can we argue that the output matching satisfies the required independence property? We show that the LOCAL model of computation can be naturally used for this purpose. We start with the formal definition of the model and then describe how it can be used in this case.

The LOCAL model [34]. In the LOCAL model, the input is a graph and there is a processor on each node of this graph. Computation proceeds in synchronous rounds and in each round, each processor can send a message (of any size) to each of its neighbors. The goal is to output a property of this communication graph, e.g. a matching of it. At the end, each node should know its part of the output, e.g. which one of its edges, if any, is part of the matching.

Why the LOCAL model. A particularly useful property of any r-round LOCAL algorithm is that the output of each node essentially depends only on its r-hop neighborhood. That is, having the r-hop neighborhood of each node v (including the random tapes of the nodes in the neighborhood), we can uniquely determine the output of v. Therefore if the shortest path between two nodes is at least $2r + 1$, their outputs are essentially independent of each other after r rounds.

This is how we prove property 4 of Lemma 4.3.8 is satisfied: We give a LOCAL algorithm operating on graph C where each vertex is initially only aware of the realization of its incident edges. We show that the algorithm within $< \lambda/2$ rounds, finds a matching satisfying the other 3 properties. Then property 4 will be automatically satisfied. That is, for every subset I of the vertices with pairwise distance at least λ, their outputs will be independent.

Overview of the algorithm. The challenge is to ensure that the algorithm has low round-complexity while also satisfying properties 1 and 2. That is, the reported matching Z should be large in expectation (property 1), and that no vertex v should be matched with a larger probability than that specified in property 2. If one ignores the 2nd property, then simply finding a $(1 - \varepsilon)$-approximate maximum matching in graph $\mathcal{C}$ will satisfy the first property. And we remark that $O(\log \Delta_C)$-round algorithms (with no dependence on n) do exist for this purpose. However, bounding at the same time, the probability that each vertex is matched complicates things.

Our general idea for the algorithm is as follows: We define a recursive algorithm $\mathsf{FindMatching}_r(\mathcal{C})$ (Algorithm 2) which uses $\mathsf{FindMatching}_{r-1}(\mathcal{C})$ as a subroutine. The base algorithm $\mathsf{FindMatching}_0(\mathcal{C})$ returns an empty matching. Let us use Z_r to denote the matching returned by $\mathsf{FindMatching}_r(\mathcal{C})$. It will hold that

$$0 = \mathbb{E}[|Z_0|] \leq \mathbb{E}[|Z_1|] \leq \mathbb{E}[|Z_2|] \leq \dots$$

until eventually for large enough $t = O_\varepsilon(1)$, $\mathbb{E}[|Z_t|]$ is desirably large, satisfying property 1. At the same time, we will ensure that for any vertex v, the probability that it gets matched in Z_r never exceeds the upper bound of property 2 for any r.

Suppose that for a vertex v, we hit this upper bound on the probability that it is matched for algorithm $\mathsf{FindMatching}_r(\mathcal{C})$. At this point, we will mark v as *saturated* and ensure that we never increase the probability of it being matched. But to keep increasing the matching's size, it may be necessary to say remove a matching edge $\{v_1, v_2\}$ between two saturated vertices v_1 and v_2, so that we can add two edges $\{v_1, v_3\}$ and $\{v_2, v_4\}$ to the matching where v_3 and v_4 are unsaturated. Such structures are similar to augmenting paths. However, since the graph is stochastic, these edges $\{v_1, v_2\}, \{v_1, v_3\}, \{v_2, v_4\}$ may not necessarily be part of one realization. We call these natural generalizations of augmenting paths, "augmenting hyperwalks" (see Section 4.5.2) and show that they can be used to increase the matching size while not increasing probability of saturated vertices getting matched.

In Section 4.5.2 we present a centralized view of the algorithm. In Section 4.5.3 we analyze the expected size of the matching returned by this algorithm and argue that it satisfies property 1 of Lemma 4.3.8. In Section 4.5.4 we prove the upper bound on the probability of each vertex getting matched, thereby proving property 2 of Lemma 4.3.8. Finally, in Section 4.5.5 we show that the algorithm has an efficient LOCAL implementation, satisfying property 4 of Lemma 4.3.8.

4.5.2 The Formal Algorithm

We say $P = ((\mathcal{C}_0, M_0), \ldots, (\mathcal{C}_\alpha, M_\alpha))$ is a *profile* if each $\mathcal{C}_i$ is a subgraph of C and each M_i is a matching of $\mathcal{C}_i$. Furthermore, we call a sequence $W = ((e_1, s_1), \ldots, (e_k, s_k))$ a *hyperwalk* of size k if the following conditions hold:

1. Each s_i is an integer in $\{0, \ldots, \alpha\}$.
2. Each e_i is an edge in graph C and sequence $(e_1, e_2, \ldots, e_k)$ is a walk in graph C.

We say $P\Delta W := ((\mathcal{C}_0, M'_0), \ldots, (\mathcal{C}_\alpha, M'_\alpha))$ is the result of *applying* W on P if:

$$M'_i = M_i \cup \{e_j \mid j \text{ is odd, and } s_j = i\} \setminus \{e_j \mid j \text{ is even, and } s_j = i\}, \qquad \forall i \in \{0, \ldots, \alpha\}.$$

Definition 4.5.1 (Augmenting hyperwalks)**.** *For every vertex v, let $d_P(v) := \big|\{i \mid v \in V(M_i)\}\big|$. We say W is an* augmenting-hyperwalk *of P if it satisfies the three following conditions.*

1. *$P\Delta W$ is a profile, i.e. each M'_i in $P\Delta W$ is a matching of graph $\mathcal{C}_i$.*
2. *For all vertices v in walk $(e_1, \ldots, e_k)$ except its first and last vertex, $d_P(v) = d_{P\Delta W}(v)$.*
3. *For the first and last vertices v in walk $(e_1, \ldots, e_k)$, $d_P(v) + 1 = d_{P\Delta W}(v)$.*

Having defined augmenting-hyperwalks, we can now formally state the algorithm—see Algorithm 2. The algorithm is recursive. Given a realization $\mathcal{C}$ of C, algorithm $\mathsf{FindMatching}_r(\mathcal{C})$ uses algorithm $\mathsf{FindMatching}_{r-1}(\mathcal{C})$ as a subroutine and then returns

Algorithm 2. $\mathsf{FindMatching}_r(\mathcal{C})$

(1) If $r = 0$, return $\emptyset$.

(2) Draw $\alpha := 1/\varepsilon^7 - 1$ realizations $\mathcal{C}_1, \ldots, \mathcal{C}_\alpha$ of C where each realization $\mathcal{C}_i$ includes each edge e of C independently with probability p_e. Also let $\mathcal{C}_0 := \mathcal{C}$.

(3) Consider profile $P = ((\mathcal{C}_0, M_0), \ldots, (\mathcal{C}_\alpha, M_\alpha))$ where $M_i = \mathsf{FindMatching}_{r-1}(\mathcal{C}_i)$.

(4) For every vertex v, define $\gamma_{v,r-1} := \Pr[v \text{ is matched in } \mathsf{FindMatching}_{r-1}(\mathcal{C}')]$ where the probability is taken over a random realization $\mathcal{C}'$ of C and the randomization of the algorithm.

(5) If $\gamma_{v,r-1} < c_v - 2\varepsilon^2$ call vertex v *unsaturated* and *saturated* otherwise.

(6) Construct a graph $H = (V_H, E_H)$ as described next. For every possible augmenting-hypewalk of size smaller than $2/\varepsilon$ from P, we put a vertex in V_H iff the first and last vertices in the walk are unsaturated. Moreover, we put an edge in E_H between two nodes $u, v \in V_H$ iff their corresponding walks share at least a vertex.

(7) $I \leftarrow \mathsf{ApproximateMIS}(H, \varepsilon)$. // This is an algorithm that returns an independent set of expected size at least $1-\varepsilon$ fraction of some maximal independent set (MIS) of H.

(8) $P' \leftarrow P$.

(9) Iterate over all augmenting-hyperwalks $W \in I$ and apply them, i.e. set $P' \leftarrow P' \Delta W$.

(10) Let $P' = ((\mathcal{C}_0, M'_0), \ldots, (\mathcal{C}_\alpha, M'_\alpha))$ be the final profile. Return matching M'_0.

a matching of $\mathcal{C}$. The base algorithm $\mathsf{FindMatching}_0(\mathcal{C})$ returns an empty matching. We will show that for $t = 1/\varepsilon^9$, algorithm $\mathsf{FindMatching}_t(\mathcal{C})$ satisfies the properties of Lemma 4.3.8.

We note a useful observation that essentially implies the entries of profile P', which can be thought of as random variables of realization $\mathcal{C}$ and randomizations of the algorithm, are all drawn from the same distribution. The proof is essentially based on the fact that matchings $M_0, \ldots, M_\alpha$ are all drawn from the same distribution and treated symmetrically in algorithm, thus the resulting matchings $M'_0, \dots, M'_\alpha$ all have the same distribution. See Section 4.9 for a more formal proof.

Observation 4.5.2. *Matchings $M'_0, \dots, M'_\alpha$ in profile P' of algorithm* $\mathsf{FindMatching}_r(\mathcal{C})$ *for any r have the same distribution. That is, for any $i, j \in \{0, \dots, \alpha\}$ and any matching M' of G, $\Pr[M'_i = M'] = \Pr[M'_j = M']$.*

The algorithm operates only on the crucial edges and is thus clearly independent of the non-crucial edges and their realizations. Therefore, property 3 of Lemma 4.3.8 is automatically satisfied. In what follows, we prove the other 3 properties in Sections 4.5.3, 4.5.4, and 4.5.5.

4.5.3 Lemma 4.3.8 Property 1: The Matching's Size

In this section, we prove that algorithm $\mathsf{FindMatching}_t(\mathcal{C})$ satisfies the first property of Lemma 4.3.8. That is the matching Z returned by this algorithm satisfies $\mathbb{E}[|Z|] \geq q(C) - 30\varepsilon\textsc{opt}$.

Let us denote by Z_r the matching returned by $\mathsf{FindMatching}_r(\mathcal{C})$. Note that Z_r is a random variable which is a function of both the randomization in realization $\mathcal{C}$ of C, and the internal randomizations used in algorithm $\mathsf{FindMatching}_r(\mathcal{C})$. (Observe that $Z = Z_t$.) Similarly, we define P_r, H_r, I_r, and P'_r as the random variables referring to the values of P, H, I, and P' in algorithm $\mathsf{FindMatching}_r(\mathcal{C})$.

Property 1 of Lemma 4.3.8 is a corollary of Lemma 4.5.3 which states that for any r, if $\mathbb{E}[|Z_r|] \leq q(C) - 30\varepsilon\textsc{opt}$, then $\mathbb{E}[|Z_r|] - \mathbb{E}[|Z_{r-1}|] \geq \varepsilon^9\textsc{opt}$. Observe that it is sufficient for us as it implies that for any r we have

$$\mathbb{E}\big[|Z_t|\big] \geq \min\{q(C) - 30\varepsilon\text{OPT}, r\varepsilon^9\text{OPT}\}.$$

This gives us the desired result that $\mathbb{E}\big[|Z_t|\big] \geq q(C) - 30\varepsilon\text{OPT}$ for $t = 1/\varepsilon^9$, since $q(C) \leq \text{OPT}$. Below we state Lemma 4.5.3 and prove it.

Lemma 4.5.3. *For any r, if $\mathbb{E}[|Z_r|] \leq q(C) - 30\varepsilon\text{OPT}$, then $\mathbb{E}[|Z_r|] - \mathbb{E}[|Z_{r-1}|] \geq \varepsilon^9\text{OPT}$.*

Proof outline. This lemma is a direct result of Lemma 4.5.4 and Lemma 4.5.6. The first one states that for any r, we have $\mathbb{E}[|Z_r|] \geq \mathbb{E}[|Z_{r-1}|] + \frac{\mathbb{E}[|I_r|]}{\alpha+1}$ and the second one is that if $\mathbb{E}[|Z_{r-1}|] \leq q(C) - 30\varepsilon\text{OPT}$, then $\mathbb{E}[|I_r|] \geq 2\varepsilon^2\text{OPT}$. Combining these two lemmas gives us $\mathbb{E}[|Z_r|] - \mathbb{E}[|Z_{r-1}|] \geq \varepsilon^9\text{OPT}$ and completes the proof as $\alpha = 1/\varepsilon^7 - 1$. ☐

Lemma 4.5.4. *For any r, it holds that $\mathbb{E}[|Z_r|] = \mathbb{E}[|Z_{r-1}|] + \frac{\mathbb{E}[|I_r|]}{\alpha+1}$.*

Proof. We start by proving that

$$\sum_{v \in V} d_{P_r}(v) + 2|I_r| = \sum_{v \in V} d_{P'_r}(v). \tag{4.23}$$

Note that, P'_r is defined to be the result of iteratively applying all the augmenting hyperwalks of I_r on P_r. Let $P_r^{(i)}$ be the result of iteratively applying the first i augmenting hyperwalks of I_r on P_r and let W_i be the hyperwalk that is to be applied in iteration i. We use proof by induction and show that for any i we have

$$\sum_{v \in V} d_{P_r}(v) + 2i = \sum_{v \in V} d_{P_r^{(i)}}(v).$$

Note that since hyperwalks in I_r are vertex disjoint, for any two hyperwalks $W_1, W_2 \in I_r$ it holds that W_2 is an augmenting hyperwalk of $P_r \Delta W_1$ as well. This means that W_i is indeed an augmenting hyperwalk of $P_r^{(i)}$. Moreover, recall that by definition of augmenting hyperwalks, after applying any augmenting hyperwalk on a profile P there are only two vertices whose $d_P(v)$ increases by one and for the rest of the vertices it is unchanged. This gives us

$$\sum_{v \in V} d_{P_r^{(i)}}(v) + 2 = \sum_{v \in V} d_{P_r^{(i+1)}}(v),$$

with completes the proof of

$$\sum_{v \in V} d_{P_r}(v) + 2|I_r| = \sum_{v \in V} d_{P_r'}(v)$$

since $P_r' = P_r^{|I_r|}$. Recall the definition $d_P(v) := \big|\{i \mid v \in V(M_i)\}\big|$ for any profile P. Based on this definition, we can rewrite Equation 4.23 as

$$\sum_{i=0}^{\alpha} |M_i| + |I_r| = \sum_{i=0}^{\alpha} |M_i'|. \tag{4.24}$$

Observe that matchings $M_0, \ldots, M_\alpha$ are coming from the same distribution and we have $\mathbb{E}\big[|M_i|\big] = \mathbb{E}\big[|Z_{r-1}|\big]$ for any $0 \leq i \leq \alpha$. The reason is that they are the results of running the same matching algorithm on random realizations of C. Moreover, by Observation 4.5.2, matchings $M_0', \ldots, M_\alpha'$ are similarly coming from the same distribution which means for any $0 \leq i \leq \alpha$ we have $Z_r = \mathbb{E}\big[|M_0'|\big] = \mathbb{E}\big[|M_i|\big]$. Combining this with Equation 4.24 we get

$$\mathbb{E}\big[(\alpha+1)|Z_{r-1}|+|I|\big]=\mathbb{E}\left[\sum_{i=0}^{\alpha}|M_i|+|I|\right]=\mathbb{E}\left[\sum_{i=0}^{\alpha}|M'_i|\right]=\mathbb{E}\big[(\alpha+1)|Z_r|\big].$$

Dividing through by $\alpha+1$ and rearranging the terms gives $\mathbb{E}\big[|Z_{r-1}|\big]+\frac{\mathbb{E}[|I|]}{\alpha+1}=\mathbb{E}\big[|Z_r|\big]$.

□

Before proceeding to Lemma 4.5.6 and its proof we need the following definition.

Definition 4.5.5 (Edge disjoint hyperwalks)**.** *Two hyperwalks* $W=((e_1,s_1),\ldots,(e_k,s_k))$ *and* $W'=((e'_1,s'_1),\ldots,(e'_{k'},s'_{k'}))$ *are edge disjoint if there does not exist indices* $i<k$ *and* $j<k'$*, where* $e_i=e'_j$ *and* $s_i=s'_j$*.*

Lemma 4.5.6. *If* $\mathbb{E}[|Z_{r-1}|]\le q(C)-30\varepsilon\text{OPT}$*, then* $\mathbb{E}[|I_r|]\ge 2\varepsilon^2\text{OPT}$*.*

Proof. To give the desired lower-bound for $\mathbb{E}[|I_r|]$ we first claim that if $\mathbb{E}[|Z_{r-1}|]\le q(C)-30\varepsilon\text{OPT}$, then there exists a set O of edge-disjoint augmenting-hyperwalks of P_r with length at most $2/\varepsilon$ and unsaturated end-points where $\mathbb{E}[|O|]\ge 8(\alpha+1)\varepsilon\text{OPT}$. We later state this claim more formally in Lemma 4.5.7 and provide a proof for it. We are interested in set O for its two following properties. First, any hyperwalk in O represents a node in graph H_r. Second, since the hyperwalks in O are edge disjoint, any hyperwalk with length smaller than $2/\varepsilon$ from P_r can share vertices with at most $(\alpha+1)(2/\varepsilon)$ hyperwalks in this set. We note that $(\alpha+1)$ is the maximum number of edge disjoint hyperwalks that can pass through a single vertex. Combining these two properties gives that the expected size of any maximal independent set of H_r is at least $\mathbb{E}\big[|O|\big]/(2(\alpha+1)/\varepsilon)=4\varepsilon^2\text{OPT}$ since there is an edge between two vertices in

H_r iff their corresponding hyperwalks share at least a vertex. As stated in Line 7 of $\mathsf{FindMatching}_r(\mathcal{C})$, set I_r is an independent set of H_r with size at least $(1-\varepsilon)$ fraction of a maximal independent set of H_r. Therefore, we have

$$\mathbb{E}\big[|I_r|\big] \geq 4(1-\varepsilon)\varepsilon^2\textsc{opt}.$$

Assuming that $\varepsilon \leq 1/2$ we complete the proof of this claim and obtain $\mathbb{E}\big[|I_r|\big] \geq 2\varepsilon^2\textsc{opt}$. □

In the rest of this section we focus on proving the following lemma which is previously used to complete the proof of Lemma 4.5.6. Since the proof is detailed and consists of independent arguments, it includes two claims that are needed to complete the proof.

Lemma 4.5.7. *For any $r \in [t]$, if $\mathbb{E}\big[|Z_{r-1}|\big] \leq q(C) - 30\varepsilon\textsc{opt}$, then there exists a set O of edge-disjoint augmenting hyperwalks of profile $P_r = ((\mathcal{C}_0, M_0), \ldots, (\mathcal{C}_\alpha, M_\alpha))$ with length at most $2/\varepsilon$ and unsaturated endpoints where $\mathbb{E}[|O|] \geq 8(\alpha+1)\varepsilon\textsc{opt}$.*

We will first construct set O and then give a lower-bound for its expected size. Draw $\alpha+1$ realizations $\mathcal{N}_0, \ldots, \mathcal{N}_\alpha$ of the non-crucial graph N. For any $0 \leq i \leq \alpha$, let $M_i^g := \mathsf{MM}(\mathcal{N}_i \cup \mathcal{C}_i)$ where M returns a unique maximum matching that was also used in Algorithm 2. Call an edge of graph $\mathcal{C}_i$ *green* iff it is in matching M_i^g but not in matching M_i. Alternatively, we call an edge *red* iff it is in M_i but not in M_i^g. To construct set O we give an algorithm to iteratively find hyperwalks that alternate between green and red edges. Since we need our hyperwalks to be edge-disjoint, after using an edge of a subgraph we mark it as used and ignore it for the rest of the algorithm.

At each iteration of the algorithm, we construct a hyperwalk W as follows until there is no such a hyperwalk left. Pick an unsaturated vertex v and a subgraph $\mathcal{C}_i$ such that v has an unused green edge in $\mathcal{C}_i$ but not a red one. Denote this green edge by $e = (v, v')$ and choose (e, i) to be the first element of our hyperwalk. If vertex v' has a red edge e' in subgraph $\mathcal{C}_i$ we add (e', i) to our hyperwalk, otherwise we look for a subgraph $\mathcal{C}_j$ in which v' has an unused red edge e' but not a green one and choose (e', j) as the second element of the hyperwalk. We continue this process by alternating the colors until it is not possible to continue. If W has length more than $2/\varepsilon$ we add it to a set T_3, otherwise we add it to one of the three sets O, T_1 and T_2 as follows. Let u be the vertex in which our hyperwalk ends. If u is saturated we add W to a set T_2. Otherwise, if the last edge of W is green we add it to O and if it is red we add W to T_1. In the following claim we show that the hyperwalks in O have the desired property and we later prove that $|O|$ is large enough.

Claim 4.5.8. *Any $W \in O$ is an augmenting-hyperwalks of length at most $2/\varepsilon$ that begins and ends in unsaturated vertices.*

Proof. By construction any hyperwalk in O has length at most $2/\varepsilon$, begins with an unsaturated vertex and ends in one. Also, hyperwalks in O are edge disjoint since after adding an element (e, i) to a hyperwalk we mark e as used in subgraph $\mathcal{C}_i$ and do not add it to other hyperwalks. It only remains to prove that every hyperwalk $W = ((e_1, s_1), \ldots, (e_k, s_k)) \in O$ is indeed an augmenting-hyperwalk.

Let $P_r \Delta W$ be the result of applying W on $P_r = ((\mathcal{C}_0, M_0), \ldots, (\mathcal{C}_\alpha, M_\alpha))$. By

Definition 4.5.1, there are three conditions that $P_r \Delta W$ should satisfy if W is an augmenting-hyperwalk. The first condition is that any M'_i is a matching in $\mathcal{C}_i$ where

$$M'_i = M_i \cup \{e_j \mid j \text{ is odd, and } s_j = i\} \setminus \{e_j \mid j \text{ is even, and } s_j = i\}, \qquad \forall i \in \{0, \ldots, \alpha\}.$$

Note that W is alternating between green and red edges with green ones being in the odd positions. Further, for any element (e, i) in an odd position j and any red edge e' adjacent to it in $\mathcal{C}_i$, hyperwalk W contains (e', i) in either position $j-1$ or position $j+1$; thus the first condition is satisfied.

As for the second condition, since W is alternating between green and red edges applying it would satisfy $d_{P_r}(v) = d_{P_r \Delta W}(v)$ for any vertex v that is not an end-point. Moreover, $P_r \Delta W$ simply satisfies the third condition that is $d_{P_r}(v) + 1 = d_{P_r \Delta W}(v)$ iff v is the first or the last vertex of the hyper-walk since W begins and ends with green edges. $\square$

To complete the proof of Lemma 4.5.7, we need to show that $\mathbb{E}[|O|] \geq 8(\alpha + 1)\varepsilon\text{OPT}$. For any vertex v, let $g_{v,i}$ be the number of subgraphs $\mathcal{C}_0, \ldots \mathcal{C}_\alpha$ in which v has an unused green edge after the i-th iteration of the algorithm and similarly define $r_{v,i}$ to be the number of subgraphs in which v has an unused red edge after the i-th iteration. Each iteration here means constructing a hyperwalk and marking its edges as used. Also, let us respectively denote the set of saturated and unsaturated vertices by S and U. Consider the hyperwalk W_i constructed in the i-th iteration. Observe that if $W_i \in O$, we have

$$\sum_{v \in U} (g_{v,i-1} - r_{v,i-1}) - \sum_{v \in U} (g_{v,i} - r_{v,i}) = 2$$

since any hyperwalk in O starts from an unsaturated vertex with a green edge and ends the same way. However, if $W_i \in T_2$, we have

$$\sum_{v \in U} (g_{v,i-1} - r_{v,i-1}) - \sum_{v \in U} (g_{v,i} - r_{v,i}) = 1,$$

and if $W_i \in T_1$ we have

$$\sum_{v \in U} (g_{v,i-1} - r_{v,i-1}) - \sum_{v \in U} (g_{v,i} - r_{v,i}) = 0.$$

Moreover, for any hyper-walk in T_3 we have

$$\sum_{v \in U} (g_{v,i-1} - r_{v,i-1}) - \sum_{v \in U} (g_{v,i} - r_{v,i}) \leq 2,$$

as it holds for any hyperwalk that alternates between green and red edges. We claim that when our algorithm stops after j iterations $\sum_{v \in U} (g_{v,j} - r_{v,j}) \leq 0$ holds. This is because otherwise, we could still find a subgraph $\mathcal{C}_i$ and a vertex v where v has a green edge in $\mathcal{C}_i$ but not a red one and start a new hyperwalk. As a result we have the following lower-bound for $|O|$, where for brevity, in the rest of the proof we use g_v and r_v instead of $g_{v,0}$ and $r_{v,0}$:

$$|O| \geq \frac{1}{2}\left(\sum_{v \in U}(g_v - r_v) - |T_2| - 2|T_3|\right).$$

Taking expectations, we have

$$\mathbb{E}\big[|O|\big] \geq \frac{1}{2}\mathbb{E}\left[\sum_{v \in U}(g_v - r_v) - |T_2| - 2|T_3|\right] = \frac{1}{2}\mathbb{E}\left[\sum_{v \in U}(g_v - r_v)\right] - \frac{1}{2}\mathbb{E}\big[|T_2|\big] - \mathbb{E}\big[|T_3|\big]. \tag{4.25}$$

First, let us note that since any hyperwalk in T_3 contains at least $1/\varepsilon$ green edges and that the expected number of green edges is obviously upper bounded by OPT, we have

$$|T_3| \leq \varepsilon(\alpha + 1)\text{OPT}. \tag{4.26}$$

We now focus on bounding $\mathbb{E}\big[\sum_{v \in U}(g_v - r_v)\big]$ and prove that it is upper-bounded by $40\alpha\varepsilon\text{OPT}$.

$$\sum_{v \in U} \mathbb{E}[g_v - r_v] = \sum_{v \in V} \mathbb{E}[g_v - r_v] - \sum_{v \in S} \mathbb{E}[g_v - r_v]$$

Note that c_v, by definition, is the probability with which vertex v is matched in any M_i^g. Moreover, $\gamma_{v,r}$ is the probability with which vertex v is matched in any M_i which means $\mathbb{E}[g_v - r_v] = (\alpha + 1)(c_v - \gamma_{v,r})$ and

$$\mathbb{E}\left[\sum_{v \in V}(g_v - r_v)\right] = 2(\alpha + 1)(q(C) - \mathbb{E}[|Z_r|]).$$

Also, since $\mathbb{E}\big[|Z_r|\big] \leq q(C) - 30\varepsilon\text{OPT}$ we obtain

$$\mathbb{E}\left[\sum_{v\in V} g_v\right] - \mathbb{E}\left[\sum_{v\in V} r_v\right] \geq 60(\alpha+1)\varepsilon\text{OPT}.$$

Moreover, by definition of saturated vertices, we know that $c_v - \gamma_{v,r} \leq 2\varepsilon^2$ holds for any saturated vertex v which results in

$$\sum_{v\in S} \mathbb{E}[g_v - r_v] \leq 2n(\alpha+1)\varepsilon^2 \leq 20(\alpha+1)\varepsilon\text{OPT}. \tag{4.27}$$

Note that $2n(\alpha+1)\varepsilon^2 \leq 20(\alpha+1)\varepsilon\text{OPT}$ comes from Assumption 4.3.1 that $\text{OPT} \geq 0.1\varepsilon n$. Combining these equation, we get

$$\sum_{v\in U} \mathbb{E}[g_v - r_v] \geq 40(\alpha+1)\varepsilon\text{OPT}. \tag{4.28}$$

In the next step, we provide an upper-bound for $\mathbb{E}\big[|T_2|\big]$ and to do so we first prove the following claim.

Claim 4.5.9. *For any vertex $v \in S$ the number of hyperwalks in T_2 that end in v is $\leq |g_v - r_v|$.*

Proof. Consider the hyperwalk W that is the first one to be constructed among the hyperwalks in set T_2 that end in vertex v and let (e, i) be its last element. W.l.o.g., assume that the color of edge e in graph $\mathcal{C}_i$ is red. The fact that W stops in vertex v means that at the time of construction of this hyperwalk, there is no subgraph $\mathcal{C}_j$ that has an unused green edge of v but not a red one. Therefore, from this point of the algorithm, any subgraph $\mathcal{C}_k$ that contains an unused green edge e_g of vertex v

also has an unused red edge e_r of this vertex. We note that based on our algorithm if a hyperwalk with last element (e, i) stops at vertex v then subgraph $\mathcal{C}_i$ either does not contain a green edge of v or a red edge of this vertex. Moreover, due to the fact that W is the first hyperwalk to stop in vertex v we know that previously constructed hyperwalks contain the same number of green and red edges of vertex v. This means that there are at most $|g_v - r_v|$ many possibilities for the last element of a hyperwalk that stops at v and since our hyperwalks are edge disjoint then for any vertex $v \in S$ the number of hyperwalks in T_2 that end in v is upper-bounded by $|g_v - r_v|$. □

Based on the aforementioned claim, the number of hyperwalks ending in saturated vertices is at most $\sum_{v \in S} \mathbb{E}[g_v - r_v]$, which means $\mathbb{E}\big[|T_2|\big] \leq \sum_{v \in S} \mathbb{E}\big[|g_v - r_v|\big]$, implying further that

$$\begin{aligned}
\mathbb{E}[|T_2|] &\leq \sum_{v \in S} \mathbb{E}\big[|g_v - r_v|\big] \\
&= \sum_{v \in S} \mathbb{E}\Big[|g_v - \mathbb{E}[g_v] - r_v + \mathbb{E}[r_v] + \mathbb{E}[g_v] - \mathbb{E}[r_v]|\Big] \\
&\leq \sum_{v \in S} \mathbb{E}\Big[|g_v - \mathbb{E}[g_v]| + |r_v - \mathbb{E}[r_v]| + |\mathbb{E}[g_v] - \mathbb{E}[r_v]|\Big] \\
&\leq \sum_{v \in S} \big(\mathbb{E}\big[|g_v - \mathbb{E}[g_v]|\big] + \mathbb{E}[|r_v - \mathbb{E}[r_v]|]\big) + \sum_{v \in S} (\mathbb{E}[g_v] - \mathbb{E}[r_v]). \qquad (4.29)
\end{aligned}$$

The last equation is due to the fact that $\mathbb{E}[g_v] \geq \mathbb{E}[r_v]$ for all v.

Using a simple application of Chebyshev's inequality, we show that for any vertex v, we have $\mathbb{E}[|r_v - \mathbb{E}[r_v]|] \leq 2(\alpha + 1)^{2/3}$ and $\mathbb{E}[|r_g - \mathbb{E}[r_g]|] \leq 2(\alpha + 1)^{2/3}$. Note that we have $\mathrm{Var}(g_v) \leq \alpha + 1$ and $\mathrm{Var}(g_v) \leq \alpha + 1$. Using Chebyshev's inequality, we have $\Pr[|r_v - \mathbb{E}[r_v]| \geq \beta(\alpha+1)^{1/2}] \leq 1/\beta^2$. By setting $\beta = (\alpha+1)^{1/6}$, we get $\Pr[|r_v - \mathbb{E}[r_v]| \geq (\alpha+1)^{2/3}] \leq (\alpha+1)^{-1/3}$, which gives us $\mathbb{E}[|r_v - \mathbb{E}[r_v]|] \leq 2(\alpha+1)^{2/3}$. Similarly, we have $\mathbb{E}[|g_v - \mathbb{E}[g_v]|] \leq 2(\alpha+1)^{2/3}$. As a result, we get

$$\sum_{v\in S}(\mathbb{E}[|r_v - \mathbb{E}[r_v]|] + \mathbb{E}[|r_g - \mathbb{E}[r_g]|]) \leq 4n(\alpha+1)^{2/3}.$$

Since in $\mathsf{FindMatching}_r(\mathcal{C})$ we set $\alpha = 1/\varepsilon^7 - 1$ and since $n \leq 10\textsc{opt}/\varepsilon$ we have

$$\begin{aligned}\sum_{v\in S}(\mathbb{E}[|r_v - \mathbb{E}[r_v]|] + \mathbb{E}[|r_g - \mathbb{E}[r_g]|]) &\leq \frac{40\textsc{opt}(1/\varepsilon^7)^{2/3}}{\varepsilon} = \frac{40\textsc{opt}}{\varepsilon^{14/3}\times\varepsilon} \\ &= \frac{40\textsc{opt}}{\varepsilon^7 \times \varepsilon^{-4/3}} = 40(\alpha+1)\varepsilon^{4/3}\textsc{opt}.\end{aligned}$$

Moreover, by (4.27) we have

$$\sum_{v\in S}(\mathbb{E}[g_v] - \mathbb{E}[r_v]) \leq 20(\alpha+1)\varepsilon\textsc{opt}.$$

Combining these two bounds into (4.29) we get

$$\sum_{v\in S}\mathbb{E}\big[|T_2|\big] \leq (\alpha+1)\varepsilon\textsc{opt}(20 + 40\varepsilon^{1/3}). \tag{4.30}$$

Incorporating (4.28) and (4.30) into (4.25) and simplifying, gives

$$\mathbb{E}\big[|O|\big] \overset{(4.25)}{\geq} \frac{1}{2}\mathbb{E}\Big[\sum_{v\in U}(g_v - r_v)\Big] - \frac{1}{2}\mathbb{E}\big[|T_2|\big] - \mathbb{E}\big[|T_3|\big]$$
$$\overset{(4.28),\,(4.30),\,(4.26)}{\geq} (\alpha+1)\varepsilon\textsc{opt}(9 - 20\varepsilon^{1/3}).$$

By letting ε be small enough, we can assume that $20\varepsilon^{1/3} \leq 1$ and get

$$\mathbb{E}[|O|] \geq 8(\alpha+1)\varepsilon\textsc{opt},$$

which completes the proof of Lemma 4.5.7. This completes all the components needed within the proof of Lemma 4.5.3 which as discussed at the start of the section, implies the needed bound on the expected size of the matching returned.

4.5.4 Lemma 4.3.8 Property 2: Matching Probabilities

In this section, we prove that algorithm $\mathsf{FindMatching}_t(\mathcal{C})$ satisfies property 2 of Lemma 4.3.8 that for each vertex v, $\Pr[X_v] \leq \max\{c_v - \varepsilon^2, 0\}$. Recall that X_v, as defined in Lemma 4.3.8, is the indicator of the event that v is matched in $\mathsf{FindMatching}_t(\mathcal{C})$, and the probability is taken over both the realization $\mathcal{C}$ and the randomization of algorithm $\mathsf{FindMatching}_t(\mathcal{C})$.

Let us use $X_{v,r}$ to denote the event that vertex v gets matched in matching $\mathsf{FindMatching}_r(\mathcal{C})$. It holds that $X_{v,t} = X_v$. Therefore, it suffices to show that $\Pr[X_{v,t}] \leq \max\{c_v - \varepsilon^2, 0\}$. We will, however, prove a stronger claim:

Claim 4.5.10. *For every integer r and for every vertex v, it holds that* $\Pr[X_{v,r}] \leq$

$\max\{c_v - \varepsilon^2, 0\}$.

We prove this by indiction on r. For the base case $r = 0$, the algorithm $\mathsf{FindMatching}_0(\mathcal{C})$ returns an empty matching $\emptyset$. Therefore $\Pr[X_{v,0}] = 0$ for all vertices v, clearly satisfying the claim. For the induction step, fix any vertex v. We suppose that $\Pr[X_{v,r-1}] \leq \max\{c_v - \varepsilon^2, 0\}$ and prove that it continues to hold that $\Pr[X_{v,r}] \leq \max\{c_v - \varepsilon^2, 0\}$. We start with a definition.

Definition 4.5.11. *Define ρ_v to be the fraction of matchings $M_0, \ldots, M_\alpha$ in which v is matched and define ρ'_v similarly with respect to matchings $M'_0, \ldots, M'_\alpha$. More precisely,*

$$\rho_v := \frac{|\{i : v \in V(M_i)\}|}{\alpha + 1}, \qquad \text{and} \qquad \rho'_v := \frac{|\{i : v \in V(M'_i)\}|}{\alpha + 1}.$$

Observation 4.5.12. $\mathbb{E}[\rho_v] = \Pr[X_{v,r-1}]$ *and* $\mathbb{E}[\rho'_v] = \Pr[X_{v,r}]$.

Proof. For any $i \in \{0, \ldots, \alpha\}$, we have $\Pr[v \in V(M_i)] = \Pr[X_{v,r-1}]$ since $M_i = \mathsf{FindMatching}_{r-1}(\mathcal{C}_i)$ and $\mathcal{C}_i$ is picked from the same distribution that the actual realization $\mathcal{C}$ is picked from. Thus:

$$\begin{aligned}\mathbb{E}[\rho_v] &= \mathbb{E}\left[\frac{\sum_{i=0}^{\alpha} \mathbb{1}(v \in V(M_i))}{\alpha + 1}\right] = \frac{1}{\alpha+1}\sum_{i=0}^{\alpha} \Pr[v \in V(M_i)] \\ &= \frac{1}{\alpha+1}\sum_{i=0}^{\alpha} \Pr[X_{v,r-1}] = \Pr[X_{v,r-1}].\end{aligned}$$

For the second equality, first observe that since M'_0 is the matching returned by $\mathsf{FindMatching}_r(\mathcal{C})$, then $X_{v,r}$ is by definition exactly the event that $v \in V(M'_0)$ and thus $\Pr[X_{v,r}] = \Pr[v \in V(M'_0)]$. Moreover, due to symmetry of the algorithm in constructing

$M'_0, \dots, M'_\alpha$, it holds for any $i \in [\alpha]$ that $\Pr[v \in V(M'_i)] = \Pr[v \in V(M'_0)] = \Pr[X_{v,r}]$. Therefore, we get:

$$\mathbb{E}[\rho'_v] = \mathbb{E}\left[\frac{\sum_{i=0}^{\alpha} \mathbb{1}(v \in V(M'_i))}{\alpha+1}\right] = \frac{1}{\alpha+1}\sum_{i=0}^{\alpha} \Pr[v \in V(M'_i)]$$
$$= \frac{1}{\alpha+1}\sum_{i=0}^{\alpha} \Pr[X_{v,r}] = \Pr[X_{v,r}],$$

concluding the proof. □

In algorithm $\mathsf{FindMatching}_r(\mathcal{C})$, we mark v as either saturated or unsaturated depending on the value of $\gamma_{v,r-1}$. Note from definition of $\gamma_{v,r-1}$ that $\gamma_{v,r-1} = \Pr[X_{v,r-1}]$. Therefore, v is marked as saturated if $\Pr[X_{v,r-1}] \geq c_v - 2\varepsilon^2$ and unsaturated if $\Pr[X_{v,r-1}] < c_v - 2\varepsilon^2$. We consider the two cases individually.

If v is saturated. In this case, by definition of graph H, vertex v cannot start or end any augmenting-hyperwalk with a corresponding vertex in H (and for that matter in I). By definition of augmenting-hyperwalks, for all vertices (except the endpoints of the walk) applying the hyperwalk does not change the number of matchings in which the vertex is part of. Therefore, if v is saturated, $\rho_v = \rho'_v$ and thus $\Pr[X_{v,r}] = \Pr[X_{v,r-1}] \leq \max\{c_v - \varepsilon^2, 0\}$ where the latter inequality comes from the induction's hypothesis.

If v is unsaturated. Note that in graph H by definition we have edges between any pair of augmenting-hyperwalks that share a vertex in the graph. Therefore, the independent set I of H can include at most one augmenting-hyperwalk W that includes vertex v. If v is not an end-point of W, then as in the case above, we get $\rho_v = \rho'_v$. However, if v is an end-point of W, then by definition of augmenting-hyperwalks, there will be one (and only one) i where $v \in V(M'_i)$ and $v \notin V(M_i)$. In this case, we get that

$$\rho'_v = \frac{|\{i : v \in V(M'_i)\}|}{\alpha+1} = \frac{|\{i : v \in V(M_i)\}|+1}{\alpha+1} = \rho_v + \frac{1}{\alpha+1} \overset{\alpha=1/\varepsilon^7-1}{<} \rho_v + \varepsilon^2.$$

Since in this case, we had $\rho_v < c_v - 2\varepsilon^2$, we get $\rho'_v < c_v - 2\varepsilon^2 + \varepsilon^2 = c_v - \varepsilon^2$. Therefore the induction's hypothesis still holds that $\Pr[X_{v,r}] < \max\{c_v - \varepsilon^2, 0\}$, completing the proof of Claim 4.5.10.

4.5.5 Lemma 4.3.8 Property 4: Matching Independence

In this section, we prove that algorithm $\mathsf{FindMatching}_t(\mathcal{C})$ satisfies property 4 of Lemma 4.3.8. That is, for every subset $I = \{v_1, \ldots, v_k\}$ of the vertices such that $d_C(v_i, v_j) \geq \lambda$ for all $v_i, v_j \in I$, random variables $X_{v_1}, \ldots, X_{v_k}$ are independent. Recall that X_v for a vertex v is the indicator of the event that v is matched in the matching returned by $\mathsf{FindMatching}_t(\mathcal{C})$. We also, again, emphasize that this "independence" is with regards to the randomization of realization $\mathcal{C}$ of C on which Z is constructed, and the randomization of algorithm $\mathsf{FindMatching}_t(\mathcal{C})$ itself.

In Section 4.5.1 we gave an overview of how we can argue about such independence via an implementation of the algorithm in the LOCAL model of computation. Here we give this implementation.

Initialization. The communication network is graph C. Each node v is initially given the following information: Its incident edges in C and how they are realized, the maximum degree Δ_C of graph C, parameter ε, and the value of c_v. Note that to gather information about realization of edges further away, the nodes need to communicate.

Also note that even though the value of c_v may reveal some information about graph G (or C), it crucially reveals no information about the realization $\mathcal{C}$ of C, or other sources of randomization used by the algorithm. Thus, property 4 can still be satisfied if we manage to show the algorithm can be implemented in few rounds.

The $\mathsf{ApproximateMIS}(H, \varepsilon)$ **algorithm.** We start by mentioning that subroutine $\mathsf{ApproximateMIS}(H, \varepsilon)$ already has an efficient LOCAL implementation whose round-complexity depends only on the maximum degree of H and ε, without essentially any dependence on the number of nodes in H. Any implementation with such round-complexity can be used in our case. For instance, we use one implied in [27] (see Appendix 4.8 for details):

Lemma 4.5.13 ([27])**.** *Given a graph H of max degree Δ and any parameter ε, there is a LOCAL algorithm $\mathsf{ApproximateMIS}(H, \varepsilon)$ that returns an independent set I of H in $O(\log \frac{\Delta}{\varepsilon})$ rounds such that the expected size of I is at least $(1 - \varepsilon)$ fraction of some maximal independent set of H.*

We give a LOCAL implementation of Algorithm 2 which proves the following:

Claim 4.5.14. *For any $r \geq 0$, algorithm $\mathsf{FindMatching}_r(\cdot)$ can be implemented in $O(r\varepsilon^{-4} \log \Delta_C)$ rounds of LOCAL.*

Proof. We prove the claim by induction on r. For the base case, observe that algorithm $\mathsf{FindMatching}_0(\cdot)$ can be implemented in 0 rounds since the output is always the empty matching. We assume that algorithm $\mathsf{FindMatching}_{r-1}(\cdot)$ can be implemented in $\beta(r - 1)\varepsilon^{-4} \log \Delta_C$ rounds where $\beta > 1$ is a sufficiently large absolute constant that we fix later, and prove that $\mathsf{FindMatching}_r(\cdot)$ can be implemented in $\beta r \varepsilon^{-4} \log \Delta_C$ rounds.

Step 1. First, the algorithm draws α realizations $\mathcal{C}_1, \ldots, \mathcal{C}_\alpha$. Since information about realization of edges is stored locally on their incident vertices, we can easily generate these random realizations in $O(1)$ rounds. After that, on each graph $\mathcal{C}_i$ for $i \in \{0, \ldots, \alpha\}$, we recursively run the $(\beta(r-1)\varepsilon^{-4} \log \Delta_C)$-round implementation of $\mathsf{FindMatching}_{r-1}(\mathcal{C}_i)$. Note that all of these can run in parallel. The overall round-complexity of this step, is thus $\beta(r-1)\varepsilon^{-4} \log \Delta_C + O(1)$.

Step 2. Next, we need to compute $\gamma_{v,r-1}$ for each vertex v, which recall is the probability that v is matched in $\mathsf{FindMatching}_{r-1}(\mathcal{C}')$ where $\mathcal{C}'$ is a random realization of C. The crucial observation here is that since $\mathsf{FindMatching}_{r-1}(\cdot)$ can, by the induction hypothesis, be implemented within only $\beta(r-1)\varepsilon^{-4} \log \Delta_C$ rounds, $\gamma_{v,r-1}$ is merely a function of the topology induced in the $(\beta(r-1)\varepsilon^{-4} \log \Delta_C)$-hop of v. We first gather this neighborhood of v, which can be done in $(\beta(r-1)\varepsilon^{-4} \log \Delta_C)$ rounds, then compute $\gamma_{v,r-1}$. We note that this gathering part can be done in parallel to the operations of Step 1. Therefore, overall, Steps 1 and 2 take $(\beta(r-1)\varepsilon^{-4} \log \Delta_C + O(1))$ rounds. Having $\gamma_{v,r-1}$ for each vertex v, we can then determine for each vertex whether it is saturated or unsaturated since we are given the value of c_v in the initialization step.

Step 3. The next step is constructing graph H. In graph H, each vertex corresponds to a walk of size at most $2/\varepsilon$ in C. Therefore, each vertex in C can first gather all such walks around it in $O(1/\varepsilon)$ rounds, and then determine which one of them are augmenting-hyperwalks satisfying the required properties to be considered as a node of H. Determining the edges of H can also be done locally; once we construct the vertices, there will be an edge between any two walks that share a vertex. Therefore, overall, graph H can be constructed in $O(1/\varepsilon)$ rounds.

Step 4. After constructing H, in this step, we run the LOCAL implementation of $\mathsf{ApproximateMIS}(H, \varepsilon)$ mentioned in Lemma 4.5.13 on graph H. We emphasize that our communication network here is graph C, not H. However, any message between two nodes of H can be sent over network C within $O(1/\varepsilon)$ rounds. This is because any two incident nodes of H, are walks of size at most $O(1/\varepsilon)$ in C that share at least a vertex. The overall running time of this procedure is thus $O(\frac{1}{\varepsilon} \times \log \frac{\Delta_H}{\varepsilon})$. We note that $\Delta_H = O(\varepsilon^{-1}((\alpha+1)\Delta_C)^{2/\varepsilon})$. To see this, fix any walk w with a corresponding node in H. This walk has at most $2/\varepsilon$ nodes in C. Now each node in C is incident to $O(((\alpha+1)\Delta_C)^{2/\varepsilon})$ hyperwalks: There are $O((\Delta_C)^{2/\varepsilon})$ walks of size $\leq 2/\varepsilon$ branching out of each of the nodes, and each edge of the walk can take on $\alpha + 1$ labels from $\{0, \ldots, \alpha\}$ to be transformed to a hyperwalk. Therefore, overall the number of rounds required for this part of the algorithm is

$$O\left(\frac{1}{\varepsilon} \times \log \frac{\Delta_H}{\varepsilon}\right) = O\left(\frac{1}{\varepsilon} \log \frac{\varepsilon^{-1}((\alpha+1)\Delta_C)^{2/\varepsilon}}{\varepsilon}\right) = O\big(\varepsilon^{-4} \log \Delta_C\big),$$

where the last equality comes from the fact that $\alpha = \text{poly}(\varepsilon^{-1})$.

Step 5. Finally, applying the augmenting-hyperwalks chosen in I is simple and can be done in $O(1/\varepsilon)$ rounds since these walks are of size $\leq 2/\varepsilon$.

Round-complexity. Let β_2 be a sufficiently large constant by multiplying which we can surpass the O-notations. We get

$$\begin{aligned}
\#\text{ of rounds} &\leq \underbrace{\beta(r-1)\varepsilon^{-4}\log\Delta_C + \beta_2}_{\text{Steps 1 and 2}} + \underbrace{\beta_2(1/\varepsilon)}_{\text{Step 3}} + \underbrace{\beta_2(\varepsilon^{-4}\log\Delta_C)}_{\text{Step 4}} + \underbrace{\beta_2(1/\varepsilon)}_{\text{Step 5}} \\
&< \beta(r-1)\varepsilon^{-4}\log\Delta_C + 4\beta_2(\varepsilon^{-4}\log\Delta_C) \\
&= \Big(\beta(r-1) + 4\beta_2\Big)\varepsilon^{-4}\log\Delta_C.
\end{aligned}$$

Since β_2 is an absolute constant that does not depend on β, we can set β to be large enough with respect to it. Setting $\beta = 4\beta_2$ is sufficient since

$$\Big(\beta(r-1) + 4\beta_2\Big)\varepsilon^{-4}\log\Delta_C = \beta r\varepsilon^{-4}\log\Delta_C.$$

This concludes the proof of the induction step, and consequently the proof of Claim 4.5.14.

□

We showed in Claim 4.5.14 that algorithm $\mathsf{FindMatching}_r(\mathcal{C})$, for any r, can be implemented within $O(r\varepsilon^{-4}\log\Delta_C)$ rounds of LOCAL. Our final algorithm for Lemma 4.3.8 is $\mathsf{FindMatching}_t(\mathcal{C})$ where we set $t = 1/\varepsilon^9$. Thus, the output of each vertex can be determined within $\lambda' = O(\varepsilon^{-13}\log\Delta_C)$ rounds. This, as described, proves property 4 of Lemma 4.3.8 since $\lambda = \varepsilon^{-20}\log\Delta_C$ is larger than $\lambda'/2$ given that ε is small enough to surpass the hidden constants in the O-notation. (Recall that we can assume ε is smaller than any needed constant.)

4.6 Concentration of the Maximum Realized Matching's Size

In this section, we prove that random variable $\mu(\mathcal{G})$, i.e. the size of the maximum realized matching of G, is highly concentrated around its mean $\mathbb{E}[\mu(\mathcal{G})] = \text{OPT}$. A similar concentration bound was previously proved also in the works of [3, 16]. Nonetheless, we provide the full proof in this section for the sake of self-containment.

Lemma 4.6.1. *For every* $0 < t \leq \text{OPT}$, $\Pr[|\mu(\mathcal{G}) - \text{OPT}| \geq t] \leq \exp\left(-\frac{t^2}{2\text{OPT}+2t/3}\right) < \exp\left(-\frac{t^2}{3\text{OPT}}\right)$.

Corollary 4.6.2. *Let Q be a subgraph of G obtained via a deterministic algorithm and suppose that* $\text{OPT} = \omega(1)$. *If* $\mathbb{E}[\mu(\mathcal{Q})]/\mathbb{E}[\mu(\mathcal{G})] \geq \alpha$ *then with high probability* $\mu(\mathcal{Q})/\mu(\mathcal{G}) \geq (1 - o(1))\alpha$.

Proof. Lemma 4.6.1 implies that w.h.p. $\mu(\mathcal{Q}) = (1 \pm o(1))\mathbb{E}[\mu(\mathcal{Q})]$ and $\mu(\mathcal{G}) = (1 \pm o(1))\mathbb{E}[\mu(\mathcal{G})]$. Therefore, w.h.p. $\mu(\mathcal{Q})/\mu(\mathcal{G}) = (1 \pm o(1))\mathbb{E}[\mu(\mathcal{Q})]/\mathbb{E}[\mu(\mathcal{G})] \geq (1 - o(1))\alpha$. □

We note that our construction of subgraph Q in Algorithm 2 is randomized, thus the corollary above cannot be used as a black-box to imply a high probability bound. However, we remark that a similar proof to that of Lemma 4.6.1 which we give below, proves $\mu(\mathcal{Q})$ in our algorithm is concentrated around its mean even considering the randomization of Algorithm 2. Therefore, our algorithm also guarantees a high probability bound for the approximation-factor.

In order to prove this lemma, we use the concentration of "self-bounding" functions. See Sections 3.3 and 6.7 of book [19] by Boucheron, Lugosi and Massart for a thorough discussion on this concentration inequality and its proof.

Definition 4.6.3 ([19, Section 6.7])**.** *A function $f : \mathcal{X}^m \to \mathbb{R}$ is "self-bounding" if for every $i \in [m]$ there is a function $f_i : \mathcal{X}^{m-1} \to \mathbb{R}$ such that for all $x = (x_1, \ldots, x_m) \in \mathcal{X}^m$,*

1. $0 \le f(x) - f_i(x^{(i)}) \le 1$ *for all* $i \in [m]$*, and*

2. $\sum_{i=1}^{m}(f(x) - f_i(x^{(i)})) \le f(x)$*,*

where $x^{(i)} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$*.*

Lemma 4.6.4 ([19, Theorem 6.12])**.** *If $X_1, \ldots, X_m$ are independent random variables taking values in $\mathcal{X}$ and $Z = f(X_1, \ldots, X_m)$ is self-bounding, then for every $0 < t \le \mathbb{E}Z$,*

$$\Pr[|Z - \mathbb{E}Z| \ge t] \le \exp\left(-\frac{t^2}{2\mathbb{E}Z + 2t/3}\right).$$

Having this inequality, Lemma 4.6.1 follows as follows.

Proof of Lemma 4.6.1. Let X_e for each edge e in graph G be the indicator of the event that e is realized. We can use vector $X = (X_{e_1}, \ldots, X_{e_m})$ to represent a realization of G where $e_1, \ldots, e_m$ are all edges in G. With a slight abuse of notation, we use $\mu(X)$ to denote the size of the maximum matching in realization X. We first prove that function $\mu(X)$ is self-bounding. For each $i \in [m]$, define

$$\mu_i(X^{(i)}) = \mu(X_{e_1}, \ldots, X_{e_{i-1}}, 0, X_{e_{i+1}}, \ldots, X_{e_m}).$$

In words, $\mu_i(X^{(i)})$ is the maximum matching size in realization X if we regard edge e_i

as unrealized. We need to show that the two conditions of Definition 4.6.3 hold. First, we have to show that

$$0 \leq \mu(X) - \mu_i(X^{(i)}) \leq 1 \qquad \text{for all } i \in [m] \text{ and all realizations } X.$$

Observe that removing a realized edge cannot increase the maximum realized matching size, thus clearly $\mu(X) - \mu(X^{(i)}) \geq 0$. Moreover, removing each edge decreases the maximum matching size by at most 1. Thus $\mu(X) - \mu(X^{(i)}) \leq 1$ proving the first condition. For the second condition, we have to show that

$$\sum_{i=1}^{m} \big(\mu(X) - \mu_i(X^{(i)})\big) \leq \mu(X).$$

To see this, fix a maximum realized matching M in realization X. For any edge e_i outside this matching, we have $\mu(X) - \mu_i(X^{(i)}) = 0$. For the rest, as discussed above $\mu(X) - \mu_i(X^{(i)}) \leq 1$. Therefore indeed $\sum_{i=1}^{m} \big(\mu(X) - \mu_i(X^{(i)})\big) \leq |M| = \mu(X)$.

We proved that $\mu(X)$ is self-bounding. Since the edges are realized independently, we can plug this into Lemma 4.6.4 and immediately obtain Lemma 4.6.1. □

4.7 On Generality of Assumption 4.3.1

In this section, we prove that Assumption 4.3.1 comes without loss of generality. Precisely, we show that solving the problem for any input graph G can be reduced to solving it for a graph H with $O(\text{OPT}/\varepsilon)$ vertices and $\mathbb{E}[\mu(\mathcal{H})] \geq (1-\varepsilon)\text{OPT}$ where $\mathcal{H}$ is a realization H. To do this, we use a "vertex sparsification" idea of Assadi *et al.* [5].

Our reduction is slightly different since we do not want parallel edges in the graph, but the main idea is essentially the same. It is also worth noting that for the reduction to work, it is crucial that our algorithm works for different edge realization probabilities. We provide the full proof for completeness.

We note that throughout the proof we may assume that OPT is larger than constant $3\varepsilon^{-3}$ and remark that the problem otherwise is trivial.

Construction of H from G. We construct graph $H = (U, F)$ as follows. For $k = \frac{8\text{OPT}}{\varepsilon}$, define k *buckets* $U = \{u_1, \ldots, u_k\}$. Each of these buckets u_i will correspond to a node in H. Assign each vertex v of graph G to a bucket $b(v) \in \{u_1, \ldots, u_k\}$ picked independently and uniformly at random. Then for any edge $\{v_1, v_2\}$ in graph G, we add an edge $\{b(v_1), b(v_2)\}$ to F. Finally, we turn H into a simple graph by removing self-loops and merging parallel edges.

Now we need to set the realization probability p_e of every edge $e \in F$ as well. For any $e \in F$, let us denote by $E(e)$ the set of edges in the original graph G that are mapped to e. We set

$$p_e := 1 - \prod_{e' \in E(e)} (1 - p_{e'}).$$

We note that p_e is defined such that it precisely equals to the probability that at least one edge in $E(e)$ is realized.

Claim 4.7.1. *Fix any matching M in G satisfying $|M| \leq 2\text{OPT}$. Then $\mathbb{E}[\mu(H)] \geq (1 - \varepsilon)|M|$ where the expectation is taken over the randomization of the algorithm in constructing H.*

Proof. Let $V(M)$ be the vertex-set of matching M in graph G and define

$$X := \{v \in V(M) \mid \exists u \in V(M) \text{ s.t. } v \neq u \text{ and } b(v) = b(u)\},$$

which is the set of vertices in $V(M)$ whose bucket is not unique with regards to others in $V(M)$.

We first claim that $\mu(H) \geq |M| - |X|$. Call an edge $\{u, v\} \in M$ *good* if $u \notin X$, $v \notin X$, and *bad* otherwise. Each bad edge has at least one endpoint in X, thus there are at least $|M|-|X|$ good edges in M. One can easily confirm that the set of corresponding edges of all good edges in M forms a matching in H. Thus $\mu(H) \geq |M| - |X|$.

To conclude, we prove that $\mathbb{E}[|X|] \leq \varepsilon|M|$ which proves $\mathbb{E}[\mu(H)] \geq |M| - \varepsilon|M| = (1 - \varepsilon)|M|$. To see why $\mathbb{E}[|X|] \leq \varepsilon|M|$, fix any vertex $v \in V(M)$ and suppose that we have adversarially fixed the bucket $b(u)$ of all other vertices $u \in V(M)$. Since the bucket of v is picked uniformly at random from $10\textsc{opt}/\varepsilon$ buckets and $|V(M)| \leq 2|M| \leq 4\textsc{opt}$, the probability of v choosing a bucket already chosen by another vertex in $V(M)$ would be $\leq \frac{4\textsc{opt}}{8\textsc{opt}/\varepsilon} \leq \varepsilon/2$. By linearity of expectation over $2|M|$ vertices in $V(M)$, we get $\mathbb{E}[|X|] \leq \varepsilon|M|$, concluding the proof. □

Claim 4.7.2. *It holds that* $\mathbb{E}[\mu(\mathcal{H})] \geq (1 - 3\varepsilon)\textsc{opt}$*. Here the expectation is taken over both the randomization in construction of* H *and the randomization in realization* $\mathcal{H}$ *of* H*.*

Proof. We first map each realization $\mathcal{G}$ of G to a realization $\mathcal{H}$ of H. To do so, we say an edge $e \in F$ is realized in $\mathcal{H}$ if and only if at least one edge $e' \in E(e)$ is realized in $\mathcal{G}$. We argue that this mapping preserves independence of edge realizations in H and their realization probabilities. First, since for any two edges $e_1, e_2 \in F$ it holds that $E(e_1) \cap E(e_2) = \emptyset$, realization of an edge $e \in F$ gives no information regarding realization of other edges. Moreover, observe that each edge $e \in F$ will be precisely realized with probability p_e as discussed above in defining p_e.

Let M be the maximum realized matching of G. By Lemma 4.6.1, $\Pr[||M| - \textsc{opt}| \geq \varepsilon\textsc{opt}] < \exp(-\frac{(\varepsilon\textsc{opt})^2}{3\textsc{opt}}) = \exp(-\frac{\varepsilon^2\textsc{opt}}{3}) < \varepsilon$ where the last inequality follows from assumption $\textsc{opt} > 3\varepsilon^{-3}$. This means that with probability at least $1-\varepsilon$, $|M| \in [(1-\varepsilon)\textsc{opt}, (1+\varepsilon)\textsc{opt}]$. Let us suppose that this event holds and denote it by A. Note that event A is only with regards to realization of G and reveals no information about the algorithm to construct H. Now plugging matching M into Claim 4.7.1, we get that $\mathbb{E}[\mu(\mathcal{H}) \mid A] \geq (1-\varepsilon)|M| \geq (1-\varepsilon)(1-\varepsilon)\textsc{opt} \geq (1-2\varepsilon)\textsc{opt}$. Incorporating also the probability that event A holds, which as described is at least $1-\varepsilon$, we get $\mathbb{E}[\mu(\mathcal{H})] \geq (1-\varepsilon)(1-2\varepsilon)\textsc{opt} \geq (1-3\varepsilon)\textsc{opt}$, concluding the proof. □

The reduction. We are now ready to give the full reduction. Suppose we are given n-vertex graph G with $\textsc{opt} = \mathbb{E}[\mu(\mathcal{G})]$ and assume that $\textsc{opt} < 0.1\varepsilon n$ (otherwise Assumption 4.3.1 holds). We first construct graph H as described. Note that H has at most $n' = \frac{8\textsc{opt}}{\varepsilon}$ nodes by the construction and that $\mathbb{E}[\mu(\mathcal{H})] \geq (1-3\varepsilon)\textsc{opt}$ by Claim 4.7.2. Replacing $\textsc{opt}$ with $\varepsilon n'/8$, we get $\mathbb{E}[\mu(\mathcal{H})] \geq (1-3\varepsilon)\frac{\varepsilon n'}{8}$. Assuming $\varepsilon < 0.05$ (recall that we can assume ε to be smaller than any needed constant), this implies $\mathbb{E}[\mu(\mathcal{H})] \geq \frac{\varepsilon n'}{10}$ and thus Assumption 4.3.1 holds for graph H.

Let Q be the result of running Algorithm 2 on graph H. Since Assumption 4.3.1

holds for H, it leads to a $(1-\varepsilon)$-approximation. That is, we get $\mathbb{E}[\mu(\mathcal{Q})] \geq (1 - \Omega(\varepsilon))\mathbb{E}[\mu(\mathcal{H})]$. We use this subgraph Q to pick a bounded-degree subgraph Q' of G that provides a $(1-\varepsilon)$-approximation: For each edge $e \in Q$, let us *pick* $\min\{p^{-1}\log\varepsilon^{-1}, |E(e)|\}$ arbitrary edges from $E(e)$ and put them in Q'. We argue that this subgraph Q' has maximum degree $O_{\varepsilon,p}(1)$ and that $\mathbb{E}[\mu(\mathcal{Q}')] \geq (1-\Omega(\varepsilon))$OPT.

Claim 4.7.3. *Q' has maximum degree $O_{\varepsilon,p}(1)$.*

Proof. Observe that an edge e' incident to a vertex $v \in V$ is in Q' only if its corresponding edge e in graph H is in Q. Since e corresponds to e', it should be incident to $b(v)$ of v by the construction of H. Moreover, since $b(v)$ has maximum degree $O_{\varepsilon,p}(1)$ in Q and that for each edge incident to $b(v)$ in Q, we put at most $O(p^{-1}\log\varepsilon^{-1})$ edges in Q', the degree of v in Q' is bounded by $O_{\varepsilon,p}(1) \times O(p^{-1}\log\varepsilon^{-1}) = O_{\varepsilon,p}(1)$. This bounds the maximum degree of Q' by $O_{\varepsilon,p}(1)$. □

Claim 4.7.4. $\mathbb{E}[\mu(\mathcal{Q}')] \geq (1-\Omega(\varepsilon))$OPT.

Proof. For any edge $e \in Q$, define p'_e to be the probability that at least one of the edges in G picked for e is realized. We first argue that $p'_e \geq (1-\varepsilon)p_e$. To see this, note that if $|E(e)| \leq p^{-1}\log\varepsilon^{-1}$, then all the edges in $E(e)$ will be picked. Thus by definition of p_e we have $p'_e = p_e$. On the other hand, if $|E(e)| > p^{-1}\log\varepsilon^{-1}$, we pick exactly $p^{-1}\log\varepsilon^{-1}$ edges for e. Since each of these edges has realization probability at least p, the probability that at least one of them is realized is at least

$$1-(1-p)^{p^{-1}\log\varepsilon^{-1}} \geq 1-\varepsilon \geq (1-\varepsilon)p_e.$$

Now let M be any matching in Q. For each edge $e \in M$, choose one arbitrary edge in $E(e)$. From the construction of H from G, one can confirm that the set of these chosen edges will form a matching of size $|M|$ in G. This concludes the proof: For each edge $e \in Q$, there is a probability at least $(1-\varepsilon)p_e$ that one picked edge in Q' is realized, thus $\mathbb{E}[\mu(\mathcal{Q}')] \geq (1-\varepsilon)\mathbb{E}[\mu(\mathcal{Q})]$. As it was previously shown that $\mathbb{E}[\mu(\mathcal{Q})] \geq (1-\Omega(\varepsilon))\textsc{opt}$, we conclude that $\mathbb{E}[\mu(\mathcal{Q}')] \geq (1-\Omega(\varepsilon))\textsc{opt}$. □

4.8 Approximate MIS

In this section we describe how Lemma 4.5.13 can be derived as a corollary of the algorithm of [27]. Theorem 1.1 of [27] gives a randomized LOCAL independent-set (IS) algorithm which guarantees that for each node v, the probability that v "has not made its decision" after $O(\log \deg(v) + \log \frac{1}{\delta})$ rounds is at most δ. The decision of v is finalized if it is in the IS or it has a neighbor that is in the IS (implying that v cannot be in the IS).

To achieve Lemma 4.5.13 we set $\delta = \frac{\varepsilon}{10\Delta}$. Let I denote the independent set returned by the algorithm after $O(\log \deg(v) + \log \frac{10\Delta}{\varepsilon}) = O(\log \frac{\Delta}{\varepsilon})$ rounds and let U and D respectively denote the set of undecided and decided vertices. We have

$$\mathbb{E}[|U|] = \mathbb{E}\Big[\sum_v \mathbb{1}(v \text{ is undecided})\Big] = \sum_v \Pr[v \text{ is undecided}] \leq \sum_v \frac{\varepsilon}{10\Delta} = \frac{\varepsilon}{10\Delta} n,$$

and thus $\mathbb{E}[|D|] = n - \mathbb{E}[|U|] \geq (1-\frac{\varepsilon}{10\Delta})n \geq 0.9n$. There is at least one IS node among the at most $\Delta+1$ inclusive neighbors of any decided vertex; thus $\mathbb{E}[|I|] \geq \frac{\mathbb{E}[|D|]}{\Delta+1} \geq \frac{0.9n}{\Delta+1} \geq \frac{0.9n}{2\Delta} = 0.45\frac{n}{\Delta}$. On the other hand, let I' be the MIS obtained by greedily adding the undecided nodes to I until they form an MIS. We have $|I'| \leq |I| + |U|$. Therefore, we indeed get that

$$\frac{\mathbb{E}[|I|]}{\mathbb{E}[|I'|]} \geq \frac{\mathbb{E}[|I|]}{\mathbb{E}[|I|] + \mathbb{E}[|U|]} \geq \frac{0.45\frac{n}{\Delta}}{0.45\frac{n}{\Delta} + \frac{\varepsilon}{10\Delta}n} = \frac{0.45\frac{n}{\Delta}}{(0.45 + 0.1\varepsilon)\frac{n}{\Delta}} = \frac{0.45}{0.45 + 0.1\varepsilon} > 1 - \varepsilon,$$

concluding the proof.

4.9 Deferred Proofs

Proof of Lemma 4.3.5. Let $t_0 = (\varepsilon p)^{50}$ and for any $i \geq 1$ let $t_i = f(t_{i-1})$. Note that $t_0 > t_1 > t_2 > \dots$ by the assumption of the lemma that $0 < f(x) < x$ for all $0 < x < 1$. For any $i \geq 1$ define $q_i = \sum_{e \in E: q_e \in (t_i, t_{i-1}]} q_e$ and let j be the smallest number where $q_j \leq \varepsilon\textsc{opt}$. We will soon prove existence of such j and also prove that $j = O(1/\varepsilon)$. We claim that setting $\tau_+ = t_{j-1}$ and $\tau_- = t_j$ satisfies the conditions of the lemma.

Condition (1): This condition holds trivially since $\tau_- = t_j = f(t_{j-1}) = f(\tau_+)$.

Condition (2): Let us define $X := \{e \mid \tau_- < q_e < \tau_+\}$. Recall that crucial and non-crucial edges are defined based on τ_+ and τ_-. That is, an edge e is crucial (i.e. $e \in C$) if $q_e \geq \tau_+$, and is non-crucial (i.e. $e \in N$) if $q_e \leq \tau_-$. This implies that the remaining edges that are neither crucial nor non-crucial belong to X. Therefore,

$$\textsc{opt} = q(E) = q(C) + q(N) + q(X).$$

To obtain $q(N) + q(C) \geq (1-\varepsilon)\textsc{opt}$ it thus suffices to show $q(X) \leq \varepsilon\textsc{opt}$. Noting that $\tau_+ = t_{j-1}$ and $\tau_- = t_j$ and also noting the definition of q_j above, we get $q(X) \leq q_j$.

Recall that we chose j such that $q_j \leq \varepsilon\text{OPT}$. Therefore we indeed get that $q(X) \leq \varepsilon\text{OPT}$.

Condition (3): We defined $t_0 = (\varepsilon p)^{50}$ and recursively defined $t_i = f(t_{i-1})$. Since $f(\cdot)$ is only a function of its input, we get via a simple induction that both t_j and t_{j-1} are also functions of only ε and p. (Recall that $j = O(1/\varepsilon)$.)

Condition (4): We defined $t_0 = (\varepsilon p)^{50}$ and recall that we showed $t_0 > t_1 > t_2 > \ldots$; this implies clearly that $\tau_+ = t_{j-1} \leq (\varepsilon p)^{50}$.

Existence of j. It only remains to prove that there exists a choice of j satisfying $q_j \leq \varepsilon\text{OPT}$ and that this j is not too large. Precisely, we show that $j = O(1/\varepsilon)$. Since intervals $(t_1, t_0], (t_2, t_1], (t_3, t_2], \ldots$ are disjoint, it holds that for each edge e there is at most one i for which $q_e \in (t_i, t_{i-1}]$. This means that $\sum_{i=1}^{\infty} q_i \leq \sum_{e \in E} q_e = \text{OPT}$. It thus has to hold that $j \leq \lceil 1/\varepsilon \rceil + 1$ or otherwise

$$\sum_{i=1}^{j-1} q_i \geq \sum_{i=1}^{\lceil 1/\varepsilon \rceil + 1} \varepsilon\text{OPT} = (\lceil 1/\varepsilon \rceil + 1)\varepsilon\text{OPT} > \text{OPT}$$

contradicting the previous statement. This concludes the proof of the lemma. □

Proof of Claim 4.4.2. We prove parts 1-3 one by one.

Part 1. The upper bound $\mathbb{E}[f_e] \leq q_e$ is simple to prove. Consider random variable $f'_e = t_e/R$ and note that $f'_e \geq f_e$. We have

$$\mathbb{E}[f'_e] = \mathbb{E}\left[\frac{t_e}{R}\right] = \frac{1}{R}\mathbb{E}[t_e] = \frac{1}{R}\left(\sum_{i=1}^{R} \Pr[e \in \mathsf{MM}(\mathcal{G}_i)]\right) = \frac{1}{R}(R \times \Pr[e \in \mathsf{MM}(\mathcal{G}_1)]) = q_e.$$

Since $f_e \leq f'_e$, we get $\mathbb{E}[f_e] \leq \mathbb{E}[f'_e] = q_e$, concluding the proof of part 1.

Part 2. Next we turn to prove the lower bound $\mathbb{E}[f_e] \geq (1-\varepsilon)q_e$. Let X_i be the indicator random variable for $e \in \mathsf{MM}(\mathcal{G}_i)$. We have $t_e = X_1 + \ldots + X_R$, $\mathbb{E}[X_i] = q_e$, and $\mathbb{E}[t_e] = Rq_e$. Note also that the X_i's are independent since graphs $\mathcal{G}_1, \ldots, \mathcal{G}_R$ are drawn independently. Therefore, $\mathrm{Var}[t_e] = \sum_{i=1}^{R} \mathrm{Var}[X_i] = R(q_e - q_e^2)$.

Noting that $R = 0.5/\tau_-$ and that $q_e < \tau_-$ since e is non-crucial, we get $Rq_e < 1$. This means that if $t_e \geq a + 1$, then $|t_e - Rq_e| \geq a$; which implies $\Pr[t_e \geq a+1] \leq \Pr[|t_e - Rq_e| \geq a]$. Therefore by setting $a = \sqrt{R/\varepsilon}$ and also using Chebyshev's inequality, we get

$$\begin{aligned}\Pr\left[t_e \geq \sqrt{R/\varepsilon} + 1\right] &\leq \Pr\left[|t_e - \mathbb{E}[t_e]| \geq \sqrt{R/\varepsilon}\right] \\ &\leq \frac{\mathrm{Var}[t_e]}{(\sqrt{R/\varepsilon})^2} = \frac{R(q_e - q_e^2)}{(\sqrt{R/\varepsilon})^2} = \varepsilon(q_e - q_e^2) \leq \varepsilon q_e. \end{aligned} \tag{4.31}$$

Finally, we have

$$\mathbb{E}\left[\frac{t_e}{R}\right] = \underbrace{\Pr\left[\frac{t_e}{R} \leq \frac{1}{\sqrt{\varepsilon R}}\right]\mathbb{E}\left[\frac{t_e}{R} \mid \frac{t_e}{R} \leq \frac{1}{\sqrt{\varepsilon R}}\right]}_{=\mathbb{E}[f_e]} + \Pr\left[\frac{t_e}{R} > \frac{1}{\sqrt{\varepsilon R}}\right] \underbrace{\mathbb{E}\left[\frac{t_e}{R} \mid \frac{t_e}{R} > \frac{1}{\sqrt{\varepsilon R}}\right]}_{\leq 1 \text{ since by definition, } t_e \leq R.}$$

Rearranging the terms and replacing the bounds specified, we get

$$\begin{aligned}\mathbb{E}[f_e] &\geq \mathbb{E}\left[\frac{t_e}{R}\right] - \Pr\left[\frac{t_e}{R} > \frac{1}{\sqrt{\varepsilon R}}\right] \\ &= \frac{1}{R}\mathbb{E}[t_e] - \Pr\left[t_e \geq \sqrt{R/\varepsilon} + 1\right] \overset{(4.31)}{\geq} \frac{1}{R} \times Rq_e - \varepsilon q_e = (1-\varepsilon)q_e,\end{aligned}$$

concluding the proof of part 2.

Part 3. Note that $f_e \leq t_e/R$ by definition. Thus, we have $\sum_{e \ni v} f_e \leq \sum_{e \ni v} t_e/R = R^{-1}\sum_{e \ni v} t_e$. Since each $\mathsf{MM}(\mathcal{G}_i)$ includes at most one incident edge of v for being a matching, it holds that $\sum_{e \ni v} t_e \leq R$, thus indeed $\sum_{e \ni v} f_e \leq R^{-1} R = 1$.

Part 4. Let X_i be the event that v is matched in $\mathsf{MM}(\mathcal{G}_i)$ via a non-crucial edge and define $X := \sum_{i=1}^{R} X_i$. Furthermore, define for each edge e,

$$f'_e := \begin{cases} \frac{t_e}{R}, & \text{if } e \text{ is non-crucial,} \\ 0, & \text{otherwise.} \end{cases}$$

Note that f'_e is very similar to the value of f_e except for the case where $t_e/R > 1/\sqrt{\varepsilon R}$. In this case, $f_e = 0$ but f'_e remains to be the ratio t_e/R. This implies that $f'_e \geq f_e$. Now let $f'_v = \sum_{e \ni v} f'_e$. Since $f_e \leq f'_e$ for all edges, we have $f_v \leq f'_v$. Therefore, instead of proving $\Pr[f_v > n_v + 0.1\varepsilon] \leq (\varepsilon p)^{10}$, it suffices to prove $\Pr[f'_v > n_v + 0.1\varepsilon] \leq (\varepsilon p)^{10}$.

It holds from the definition that

$$f'_v = \sum_{e: e \in N, v \in e} \frac{t_e}{R} = \frac{1}{R} \sum_{e: e \in N, v \in e} t_e = \frac{1}{R} \times (X_1 + \ldots + X_R) = X/R.$$

Replacing this into $\Pr[f'_v > n_v + 0.1\varepsilon] \leq (\varepsilon p)^{10}$, we thus have to prove

$$\Pr[X/R > n_v + 0.1\varepsilon] \leq (\varepsilon p)^{10},$$

or equivalently:

$$\Pr[X > R n_v + 0.1 R\varepsilon] \leq (\varepsilon p)^{10}.$$

To prove this we use a concentration bound on X. Note that the X_i's are independent since graphs $\mathcal{G}_1, \ldots, \mathcal{G}_R$ are drawn independently. Moreover, for each $i \in [R]$, we have

$\mathbb{E}[X_i] = n_v$ since recall $X_i = 1$ iff v is matched via a non-crucial edge in $\mathsf{MM}(\mathcal{G}_i)$ and this has probability $\sum_{e:e\in N, v\in e} q_e = n_v$. Thus $\mathbb{E}[X] = Rn_v$. While we can use Chernoff's bound here since all X_i's are independent, even the second-moment method is enough for our desired inequality. The variance of X can be bounded as follows:

$$\mathrm{Var}[X] = \sum_{i=1}^{R} \mathrm{Var}[X_i] = \sum_{i=1}^{R} E[X_i^2] - \mathbb{E}[X_i]^2 = R(n_v - n_v^2).$$

By Chebyshev's inequality, we get

$$\Pr[X > Rn_v + 0.1R\varepsilon] \leq \frac{R(n_v - n_v^2)}{(0.1R\varepsilon)^2} = \frac{100(n_v - n_v^2)}{R\varepsilon^2} \leq \frac{100}{R\varepsilon^2}.$$

Since $R = 1/2\tau_-$ and $\tau_- < (\varepsilon p)^{50}$ by Corrolary 4.3.6, we get

$$\Pr[X > Rn_v + R\varepsilon] \leq \frac{100}{R\varepsilon^2} < \frac{200(\varepsilon p)^{50}}{\varepsilon^2} < (\varepsilon p)^{10},$$

which as described above concludes the proof. □

Proof of Observation 4.5.2. First note that realizations $\mathcal{C}_1, \ldots, \mathcal{C}_\alpha$ are all drawn precisely from the same distribution that realization $\mathcal{C} = \mathcal{C}_0$ is drawn from. Thus due to symmetry, matchings $M_0, \ldots, M_\alpha$ are all derived from the same distribution. Matchings $M'_0, \ldots, M'_\alpha$ are then the result of applying the augmenting-hyperwalks I found by $\mathsf{ApproximateMIS}(H, \varepsilon)$ on graph H. Construction of graph H is symmetrical w.r.t. matchings $M_0, \ldots, M_\alpha$. The only remaining component of the algorithm where this symmetry may break is in algorithm $\mathsf{ApproximateMIS}(H, \varepsilon)$ that may be biased towards picking augmenting-hyperwalks depending on which matching M_i they would augment. This can be avoided by using an algorithm for $\mathsf{ApproximateMIS}(H, \varepsilon)$ that is oblivious to the indices of matchings $M_0, \ldots, M_\alpha$ used to construct graph H. That is, suppose e.g. that we pick the ID of nodes in H randomly before feeding it into $\mathsf{ApproximateMIS}(H, \varepsilon)$. This guarantees that the obtained matchings $M'_0, \ldots, M'_\alpha$ will all have the same distribution due to their symmetry. □

Chapter 5: Stochastic Weighted Matching

In this chapter, we study the *stochastic weighted matching* problem, the weighted version of the stochastic matching problem discussed in the previous chapter. The problem is defined as follows: An arbitrary n-vertex graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}_{\geq 0}$ is given. A random subgraph $\mathcal{G}$ of G, called the *realization*, is then drawn by retaining each edge $e \in E$ independently with some fixed probability $p \in (0, 1]$. The goal is to choose a subgraph Q of G without knowing the realization $\mathcal{G}$ such that:

1. The maximum weight matching (MWM) among the *realized* edges of Q (i.e. graph $Q \cap \mathcal{G}$) approximates in expectation the MWM of the whole realization $\mathcal{G}$. Formally, we want the "approximation factor" $\mathbb{E}[\mu(Q \cap \mathcal{G})]/\mathbb{E}[\mu(\mathcal{G})]$ to be large where $\mu(\cdot)$ denotes the MWM's weight.

2. The subgraph Q has maximum degree $O(1)$. The constant here can (and in fact must) depend on p, but cannot depend on the structure of G such as the number of nodes or edge-weights.

Observe that by setting $Q = G$ we get an optimal solution, but the second constraint would be violated as the maximum degree in G could be very large. On the other hand, if we choose Q to be a single maximum weight matching of G, the maximum degree in Q would desirably be only one, but it is not possible to guarantee anything better than a p-approximation for this algorithm[1]. The stochastic matching problem therefore essentially asks whether it is possible to interpolate between these two extremes and pick a subgraph that is both sparse and provides a good approximation.

Applications. As its most straightforward application, the stochastic matching problem can be used as a *matching sparsifier* that approximately preserves the maximum (weight) matching under random edge failures [4]. It also has various applications in e.g. kidney exchange (see [18] for an extensive discussion) and online labor markets [11, 12]. For these applications, one is only given the base graph G but is tasked to find a matching in the realized subgraph $\mathcal{G}$. To do so, an algorithm can *query* each edge of G to see whether it is realized. Each of these queries typically maps to a time-consuming operation such as interviewing a candidate and thus few queries must be conducted. To do so, one can (non-adaptively) query only the $O(n)$ edges of Q and still expect to find an approximate MWM in the whole realization $\mathcal{G}$ which note may have $\Omega(n^2)$ edges.

Known bounds. Both the weighted and unweighted variants of this problem have been studied extensively [4, 5, 6, 9, 10, 11, 12, 17, 41] since the pioneering work of Blum *et al.* [17]. As discussed in the previous chapter, after a series of works on the unweighted version of the problem, we show that the approximation factor can be made $(1-\varepsilon)$ for any constant $\varepsilon > 0$ [10]. However, all these works rely heavily on the underlying graph being unweighted.

[1]To see this, let G be a clique with unit weights. It is easy to prove that a realization of G has a near-perfect matching with high probability, whereas only p fraction of the edges in the matching that forms Q are realized.

For the weighted case, in contrast, all known results remain close to a half approximation. The first result of this kind was proved by [41] who showed that by allowing Q's maximum degree to depend on the maximum weight W, one can obtain a 0.5-approximation. It was later proved in [12] through a different analysis of the same construction that dependence on W is not necessary to achieve a 0.5-approximation. Subsequently, the approximation factor was slightly improved to 0.501 using a different construction [11].

Our contribution. The main result of this chapter is as follows:

Theorem 6. *For any weighted graph G, any $p \in (0, 1]$, and any $\varepsilon > 0$, there is a subgraph Q of G with maximum degree $O_{\varepsilon,p}(1)$ that achieves a $(1-\varepsilon)$-approximation for the stochastic weighted matching problem.*

Not only Theorem 6 is the first result showing that a significantly better than 0.5-approximation is achievable for weighted graphs, but it essentially settles the approximation ratio. The remark below shows also that the dependence of the maximum degree of Q on both ε and p is necessary:

Remark 5.0.1. *For any $\varepsilon > 0$, any $(1 - \varepsilon)$-approximate subgraph Q must have maximum degree $\Omega(\frac{\log 1/\varepsilon}{p})$ even when G is a unit-weight clique [5]. This also implies that the approximation ratio cannot be made $(1 - o(1))$ unless Q has $\omega(1)$ degree.*

For simplicity of presentation, we do not calculate the precise dependence of the maximum degree of Q on ε and p. Though we remark that the $O_{\varepsilon,p}(1)$ term in Theorem 6 hides an exponential dependence on ε and p. We leave it as an open problem to determine whether a $\text{poly}(\frac{1}{\varepsilon p})$ degree subgraph can also achieve a $(1-\varepsilon)$-approximation.

5.1 Technical Overview and the Challenge with Weighted Graphs

In the literature of the stochastic matching problem, the subgraph Q typically has a very simple construction and much of the effort is concentrated on analyzing its approximation factor. A good starting point is the following Sampling algorithm proposed in [11]:[2] For some parameter $R = O_{\varepsilon,p}(1)$, draw R independent realizations $\mathcal{G}_1, \ldots, \mathcal{G}_R$ of G and let $Q \leftarrow \mathsf{MM}(\mathcal{G}_1) \cup \ldots \cup \mathsf{MM}(\mathcal{G}_R)$ where here $\mathsf{MM}(\cdot)$ returns a maximum weight matching. It is clear that the maximum degree of Q is $R = O_{\varepsilon,p}(1)$, but what approximation does it guarantee? Clearly $\mathbb{E}[\mu(\mathcal{G}_i)] = \mathbb{E}[\mu(\mathcal{G})]$ since each $\mathcal{G}_i$ is drawn from the same distribution as $\mathcal{G}$. However, observe that only p fraction of the edges of each matching $\mathsf{MM}(\mathcal{G}_i)$ in expectation appear in the actual realization $\mathcal{G}$. Hence, the challenge in the analysis is to show that the realized edges of these matchings can augment each other to construct a matching whose weight approximates $\text{OPT} := \mathbb{E}[\mu(\mathcal{G})]$.

Since the weighted stochastic matching problem is a generalization of the unweighted version, all the challenges that occur for the unweighted variant carry over to the weighted case. Of key importance, is the so called "Ruzsa-Szemerédi barrier" which was first observed by [5] toward achieving a $(1-\varepsilon)$-approximation. As discussed in the previous chapter, we break this barrier for unweighted graphs using a notion of

[2] As we will soon discuss, we do not analyze just the Sampling algorithm in this work, and combine it with a Greedy algorithm stated formally as Algorithm 1.

"vertex independent matchings". To overcome this challenge for weighted graphs, in this chapter we generalize this notion to weighted graphs. Below we will discuss two challenges specific to weighted graphs and how we overcome them.

Challenge 1: Low-probability/high-weight edges. The analysis of the Sampling algorithm for unweighted graphs typically relies on a partitioning of the edge-set E into "crucial" and "non-crucial" edges (Similar to what we had in Chapter 4). Define $q_e := \Pr[e \in \mathsf{MM}(\mathcal{G})]$ and let $\tau = \tau(\varepsilon, p) \ll p$ be a sufficiently small threshold; an edge e is called "crucial" if $q_e \geq \tau$ and "non-crucial" if $q_e < \tau$. Observe that if we draw say $R = \frac{\log 1/\varepsilon}{\tau} = O_{\varepsilon,p}(1)$ realizations in the Sampling algorithm, then nearly all crucial edges appear in at least one of $\mathsf{MM}(\mathcal{G}_1), \ldots, \mathsf{MM}(\mathcal{G}_R)$ and thus belong to Q. On the other hand, non-crucial edges can be used very much interchangeably, at least when the graph is unweighted.

For weighted graphs there is a third class of edges: Edges e with a small probability q_e of appearing in $\mathsf{MM}(\mathcal{G})$ but a relatively large weight w_e. On one hand, there could be a super-constant number of these edges connected to each vertex, so we cannot consider them crucial and add all of them to Q. On the other hand, even "ignoring" few edges of this type can significantly hurt the weight of the matching, so they cannot be regarded as non-crucial. This is precisely the reason that the analysis of [11] only guarantees a 0.501-approximation for weighted graphs but achieves up to 0.65-approximation for unweighted graphs. (See [11, Section 6] and in particular Figure 4 of [11].)

We handle low-probability/high-weight edges in a novel way. Particularly, we complement the Sampling algorithm (stated as Algorithm 2) with a Greedy algorithm (stated as Algorithm 1) which hand picks *some* of the low-probability high-weight edges and adds them to Q. Then in our analysis, any low-probability/high-weight edge that is picked by the Greedy algorithm is treated as if they are crucial, while the rest are regarded as non-crucial. Describing how the Greedy algorithm decides which low-probability/high-weight edges to pick requires a number of careful definitions which are out of the scope of this section. However, in a rough sense, it picks edges that would be "ignored" in the analysis if we regarded them as non-crucial.

Challenge 2: Lack of the "sparsification lemma" for weighted graphs. Let us for now suppose that graph G is unweighted. It is often useful to assume $\mathbb{E}[\mu(\mathcal{G})] = \Omega(n)$ as for instance even by losing an additive εn factor in the size of the matching (say because a certain event fails around each vertex with probability ε), we can still guarantee a multiplicative $(1 - O(\varepsilon))$-approximation. A "sparsification lemma" of Assadi *et al.* [5] which was also used in a crucial way in Chapter 4 guarantees that this assumption comes without loss of generality for unweighted graphs. This is achieved by modifying the graph and ensuring that each vertex is matched with a large probability.

For weighted graphs, in contrast, the probability with which a vertex is matched is not a useful indicator of the weight that it contributes to the matching. For this reason, no equivalent of the sparsification lemma exists for weighted graphs. For another evidence that the sparsification lemma is not useful for weighted graphs, observe that by adding zero-weight edges we can assume w.l.o.g. that G is a clique. Therefore, each vertex v already has a probability $1 - o(1)$ of being matched (but perhaps via a zero-weight edge) and thus the reduction of [5] does not help.

Due to lack of the sparsification lemma, it is not sufficient to simply bound the probability of a "bad event" around each vertex by say ε when the graph is weighted. Rather, it is important to analyze the actual expected loss to the weight conditioned on that this bad event occurs. For this reason, our analysis turns out to be much more involved than the unweighted case. This appears both in generalizing the vertex-independent lemma (Section 5.4) to the weighted case, and in various other places in the analysis (in particular Claims 5.3.12 and 5.3.18).

5.2 Basic Definitions and The Algorithm

5.2.1 General Notation

For any matching M, we use $w(M) := \sum_{e \in M} w_e$ to denote the weight of M; and use $v \in M$ for any vertex v to indicate that there is an edge incident to v that belongs to M. We use $\mu(H)$ to denote the weight of the maximum weight matching in graph H. For any two vertices u and v, we use $d_G(u, v)$ to denote the size of the shortest path between u and v in graph G (note that this is not their weighted distance). For any event A, we use $\mathbf{1}(A)$ as the indicator of the event, i.e. $\mathbf{1}(A) = 1$ if A occurs and $\mathbf{1}(A) = 0$ otherwise.

5.2.2 Basic Stochastic Matching Notation/Definitions

We use $\textsc{opt}$ to denote $\mathbb{E}[\mu(\mathcal{G})]$. Note that $\textsc{opt}$ is just a number, the expected weight of the maximum weight matching in the realization $\mathcal{G}$. With this notation, to prove Theorem 6, we should prove that $\mathbb{E}[\mu(\mathcal{Q})] \geq (1-\varepsilon)\textsc{opt}$, where $\mathcal{Q} := Q \cap \mathcal{G}$ is the realized subgraph of Q.

For any graph H, we use $\mathsf{MM}(H)$ to denote a maximum weight matching of H. In case H has multiple maximum weight matchings, $\mathsf{MM}(H)$ returns an arbitrary one. It would be useful to think of $\mathsf{MM}(\cdot)$ as a *deterministic* maximum weight matching algorithm that always returns the same matching for any specific input graph. Having this, for each edge e define

$$q_e := \Pr_{\mathcal{G}}[e \in \mathsf{MM}(\mathcal{G})] \qquad \text{and} \qquad \chi_e = w_e \cdot q_e. \tag{5.1}$$

Observe that χ_e is the expected weight that e contributes to matching $\mathsf{MM}(\mathcal{G})$. These definitions also naturally extend to subsets of edges $F \subseteq E$ for which we denote

$$q(F) := \sum_{e \in F} q_e, \qquad \text{and} \qquad \chi(F) := \sum_{e \in F} \chi_e.$$

Observation 5.2.1. $\chi(E) = \textsc{opt}$.

Proof. By definition $\textsc{opt} = \mathbb{E}[\mu(\mathcal{G})]$. The proof therefore follows since:

$$\begin{aligned}
\mathbb{E}[\mu(\mathcal{G})] = \mathbb{E}[w(\mathsf{MM}(\mathcal{G}))] = \mathbb{E}\left[\sum_{e \in \mathsf{MM}(\mathcal{G})} w_e\right] &= \mathbb{E}\left[\sum_{e \in E} \mathbf{1}(e \in \mathsf{MM}(\mathcal{G})) \cdot w_e\right] \\
&= \sum_{e \in E} \Pr[e \in \mathsf{MM}(\mathcal{G})] \cdot w_e \\
&= \sum_{e \in E} q_e \cdot w_e = \sum_{e \in E} \chi_e = \chi(E),
\end{aligned}$$

where the fourth equality follows simply from linearity of expectation. □

5.2.3 The Algorithm

In what follows we describe two different algorithms that each picks a subgraph of graph G. The final subgraph Q is the union of the two subgraphs picked by these algorithms.

To state the first algorithm, let us first define function $\lambda : \mathbb{R} \times [0,1] \to \mathbb{R}$ as:

$$\lambda(\Delta, \varepsilon) := \varepsilon^{-24}(\log \Delta)(\log\log \Delta)^C, \tag{5.2}$$

where $C \geq 1$ is a large enough absolute constants that we fix later. This perhaps strange-looking function is defined in this way so that it satisfies the various equations that we will need throughout the analysis. Having it, the first algorithm we use is as follows:

Algorithm 3. GreedyAlgorithm($G = (V, E), p, \varepsilon$)

$P \leftarrow \emptyset$.
while *true* **do**
    $\Delta \leftarrow \max\{1, \text{maximum degree in subgraph } P\}$. // So in the first iteration, $\Delta = 1$.
    $I_q \leftarrow \{(u,v) \in E \setminus P \mid q_e \geq p^2\varepsilon^{10} \cdot \Delta^{-\lambda(\Delta,\varepsilon)}\}$.
    $I_d \leftarrow \{(u,v) \in E \setminus P \mid d_P(u,v) < \lambda(\Delta, \varepsilon)\}$.
    $I \leftarrow I_d \cup I_q$.
    if $\chi(I) \geq \varepsilon\text{OPT}$ **then**
        $P \leftarrow P \cup I$.
    else
        return P.

From now on, when we use Δ we refer to the final value assigned to it during

Algorithm 1, which is equivalent to the maximum degree of P (unless P remains empty, which in that case $\Delta = 1$).

Remark 5.2.2. *Algorithm 1 uses the value of q_e which is not given a priori. Naively, it can be computed for all edges by enumerating all possible realizations of G in exponential time. However, it is not hard to see that we only need to check $q_e > \tau$ where τ is a constant dependent on ε and p, hence a simple Monte Carlo algorithm can also find all the edges in I_q in polynomial time with high probability.*

The second algorithm which was proposed first in [11] is very simple and natural: Draw multiple random realizations and pick a maximum weight matching of each; formally:

Algorithm 4. $\mathsf{SamplingAlgorithm}(G = (V, E), p, \varepsilon)$

1. $R \leftarrow \lceil p^{-2}\varepsilon^{-10}\Delta^{\lambda(\Delta,\varepsilon)} \rceil$.
2. **for** i *in* $1 \dots R$ **do**
3. Draw a realization $\mathcal{G}_i$ by retaining each edge $e \in E$ independently with probability p.
4. **return** $S := \mathsf{MM}(\mathcal{G}_1) \cup \dots \cup \mathsf{MM}(\mathcal{G}_R)$.

As mentioned earlier, the final subgraph Q is the union of the outputs of Algorithms 1 and 2. That is, $Q = S \cup P$. We first prove in this section that the algorithms terminate and the resulting subgraph Q has $O_{\varepsilon,p}(1)$ maximum degree. We then turn to analyze the approximation-factor in the forthcoming sections.

Lemma 5.2.3. *Algorithms 1 and 2 terminate and the subgraph Q has maximum degree $O_{\varepsilon,p}(1)$.*

Proof. Algorithm 1 has an unconditional while loop, but we argue that it will terminate within at most $1/\varepsilon$ iterations. To see this, consider the progress of $\chi(P)$ after each iteration. Since none of the edges in I are in P due to its definition, in every iteration that the condition $\chi(I) \geq \varepsilon\textsc{opt}$ of Line 7 holds, the value of $\chi(P)$ increases by at least $\varepsilon\textsc{opt}$. On the other hand, since $P \subseteq E$ and $\chi(E) = \textsc{opt}$ (Observation 5.2.1), we have $\chi(P) \leq \textsc{opt}$. Hence, after at most $1/\varepsilon$ iterations, the condition of Line 7 cannot continue to hold and the algorithm returns P. Algorithm 2 also clearly terminates as it simply runs a for loop finitely many times.

To bound the maximum degree of Q by $O_{\varepsilon,p}(1)$ we show that it suffices to bound the maximum degree Δ of P by $O_{\varepsilon,p}(1)$. To see this, first observe that if $\Delta = O_{\varepsilon,p}(1)$ then also $\lambda(\Delta, \varepsilon) = O_{\varepsilon,p}(1)$ by definition of λ. On the other hand, since S is simply the union of $R = O(p^{-2}\varepsilon^{-10}\Delta^{\lambda(\Delta,\varepsilon)})$ matchings, its maximum degree can also be bounded by $O(p^{-2}\varepsilon^{-10}\Delta^{\lambda(\Delta,\varepsilon)}) = O_{\varepsilon,p}(1)$. It thus only remains to prove $\Delta = O_{\varepsilon,p}(1)$.

To bound Δ, let Δ_i be the maximum degree of P by the end of iteration i of the while loop in Algorithm 1. We prove via induction that for any $i \leq 1/\varepsilon$ we have $\Delta_i = O_{\varepsilon,p}(1)$. This is sufficient for our purpose since we already showed above that the algorithm terminates within $1/\varepsilon$ iterations.

For the base case with $i = 0$ (i.e. before the start of the while loop) P is empty, hence indeed $\Delta_0 = O_{\varepsilon,p}(1)$. Now consider any iteration i. Take any vertex v and let $e = (u, v)$ be an edge that belongs to I at iteration i. By definition of I in Line 7, $e \in I_d \cup I_q$ so it remains to bound the maximum degree of I_d and I_q. If $e \in I_d$, there should be a path between u and v consisting of only the edges already in P that has length less than $\ell := \lambda(\Delta_{i-1}, \varepsilon)$. Since the maximum degree in P at this point is Δ_{i-1}, there are at most Δ_{i-1}^{ℓ} such paths ending at v. This is a simple upper bound on the number of edges in I_d connected to v at iteration i. On the other hand, if $e \in I_q$, then by definition $q_e \geq p^2\varepsilon^{10} \cdot \Delta_{i-1}^{-\ell}$. Combined with $\sum_{e \ni v} q_e \leq 1$, this means there are at most $p^{-2}\varepsilon^{-10} \cdot \Delta_{i-1}^{\ell}$ edges in I_q connected to v. Thus the degree of any vertex v increases by at most $\Delta_{i-1}^{\ell} + p^{-2}\varepsilon^{-10}\Delta_{i-1}^{\ell}$ and as a result:

$$\Delta_i \leq \Delta_{i-1} + \Delta_{i-1}^{\ell} + p^{-2}\varepsilon^{-10}\Delta_{i-1}^{\ell}.$$

By the induction hypothesis, $\Delta_{i-1} = O_{\varepsilon,p}(1)$ which also consequently implies $\ell = O_{\varepsilon,p}(1)$ since ℓ is a function of only Δ_{i-1} and ε. Therefore, $\Delta_i \leq O_{\varepsilon,p}(1)^{O_{\varepsilon,p}(1)} = O_{\varepsilon,p}(1)$. Observe that since $i \leq 1/\varepsilon$, this use of the asymptotic notation over the steps of the inductive argument does not lead to any undesirable blow-up and the final maximum degree is indeed $O_{\varepsilon,p}(1)$ as desired. □

5.3 The Analysis

In this section, we analyze the approximation factor of the construction of Q described in the previous section.

Analysis via fractional matchings. Recall that our goal is to show graph $\mathcal{Q} := Q \cap \mathcal{G}$ has a matching of weight $(1 - O(\varepsilon))$OPT in expectation. Since Q is constructed independently from the realization $\mathcal{G}$, one can think of $\mathcal{Q}$ as a subgraph of Q that includes each edge of Q independently with probability p. To show this subgraph $\mathcal{Q}$ has a matching of weight close to OPT, we follow the by now standard recipe [10, 11] of constructing a fractional matching $\mathbf{x}$ on $\mathcal{Q}$, such that:

$$x_v := \sum_{e \ni v} x_e \leq 1 \qquad \forall v \in V \tag{5.3}$$

$$x_e \geq 0 \qquad \forall e \in \mathcal{Q} \tag{5.4}$$

$$x(U) := \sum_{e=(u,v):u,v \in U} x_e \leq \frac{|U|-1}{2} \qquad \forall U \subseteq V \text{ such that } |U| \text{ is odd and } \leq 1/\varepsilon. \tag{5.5}$$

Here (5.3) and (5.4) are simply fractional matching constraints. The last set of constraints (5.5), known as "blossom" [25] constraints, are needed to ensure that our fractional matching $\mathbf{x}$ can be turned into an integral matching of weight at least $(1-\varepsilon)$ times that of $\mathbf{x}$. (See [40, Section 25.2] for more context on the matching polytope and blossom constraints. See also [11, Section 2.2] for a simple proof of this folklore lemma that blossom inequalities over subsets of size up to $1/\varepsilon$ are sufficient for a $(1-\varepsilon)$-approximation.) In addition to the constraints above, we want fractional matching $\mathbf{x}$ to have weight close to OPT so that we can argue $\mathcal{Q}$ has an integral matching of size $(1-O(\varepsilon))$OPT. Formally, our goal is to construct $\mathbf{x}$ such that in addition to constraints (5.3–5.5), it satisfies:

$$\mathbb{E}\left[\sum_{e \in \mathcal{Q}} x_e w_e\right] \geq (1 - O(\varepsilon))\text{OPT}. \tag{5.6}$$

If $\mathbf{x}$ satisfies all these constraints, then we have $\mathbb{E}[\mu(\mathcal{Q})] \geq (1-O(\varepsilon))$OPT, proving Theorem 6.

Observation 5.3.1. *To prove Theorem 6, it suffices to give a construction* $\mathbf{x} : \mathcal{Q} \to [0,1]$ *satisfying constraints (5.3-5.5) and (5.6).*

5.3.1 Toward Constructing x: A Partitioning of E

To construct fractional matching $\mathbf{x}$, we first partition the edge set E into $P \cup I' \cup N$, where P is simply the output of Algorithm 1, I' is the set of edges in set I defined in the last iteration of Algorithm 1 (for which the condition $\chi(I) \geq \varepsilon\textsc{opt}$ of Line 7 fails), and N is the rest of edges, i.e. $N = E - P - I'$. On all edges $e \in I'$ we simply set $x_e = 0$, i.e., we do not use them in the fractional matching $\mathbf{x}$. For other edges $e \notin I'$, we use different constructions for $\mathbf{x}$ depending on whether $e \in P$ or $e \in N$. We describe the construction of $\mathbf{x}$ on P in Section 5.3.2 and the construction on N in Section 5.3.3. Before that, let us state a number of simple observations regarding this partitioning.

Observation 5.3.2. $\chi(P) + \chi(N) \geq (1 - \varepsilon)\textsc{opt}$.

Proof. Recall that $\chi(E) = \textsc{opt}$ by Observation 5.2.1. Combined with $E = P \cup I' \cup N$, this implies $\chi(P) + \chi(N) + \chi(I') \geq \textsc{opt}$. To complete the proof, we argue that $\chi(I') \leq \varepsilon\textsc{opt}$. To see this, recall that I' is defined as the set I in the last iteration of Algorithm 1. In the last iteration, the condition $\chi(I) \geq \varepsilon\textsc{opt}$ of Line 7 in Algorithm 1 must fail (otherwise there would be another iteration), and thus $\chi(I') < \varepsilon\textsc{opt}$. □

Observation 5.3.3. *For any edge $e = (u, v) \in N$, $q_e < p^2\varepsilon^{10}\Delta^{-\lambda(\Delta,\varepsilon)}$ and $d_P(u, v) \geq \lambda(\Delta, \varepsilon)$.*

Proof. In the last iteration of Algorithm 1, all edges $e = (u, v)$ with $q_e \geq p^2\varepsilon^{10}\Delta^{-\lambda(\Delta,\varepsilon)}$ or $d_P(u, v) < \lambda(\Delta, \varepsilon)$ are either already in P or are added to $I = I'$; thus $e \notin N$ since $N = E - P - I'$. □

5.3.2 Construction of the Fractional Matching x on P

To describe the construction, let us first state a "vertex-independent matching lemma" which we will prove in Section 5.4.

Lemma 5.3.4. *Let $G' = (V', E', w')$ be an edge-weighted base graph with maximum degree Δ'. Let $\mathcal{G}'$ be a random subgraph of G' that includes each edge $e \in E'$ independently with some probability $p \in (0, 1]$. Let $\mathcal{A}(H)$ be any (possibly randomized) algorithm that given any subgraph H of G', returns a (not necessarily maximum weight) matching of H. For any $\varepsilon > 0$ there is a randomized algorithm $\mathcal{B}$ to construct a matching $Z = \mathcal{B}(\mathcal{G}')$ of $\mathcal{G}'$ such that*

1. *For any vertex v, $\Pr_{\mathcal{G}' \sim G', \mathcal{B}}[v \in Z] \leq \Pr_{\mathcal{G}' \sim G', \mathcal{A}}[v \in \mathcal{A}(\mathcal{G}')] + \varepsilon^3$.*
2. *$\mathbb{E}[w(Z)] \geq (1 - \varepsilon)\mathbb{E}[w(\mathcal{A}(\mathcal{G}'))]$.*
3. *For any vertex-subset $\{v_1, v_2, \ldots\} \subseteq V'$ such that for all i, j, $d_{G'}(v_i, v_j) \geq \lambda$ where $\lambda = O(\varepsilon^{-24} \log \Delta' \operatorname{poly}(\log \log \Delta'))$, events $\{v_1 \in Z\}, \{v_2 \in Z\}, \{v_3 \in Z\}, \ldots$ are all independent with respect to both the randomizations used in algorithm $\mathcal{B}$ and in drawing $\mathcal{G}'$.*

We use this lemma in the following way: The graph $G' = (V', E', w')$ of the lemma, is simply the subgraph P picked by Algorithm 1 and thus Δ' is simply the maximum degree of P which recall we denote by Δ. We let the random subgraph $\mathcal{G}'$ be the subset of edges in P that are realized, which we denote by $\mathcal{P}$. As discussed before, since P is chosen independently from how the edges are realized, conditioned on P each edge is still realized independently from the others, so the assumption that

$\mathcal{P}$ is a random subgraph of P with edges realized independently is valid. Finally, we define the algorithm $\mathcal{A}(H)$ of the lemma for any subgraph $H \subseteq P$ as follows:

Algorithm 5. $\mathcal{A}(H)$

1 $H' \leftarrow H$.
2 Add any edge $e \in E \setminus P$ independently with probability p to H'.
3 **return** $\mathsf{MM}(H') \cap H$.

Observe that with definition above, $\mathcal{A}(\mathcal{P})$ can be interpreted in the following useful way: The input subgraph $H = \mathcal{P}$ already includes each edge of P independently with probability p. Since initially $H' \leftarrow H$, and every edge $e \in E \setminus P$ is then added to H' independently with probability p, by the end of Line 2, H' will have the same distribution as the realization $\mathcal{G}$ of G. This means:

Observation 5.3.5. *The output of* $\mathcal{A}(\mathcal{P})$ *has the same distribution as* $\mathsf{MM}(\mathcal{G}) \cap P$.

Finally, once we obtain a matching using the algorithm above, we remove each edge from the matching independently with probability ε. Doing so, we only lose ε fraction of the weight of the matching in expectation, but we ensure that each vertex is matched with probability at most $1 - \varepsilon$ which will be useful later.

Let us for each vertex v define $q_v^P := \sum_{e: v \in e, e \in P} q_e$ to be the probability that v is matched in $\mathsf{MM}(\mathcal{G})$ via an edge in P. Using Lemma 5.3.4 as discussed above, we get:

Claim 5.3.6. *There is an algorithm* $\mathcal{B}$ *to construct a matching* Z *on the realized edges* $\mathcal{P}$ *of* P *s.t.:*

1. *For any vertex* v, $\Pr_{\mathcal{P},\mathcal{B}}[v \in Z] \leq \min\{q_v^P + \varepsilon^3, 1 - \varepsilon\}$.

2. $\mathbb{E}[w(Z)] \geq (1 - 2\varepsilon)\chi(P)$.

3. *For any vertex-subset* $\{v_1, v_2, \ldots\} \subseteq V$ *such that for all* i, j, $d_P(v_i, v_j) \geq \lambda(\Delta, \varepsilon)$, *events* $\{v_1 \in Z\}, \{v_2 \in Z\}, \{v_3 \in Z\}, \ldots$ *are all independent with respect to both the randomizations used in algorithm* $\mathcal{B}$ *and the randomization in drawing* $\mathcal{P}$.

4. *Matching* Z *is independent of the realization of edges in* $E \setminus P$.

Proof. For property 1, Lemma 5.3.4 guarantees

$$\Pr_{\mathcal{P},\mathcal{B}}[v \in Z] \leq \Pr_{\mathcal{P},\mathcal{A}}[v \in \mathcal{A}(\mathcal{P})] + \varepsilon^3.$$

Moreover,

$$\begin{aligned}\Pr_{\mathcal{P},\mathcal{A}}[v \in \mathcal{A}(\mathcal{P})] &= \sum_{e \ni v} \Pr[e \in \mathcal{A}(\mathcal{P})] \stackrel{\text{Obs 5.3.5}}{=} \sum_{e \ni v} \Pr[e \in \mathsf{MM}(\mathcal{G}) \cap P] \\ &= \sum_{e : v \in e, e \in P} \Pr[e \in \mathsf{MM}(\mathcal{G})] = q_v^P.\end{aligned}$$

Therefore, $\Pr[v \in Z] \leq q_v^P + \varepsilon^3$. On the other hand, since as discussed above, at the end we drop each edge from the matching independently with probability ε, $\Pr[v \in Z] \leq 1 - \varepsilon$. Combination of these two bounds proves property 1.

For property 2, Lemma 5.3.4 already guarantees that the reported matching has weight at least $(1-\varepsilon)\mathbb{E}[w(\mathcal{A}(\mathcal{P}))]$. Since on top of that we retain each edge of the final matching with probability $1-\varepsilon$, we lose another $(1-\varepsilon)$ factor and have $\mathbb{E}[w(Z)] \geq (1-2\varepsilon)\mathbb{E}[w(\mathcal{A}(\mathcal{P}))]$. To see why this is the claimed bound of property 2, observe that:

$$\mathbb{E}[w(\mathcal{A}(\mathcal{P}))] \overset{\text{Obs 5.3.5}}{=} \mathbb{E}[w(\mathsf{MM}(\mathcal{G}) \cap P)] = \sum_{e \in P} \Pr[e \in \mathsf{MM}(\mathcal{G})] w_e = \sum_{e \in P} \chi_e = \chi(P).$$

For property 3, it just suffices to make sure $\lambda(\Delta, \varepsilon) \geq \lambda$ where recall $\lambda(\Delta, \varepsilon)$ was defined in (5.2) whereas λ is defined in Lemma 5.3.4. By definition (5.2), we already have $\lambda(\Delta, \varepsilon) = \Omega(\lambda)$. On the other hand, in definition (5.2) of $\lambda(\Delta, \varepsilon)$ there is a constant C that we can tune. Picking this constant to be large enough, we can guarantee that $\lambda(\Delta, \varepsilon) \geq \lambda$ and satisfy this property.

Finally, property 4 holds since in construction of Z the algorithm is essentially unaware of the actual realization of edges in $E \setminus P$ and is thus independent of it. □

Once we construct matching Z on the realized edges of P using the algorithm above, for any edge $e \in P$ we set $x_e = 1$ if $e \in Z$ and $x_e = 0$ otherwise. Therefore, $\mathbf{x}$ is in fact integral on all edges of P. The properties of Z highlighted in Claim 5.3.6 will be later used in augmenting $\mathbf{x}$ via the realized edges among the edges in N.

5.3.3 Construction of the Fractional Matching x on N

We first formally describe construction of $\mathbf{x}$ on the edges in N, then discuss the main intuitions behind the construction, and finally prove that it satisfies the needed properties.

5.3.3.1 The Construction

We first define an "assignment" $\mathbf{f} : E \to [0,1]$, then based on $\mathbf{f}$ define an assignment $\mathbf{g} : E \to [0,1]$, then based on $\mathbf{g}$ define an assignment $\mathbf{h} : E \to [0,1]$, and finally construct $\mathbf{x}$ from $\mathbf{h}$. For any assignment $\mathbf{a} \in \{\mathbf{f}, \mathbf{g}, \mathbf{h}, \mathbf{x}\}$ we may use the following notation: For an edge e, a_e denotes the value of $\mathbf{a}$ on edge e. For a vertex v, $a_v := \sum_{e \ni v} a_e$ denotes the sum of assignments adjacent to v. The weight $w(\mathbf{a})$ denotes $\sum_{e \in E} a_e w_e$.

As outlined above, we first define $\mathbf{f} : E \to [0,1]$ on each edge e as follows:

$$f_e := \begin{cases} \frac{1}{R} \sum_{i=1}^{R} \mathbf{1}(e \in \mathsf{MM}(\mathcal{G}_i)) & \text{if } e \in N, \\ 0 & \text{otherwise,} \end{cases} \tag{5.7}$$

where recall that $\mathcal{G}_i$ is the ith drawn realization in Algorithm 2 and R is the total number of realizations drawn in Algorithm 2. In words, for any $e \in N$, the value of f_e denotes the fraction of matchings $\mathsf{MM}(\mathcal{G}_1), \ldots, \mathsf{MM}(\mathcal{G}_R)$ that include e.

Based on $\mathbf{f}$, we define $\mathbf{g}$ on each $e = (u, v)$ as:

$$g_e := \begin{cases} f_e & \text{if } f_e \leq p^2 \varepsilon^7 \Delta^{-\lambda(\Delta,\varepsilon)},\ f_u \leq 1 - q_u^P + \varepsilon^3, \text{ and } f_v \leq 1 - q_v^P + \varepsilon^3, \\ 0 & \text{otherwise.} \end{cases} \tag{5.8}$$

Next, based on $\mathbf{g}$, we define $\mathbf{h}$ on each edge $e = (u, v)$ as:

$$h_e := \begin{cases} \frac{g_e}{p \Pr[v \notin Z] \Pr[u \notin Z]} & \text{if } u \notin Z,\ v \notin Z, \text{ and } e \text{ is realized} \\ 0 & \text{otherwise.} \end{cases} \tag{5.9}$$

Here, as defined in the previous section, the value of q_v^P for a vertex v denotes the probability that v is matched in $\mathsf{MM}(\mathcal{G})$ via an edge in P.

We are finally ready to define the construction of $\mathbf{x}$ on N. On each edge $e = (u, v) \in N$, we set:

$$x_e \leftarrow \begin{cases} \frac{h_e}{1+3\varepsilon} & \text{if } h_v \leq 1 + 3\varepsilon \text{ and } h_u \leq 1 + 3\varepsilon, \\ 0 & \text{otherwise.} \end{cases} \tag{5.10}$$

5.3.3.2 Intuitions and Proof Outline

Here we discuss the main intuitions behind the construction above for $\mathbf{x}$ on N in a slightly informal way. The rigorous proof that the final fractional matching $\mathbf{x}$ satisfies properties (5.3-5.6) is given in the forthcoming sections.

As mentioned above, for every edge $e \in N$, f_e simply denotes the fraction of matchings $\mathsf{MM}(\mathcal{G}_1), \ldots, \mathsf{MM}(\mathcal{G}_R)$ that include e. Therefore $\mathbf{f}$ is a linear combination of these integral matchings, and thus is a valid fractional matching. Another key observation here is that since each $\mathcal{G}_i$ has the same distribution as $\mathcal{G}$, the probability of each edge e appearing in each matching $\mathsf{MM}(\mathcal{G}_i)$ is exactly equal to the probability q_e that it appears in $\mathsf{MM}(\mathcal{G})$. This can be used to prove $\mathbb{E}[f_e] = q_e$ (see Observation 5.3.8) which also implies $\mathbb{E}[w(\mathbf{f})] = \chi(N)$ (see Observation 5.3.9). Thus, fractional matching $\mathbf{f}$ has precisely the weight $\chi(N)$ we need $\mathbf{x}$ to have on N. In addition (unlike q_e) the value of f_e is only non-zero on edges $e \in N$ that also belong to the output S of Algorithm 2. This is desirable since recall that if an edge $e \in N$ does not belong to S, then $e \notin Q$ and as a result $e \notin \mathcal{Q}$. Thus, we should ensure $x_e = 0$ since we want $\mathbf{x}$ to be a fractional matching of subgraph $\mathcal{Q}$.

In the next step of the construction, we define $\mathbf{g}$ based on $\mathbf{f}$. The key idea behind this definition is to get rid of possible "deviations" in $\mathbf{f}$ and ensure that $\mathbf{g}$ satisfies certain deterministic inequalities for g_e on all edges e, and g_v for all vertices v. It turns out that by carefully bounding the probability of these deviations, we can still argue that $\mathbf{g}$ has weight close to $\chi(N)$ (see Claim 5.3.12) just like $\mathbf{f}$.

Despite the desirable properties mentioned above, $\mathbf{g}$ is still far from the values we would like to assign to edges N in $\mathbf{x}$, for the following two reasons. First, we want $\mathbf{x}$ to be non-zero only on $\mathcal{Q}$, i.e. the realized edges in Q. However, in defining $\mathbf{g}$ we never look at edge realizations. Hence, it could be that $g_e > 0$ for an edge e that is not realized. The second problem is that we need to augment the matching Z already constructed in Section 5.3.2. More specifically, recall from Section 5.3.2 that we have already assigned $x_e = 1$ to any edge $e \in Z$. Therefore, if we want $\mathbf{x}$ to be a valid fractional matching, all edges e that are incident to a matched vertex of Z should have $x_e = 0$. In defining $\mathbf{h}$, we address both issues at the same time. That is, for any edge e, if e is not realized or at least one of its endpoints is matched in Z, we set $h_e = 0$. Though note that we still want $\mathbb{E}[w(\mathbf{h})]$ to be close to $\mathbb{E}[w(\mathbf{g})]$ and $\chi(N)$. To compensate for the loss to the weight due to edges e for which $g_e > 0$ but $h_e = 0$, on each edge e that is eligible to be assigned $h_e > 0$, we multiply g_e by an appropriate amount that cancels out the probability of assigning $h_e = 0$. Doing so, we can ensure that $\mathbb{E}[w(\mathbf{h})]$ remains sufficiently close to $w(\mathbf{g})$ and thus $\chi(N)$ (Claim 5.3.16).

Finally, recall from above that $\mathbf{f}$ is a valid fractional matching and thus so is $\mathbf{g}$

since $g_e \leq f_e$ on all edges. A next challenge is to make sure that once we obtain $\mathbf{h}$ by multiplying $\mathbf{g}$ on some edges, we still have a valid fractional matching. That, e.g. $h_v \leq 1$ for all vertices v. Toward achieving this, we first show in Claim 5.3.19 that for each vertex v, the probability that $h_v > 1 + 3\varepsilon$ is very small. But these deviations do occur. Thus, in our final construction of $\mathbf{x}$, on any edge $e = (u, v)$ for which at least one of h_u and h_v exceeds $1+3\varepsilon$, we set $x_e = 0$ and set $x_e = h_e/(1+3\varepsilon)$ on the rest of the edges. This way, we guarantee that for any vertex v, $x_v \leq 1$. Moreover, due to the low probability of violations in $\mathbf{h}$, there is a small probability for any edge e to have $x_e = 0$ but $h_e > 0$. Therefore, $\mathbf{x}$ as defined, will have weight close to $\chi(N)$ in expectation on the edges in $\mathcal{Q} \cap N$ (Claim 5.3.18). Combined with the construction of $\mathbf{x}$ on the edges in P which guarantees a weight of $\approx \chi(P)$ there, we obtain that overall $\mathbf{x}$ will have weight close to $\chi(P) + \chi(N)$ which is $\approx$ OPT as guaranteed by Observation 5.3.2. Therefore, $\mathbf{x}$ can be shown to satisfy all the needed properties required by Observation 5.3.1 thereby proving Theorem 6 (see Section 5.3.4).

5.3.3.3 Properties of f and g.

We start with a few simple observations.

Observation 5.3.7. *For any $i \in [R]$ and any edge e,* $\Pr[e \in \mathsf{MM}(\mathcal{G}_i)] = q_e$.

Proof. Since each realization $\mathcal{G}_i$ in Algorithm 2 has the same distribution as $\mathcal{G}$, we have $\Pr[e \in \mathsf{MM}(\mathcal{G}_i)] = \Pr[e \in \mathsf{MM}(\mathcal{G})]$. The claim follows from the definition (5.1) that $\Pr[e \in \mathsf{MM}(\mathcal{G})] = q_e$. □

Observation 5.3.8. *For each edge $e \in N$,* $\mathbb{E}[f_e] = q_e$.

Proof. For any $e \in N$, it holds by definition (5.7) that

$$\mathbb{E}[f_e] = \frac{1}{R}\sum_{i=1}^{R} \Pr[e \in \mathsf{MM}(\mathcal{G}_i)] \overset{\text{Obs } 5.3.7}{=} \frac{1}{R}\sum_{i=1}^{R} q_e = q_e,$$

which is the desired bound. □

Observation 5.3.9. $\mathbb{E}[w(\mathbf{f})] = \chi(N)$.

Proof. We have $w(\mathbf{f}) = \sum_{e\in E} f_e w_e = \sum_{e \in N} f_e w_e$ since $f_e = 0$ for all $e \notin N$. Thus by linearity of expectation,

$$\mathbb{E}[w(\mathbf{f})] = \sum_{e\in N} \mathbb{E}[f_e] w_e = \sum_{e \in N} q_e w_e = \chi(N),$$

where the second equality holds by Observation 5.3.8. □

Observation 5.3.10. *For any edge e, $g_e \le p^2 \varepsilon^7 \Delta^{-\lambda(\Delta,\varepsilon)}$.*

Proof. By construction of $\mathbf{g}$, if g_e is non-zero, then $g_e = f_e$ and $f_e \le p^2\varepsilon^7\Delta^{-\lambda(\Delta,\varepsilon)}$. □

Observation 5.3.11. *For any vertex v, $g_v \le 1 - q_u^P + \varepsilon^3$.*

Proof. By construction of $\mathbf{g}$, if $g_v \neq 0$, then $f_v \le 1 - q_u^P + \varepsilon^3$, and thus so is g_v since $\mathbf{g} \le \mathbf{f}$. □

The main takeaway of this section is the following claim, which guarantees $\mathbb{E}[w(\mathbf{g})]$ is large enough for our purpose.

Claim 5.3.12. $\mathbb{E}[w(\mathbf{g})] \ge (1-\varepsilon)\chi(N)$.

The proof of Claim 5.3.12 is rather involved. The main difficulty is the lack of an equivalent of a sparsification lemma for weighted graphs (as discussed in Section 5.1). The rest of this section is devoted to proving Claim 5.3.12 for which we need a number of other auxiliary claims.

For simplicity, let us for each edge e use F_e as a shorthand for event $f_e \leq p^2\varepsilon^7\Delta^{-\lambda(\Delta,\varepsilon)}$ and for each vertex v use F_v as a shorthand for event $f_v \leq 1 - q_v^P + \varepsilon^3$. These are precisely the events used in definition (5.8) of $\mathbf{g}$. In particular, for any $e = (u, v) \in E$, $g_e = f_e$ if event $F_e \wedge F_u \wedge F_v$ holds.

Claim 5.3.13. *For any edge* $e \in N$*,*

$$\mathbb{E}[g_e] \geq q_e(1 - \Pr[\overline{F_e} \mid \mathcal{G}_1] - \Pr[\overline{F_u} \mid \mathcal{G}_1] - \Pr[\overline{F_v} \mid \mathcal{G}_1]),$$

where here as usual, $\overline{F_e}, \overline{F_v}$*, and* $\overline{F_u}$ *denote the complement of events* F_e*,* F_v*, and* F_u *respectively.*

Proof. We have

$$
\begin{aligned}
\mathbb{E}[g_e] &= \mathbb{E}[f_e \mid F_e \wedge F_u \wedge F_v] \\
&= \mathbb{E}\left[\frac{1}{R}\sum_{i=1}^{R} \mathbf{1}(e \in \mathsf{MM}(\mathcal{G}_i)) \,\middle|\, F_e \wedge F_u \wedge F_v\right] && \text{By definition (5.7).} \\
&= \frac{1}{R}\sum_{i=1}^{R} \Pr[e \in \mathsf{MM}(\mathcal{G}_i) \mid F_e \wedge F_u \wedge F_v] && \text{Linearity of expectation.} \\
&= \frac{1}{R}\sum_{i=1}^{R} \Pr[e \in \mathsf{MM}(\mathcal{G}_1) \mid F_e \wedge F_u \wedge F_v] && \text{By symmetry.} \\
&= \Pr[e \in \mathsf{MM}(\mathcal{G}_1) \mid F_e \wedge F_u \wedge F_v] \\
&= \Pr[e \in \mathsf{MM}(\mathcal{G}_1)] \cdot \frac{\Pr[F_e \wedge F_u \wedge F_v \mid \mathcal{G}_1]}{\Pr[F_e \wedge F_v \wedge F_u]} && \text{Bayes' rule.} \\
&\geq \Pr[e \in \mathsf{MM}(\mathcal{G}_1)] \cdot \Pr[F_e \wedge F_u \wedge F_v \mid \mathcal{G}_1] && \text{Since } \Pr[F_e \wedge F_v \wedge F_u] \leq 1. \\
&= q_e \Pr[F_e \wedge F_u \wedge F_v \mid \mathcal{G}_1] && \text{By Observation 5.3.7.} \\
&\geq q_e(1 - \Pr[\overline{F_e} \mid \mathcal{G}_1] - \Pr[\overline{F_u} \mid \mathcal{G}_1] - \Pr[\overline{F_v} \mid \mathcal{G}_1]). && \text{By union bound.}
\end{aligned}
$$

The last inequality matches the one stated in the claim and the proof is complete. □

Claim 5.3.14. *For any edge* $e \in N$*, it holds that* $\Pr[\overline{F_e} \mid \mathcal{G}_1] \leq 2\varepsilon^3$*.*

Proof. We have

$$
\begin{aligned}
\mathbb{E}[f_e \mid \mathcal{G}_1] &= \mathbb{E}\left[\frac{1}{R}\sum_{i=1}^{R} \mathbf{1}(e \in \mathsf{MM}(\mathcal{G}_i)) \,\middle|\, \mathcal{G}_1\right] \\
&\leq \frac{1}{R} + \frac{1}{R}\sum_{i=2}^{R} \Pr[e \in \mathsf{MM}(\mathcal{G}_i)] \overset{\text{Obs 5.3.7}}{\leq} \frac{1}{R} + q_e.
\end{aligned}
$$

We have $R \geq p^{-2}\varepsilon^{-10}\Delta^{\lambda(\Delta,\varepsilon)}$ by its definition in Algorithm 2 and also $q_e \leq p^2\varepsilon^{10}\Delta^{-\lambda(\Delta,\varepsilon)}$

by Observation 5.3.3. Hence, $\mathbb{E}[f_e \mid \mathcal{G}_1] < 2p^2\varepsilon^{10}\Delta^{-\lambda(\Delta,\varepsilon)}$. Applying Markov's inequality, we thus get

$$\Pr\big[f_e > p^2\varepsilon^7\Delta^{-\lambda(\Delta,\varepsilon)} \mid \mathcal{G}_1\big] = \Pr[\overline{F_e} \mid \mathcal{G}_1] \leq 2\varepsilon^3,$$

which is the desired bound. □

Claim 5.3.15. *For any vertex v, $\Pr[\overline{F_v} \mid \mathcal{G}_1] \leq 4\varepsilon^4$.*

Proof. Let us for any $i \in [R]$ define $X_i = 1$ if vertex v is matched in $\mathsf{MM}(\mathcal{G}_i)$ via an edge $e \in N$ and $X_i = 0$ otherwise. Also let $X := \sum_{i=2}^{R} X_i$ (note that the sum index starts from 2). We have:

$$\begin{aligned}
f_v &= \sum_{e \ni v} f_e = \sum_{e: v\in e, e\in N} f_e && \text{By (5.7), } f_e = 0 \text{ if } e \notin N.\\
&= \sum_{e: v\in e, e\in N} \left(\frac{1}{R}\sum_{i=1}^{R} \mathbf{1}(e \in \mathsf{MM}(\mathcal{G}_i))\right) = \frac{1}{R}\sum_{i=1}^{R}\sum_{e: v\in e, e\in N} \mathbf{1}(e \in \mathsf{MM}(\mathcal{G}_i))\\
&= \frac{1}{R}\sum_{i=1}^{R} X_i \leq \frac{1}{R} + \frac{1}{R}\sum_{i=2}^{R} X_i \leq \frac{X+1}{R}. && (5.11)
\end{aligned}$$

Furthermore,

$$\begin{aligned}
\Pr[\overline{F_v} \mid \mathcal{G}_1] &= \Pr[f_v > 1 - q_v^P + \varepsilon^3 \mid \mathcal{G}_1] && \text{Definition of } F_v.\\
&\leq \Pr\left[\frac{X+1}{R} > 1 - q_v^P + \varepsilon^3 \mid \mathcal{G}_1\right] && \text{By (5.11), } f_v \leq \frac{X+1}{R}.\\
&= \Pr\big[X > R(1 - q_v^P + \varepsilon^3) - 1 \mid \mathcal{G}_1\big]\\
&= \Pr\big[X > R(1 - q_v^P + \varepsilon^3) - 1\big], && (5.12)
\end{aligned}$$

where the last inequality follows from the fact that $X = \sum_{i=2}^{R} X_i$ depends only on realizations $\mathcal{G}_2, \dots, \mathcal{G}_R$ and is independent of realization $\mathcal{G}_1$.

Therefore to bound $\Pr[\overline{F_v} \mid \mathcal{G}_1]$ we should analyze the behavior of random variable X. Let us start with its expected value:

$$\begin{aligned}
\mathbb{E}[X] &= \sum_{i=2}^{R} \Pr[X_i = 1] = \sum_{i=2}^{R} \Pr[X_2 = 1] \qquad \text{As } \Pr[X_2 = 1] = \ldots = \Pr[X_R = 1]. \\
&= (R-1)\Pr[X_2 = 1] \le R\Pr[X_2 = 1] \\
&\le R(1 - q_v^P). \qquad (5.13)
\end{aligned}$$

The last inequality holds for the following reason: By definition $q_v^P = \sum_{e: v \in e, e \in P} q_e$; since each edge e belongs to $\mathsf{MM}(\mathcal{G}_2)$ with probability q_e by Observation 5.3.7, we get that with probability q_v^P, vertex v is matched in $\mathsf{MM}(\mathcal{G}_2)$ via an edge $e \in P$; in this case, event $X_2 = 1$ which requires v to be matched via an edge in N cannot hold since $N \cap P = \emptyset$; hence $\Pr[X_2 = 1] \le 1 - q_v^P$.

We also need a concentration bound on X which we prove via Chebyshev's inequality[3] using the independence of events $X_2, \dots, X_R$. For any $t \ge 0$ we have

$$\begin{aligned}
\Pr[X > \mathbb{E}[X] + t] &\le \frac{\mathrm{Var}[X]}{t^2} = \frac{\sum_{i=2}^{R} \mathrm{Var}[X_i]}{t^2} \\
&\le \frac{R\,\mathrm{Var}[X_2]}{t^2} = \frac{R(\mathbb{E}[X_2^2] - E[X_2]^2)}{t^2} \le \frac{R}{t^2}. \qquad (5.14)
\end{aligned}$$

[3]One can also attempt to get a stronger concentration bound via Chernoff-type bounds, but the second moment method suffices for our purpose here.

As a result,

$$\Pr[X > R(1-q_v^P+\varepsilon^3)-1] = \Pr[X > R(1-q_v^P)+(\varepsilon^3 R-1)] \overset{(5.13),(5.14)}{\leq} \frac{R}{(\varepsilon^3 R-1)^2} \leq 4\varepsilon^4, \tag{5.15}$$

where the last inequality follows from

$$\frac{R}{(\varepsilon^3 R-1)^2} \leq \frac{R}{(\varepsilon^3 R/2)^2} \leq \frac{4}{\varepsilon^6 R} \overset{R\geq p^{-2}\varepsilon^{-10}}{\leq} \frac{4\varepsilon^{10}p^2}{\varepsilon^6} \leq 4\varepsilon^4.$$

Replacing (5.15) into (5.12) gives the desired bound that $\Pr[\overline{F_v} \mid \mathcal{G}_1] \leq 4\varepsilon^4$. □

We finally have the tools needed to prove Claim 5.3.12.

Proof of Claim 5.3.12. We have

$$\mathbb{E}[w(\mathbf{g})] = \mathbb{E}\left[\sum_{e\in E} g_e w_e\right] \geq \mathbb{E}\left[\sum_{e\in N} g_e w_e\right] = \sum_{e\in N} \mathbb{E}[g_e] w_e. \tag{5.16}$$

Furthermore, by Claim 5.3.13, for any $e \in N$ we have

$$\mathbb{E}[g_e] \geq q_e(1 - \Pr[\overline{F_e} \mid \mathcal{G}_1] - \Pr[\overline{F_u} \mid \mathcal{G}_1] - \Pr[\overline{F_v} \mid \mathcal{G}_1]).$$

Incorporating the bounds of Claims 5.3.14 and 5.3.15, we get for any $e \in N$ that

$$\mathbb{E}[g_e] \geq q_e(1 - 2\varepsilon^3 - 4\varepsilon^4 - 4\varepsilon^4) > (1 - 10\varepsilon^3) q_e.$$

Therefore, from (5.16) we get

$$\mathbb{E}[w(\mathbf{g})] \geq \sum_{e \in N} (1 - 10\varepsilon^3) q_e w_e = (1 - 10\varepsilon^3) \sum_{e \in N} q_e w_e = (1 - 10\varepsilon^3)\chi(N) \geq (1 - \varepsilon)\chi(N),$$

concluding the proof. □

5.3.3.4 Properties of h, and x on N.

In this section we turn to prove a number of useful properties of $\mathbf{h}$. We emphasize that in the previous section all expectations and probabilities are taken only over the randomization inherent in Algorithm 2. In contrast, in this section, all the probabilistic statements are with regards to the randomization of realization $\mathcal{G}$, and the randomization used in drawing matching Z in Section 5.3.2.

Claim 5.3.16. $\mathbb{E}[w(\mathbf{h})] \geq w(\mathbf{g})$.

Proof. Take any edge $e = (u, v) \in N$. By definition of $\mathbf{h}$ we have $h_e = \frac{g_e}{p \Pr[v \notin Z] \Pr[u \notin Z]}$ if e is realized and both u and v are unmatched in Z, and $h_e = 0$ otherwise. Since $d_P(u, v) \geq \lambda(\Delta, \varepsilon)$ by Observation 5.3.3, the condition of Claim 5.3.6 part 3 is satisfied and events $u \in Z$ and $v \in Z$ are independent. Moreover, since $e \notin P$, its realization is also independent of Z by Claim 5.3.6 property 4. Hence,

$$\mathbb{E}[h_e] = \Pr[e \text{ realized}] \Pr[v \notin Z] \Pr[u \notin Z] \frac{g_e}{p \Pr[v \notin Z] \Pr[u \notin Z]} = g_e.$$

This means that

$$\mathbb{E}[w(\mathbf{h})] = \sum_{e \in N} \mathbb{E}[h_e] w_e = \sum_{e \in N} g_e w_e = w(\mathbf{g}),$$

completing the proof. □

Observation 5.3.17. *For any edge e, $h_e \leq \frac{g_e}{p\varepsilon^2} \leq p\varepsilon^5 \Delta^{-\lambda(\Delta,\varepsilon)}$.*

Proof. By construction of $\mathbf{h}$ for any $e = (u, v)$ we have

$$h_e \leq \frac{g_e}{p \Pr[v \notin Z] \Pr[u \notin Z]} \stackrel{\star}{\leq} \frac{g_e}{p\varepsilon^2} \stackrel{\text{Observation 5.3.10}}{\leq} \frac{p^2 \varepsilon^7 \Delta^{-\lambda(\Delta,\varepsilon)}}{p\varepsilon^2} = p\varepsilon^5 \Delta^{-\lambda(\Delta,\varepsilon)},$$

where the inequality marked by $\star$ follows from the fact that $\Pr[v \in Z] \leq 1 - \varepsilon$ by property 1 of Claim 5.3.6 and thus $\Pr[v \notin Z] \geq \varepsilon$ and similarly $\Pr[u \notin Z] \geq \varepsilon$. □

Claim 5.3.18 below is one of the key components towards achieving our main result in Theorem 6. We present the proof in multiple steps, by proving a number of properties of $\mathbf{h}$.

Claim 5.3.18. *It holds that $\mathbb{E}[\sum_{e \in N} x_e w_e] \geq (1 - 15\varepsilon) w(\mathbf{g})$.*

Proof. We already know from Claim 5.3.16 that $\mathbb{E}[w(\mathbf{h})] \geq w(\mathbf{g})$. Thus, if we show $\mathbb{E}[\sum_{e \in N} x_e w_e] \geq (1 - 3\varepsilon)\mathbb{E}[w(\mathbf{h})]$ we are done. For brevity, for any edge $e = (u, v)$ we use H_e to indicate the event ($u \notin Z, v \notin Z, e$ realized). Also we use X_e to indicate event ($h_v \leq 1 + 3\varepsilon$ and $h_u \leq 1 + \varepsilon$). Observe that H_e is the event used in construction (5.9) of h_e and X_e is the event used in construction (5.10) of $\mathbf{x}$ on N. Putting together

(5.9) and (5.10), for any $e = (u, v) \in N$, we have

$$x_e = \begin{cases} \frac{1}{1+3\varepsilon} \cdot \frac{g_e}{p \Pr[u \notin Z] \Pr[v \notin Z]} & H_e \wedge X_e, \\ 0 & \text{otherwise.} \end{cases}$$

This means that

$$\begin{aligned}
\mathbb{E}\left[\sum_{e \in N} x_e w_e\right] &= \sum_{e \in N} \mathbb{E}[x_e] w_e \\
&= \sum_{e \in N} \Pr[H_e \wedge X_e] \frac{1}{1+3\varepsilon} \cdot \frac{g_e}{p \Pr[u \notin Z] \Pr[v \notin Z]} w_e \\
&= \frac{1}{1+3\varepsilon} \sum_{e \in N} \Pr[X_e \mid H_e] \Pr[H_e] \frac{g_e}{p \Pr[u \notin Z] \Pr[v \notin Z]} w_e \\
&= \frac{1}{1+3\varepsilon} \sum_{e \in N} \Pr[X_e \mid H_e] \mathbb{E}[h_e] w_e \\
&= \frac{1}{1+3\varepsilon} \sum_{e=(u,v) \in N} \Pr[h_v \leq 1+3\varepsilon \wedge h_u \leq 1+3\varepsilon \mid H_e] \mathbb{E}[h_e] w_e \\
&= \frac{1}{1+3\varepsilon} \sum_{e=(u,v) \in N} (1 - \Pr[h_v > 1+3\varepsilon \mid H_e] - \Pr[h_u > 1+3\varepsilon \mid H_e]) \mathbb{E}[h_e] w_e.
\end{aligned}$$

Therefore it only remains to bound $\Pr[h_v > 1 + 3\varepsilon \mid H_e]$. The following claim, whose proof we present after the proof of the current Claim 5.3.18, gives us the desired bound for it.

Claim 5.3.19. *Let edge $e = (u, v) \in N$ be the one fixed above, then*

$$\Pr_{\mathcal{G},Z}[h_v > 1 + 3\varepsilon \mid F_e] \leq 6\varepsilon.$$

Plugging Claim 5.3.19 this into the equation above, we thus get

$$\mathbb{E}\left[\sum_{e\in N} x_e w_e\right] \geq \frac{1-12\varepsilon}{1+3\varepsilon}\sum_{e\in N}\mathbb{E}[h_e]w_e = \frac{1-12\varepsilon}{1+3\varepsilon}\mathbb{E}[w(\mathbf{h})]$$
$$> (1-15\varepsilon)\mathbb{E}[w(\mathbf{h})] \overset{\text{Claim 5.3.16}}{\geq} (1-15\varepsilon)w(\mathbf{g}),$$

which is our desired bound. □

For the rest of this section, we fix $e=(u,v)\in N$ and focus on proving Claim 5.3.19. To do so, we first bound the expected value of h_v conditioned on H_e in Claim 5.3.20 and then finish the proof via a concentration bound.

Note from constructions (5.7), (5.8), and (5.9) of respectively $\mathbf{f}$, $\mathbf{g}$, and $\mathbf{h}$, that $h_{e'} = g_{e'} = f_{e'} = 0$ for any $e' \notin N$. Hence, we have $h_v = \sum_{e'\ni v} h_{e'} = \sum_{e':e'\in N, v\in e'} h_{e'}$. Now let $e_1=(v,u_1), e_2=(v,u_2),\ldots,e_k=(v,u_k)$ be all edges connected to vertex v that belong to N and assume that $e_1=e=(v,u)$. We thus have

$$h_v = \sum_{i=1}^{k} h_{e_i}. \tag{5.17}$$

Claim 5.3.20. *Let edge $e=(u,v)\in N$ be the one fixed above, then* $\mathbb{E}[h_v \mid H_e] \leq 1+2\varepsilon$.

Proof. We have

$$\mathbb{E}[h_v \mid H_e] = \mathbb{E}\left[\sum_{i=1}^{k} h_{e_i} \,\middle|\, H_e\right] = \sum_{i=1}^{k}\mathbb{E}[h_{e_i} \mid H_e]. \tag{5.18}$$

To bound this, consider the following partitioning of $\{e_1,\ldots,e_k\}$ into two subsets A

and B:

$$A = \{e_i \mid d_P(u_i, u) < \lambda(\Delta, \varepsilon)\}, \qquad B = \{e_i \mid d_P(u_i, u) \geq \lambda(\Delta, \varepsilon)\}.$$

In particular, observe that $e_1 \in A$ since $u_1 = u$ which implies $d_P(u_1, u) = 0$. Separating A and B in the sum of (5.18) we get

$$\mathbb{E}[h_v \mid H_e] = \sum_{e_i \in A} \mathbb{E}[h_{e_i} \mid H_e] + \sum_{e_i \in B} \mathbb{E}[h_{e_i} \mid H_e]. \tag{5.19}$$

We bound the two sums over A and B in the inequality above separately.

Bounding the sum over A. For each $h_{e_i} \in A$, we use the pessimistic upper bound of Observation 5.3.17 for h_{e_i}. But instead we bound the size of A by

$$|A| \leq \Delta^{\lambda(\Delta,\varepsilon)} + 1 \leq 2\Delta^{\lambda(\Delta,\varepsilon)}. \tag{5.20}$$

This first inequality follows from the fact that the maximum degree in P is bounded by Δ, and hence there are at most $\Delta^{\lambda(\Delta,\varepsilon)}$ nodes (other than u itself) that have distance less than $\lambda(\Delta, \varepsilon)$ to u in graph P. The second inequality simply follows from the fact that both Δ and $\lambda(\Delta, \varepsilon)$ are ≥ 1 (see Algorithm 1). We thus have

$$\begin{aligned} \sum_{e_i \in A} h_{e_i} &\leq p\varepsilon^5 \Delta^{-\lambda(\Delta,\varepsilon)} |A| && \text{By Observation 5.3.17} \\ &\leq 2p\varepsilon^5. && \text{By (5.20).} \end{aligned} \tag{5.21}$$

Bounding the sum over B. Recall that $H_e = (e \text{ realized}, v \notin Z, u \notin Z)$ and $H_{e_i} = (e_i \text{ realized}, v \notin Z, u_i \notin Z)$. Therefore for any edge $e_i \in B$, we have

$$\begin{aligned}
\Pr[H_{e_i} \mid H_e] &= \Pr[e_i \text{ realized}, v \notin Z, u_i \notin Z \mid e \text{ realized}, v \notin Z, u \notin Z] \\
&= \Pr[e_i \text{ realized}, u_i \notin Z \mid e \text{ realized}, v \notin Z, u \notin Z] \\
&= p \Pr[u_i \notin Z \mid e \text{ realized}, v \notin Z, u \notin Z] \\
&= p \Pr[u_i \notin Z \mid v \notin Z, u \notin Z],
\end{aligned}$$

where the last two equalities follow from property 4 of Claim 5.3.6 regarding independence of matching Z from realization of edges in N (such as e_i and e), and noting that $e_i \neq e$ since $e_i \in B$. On the other hand, since $d_P(u_i, u) \geq \lambda(\Delta, \varepsilon)$ based on definition of B, and $d_P(u_i, v) \geq \lambda(\Delta, \varepsilon)$ by Observation 5.3.3, we get that event $u_i \in Z$ is independent of $v \in Z, u \in Z$ due to property 3 of Claim 5.3.6. Therefore $\Pr[u_i \notin Z \mid v \notin Z, u \notin Z] = \Pr[u_i \notin Z]$ and thus

$$\Pr[H_{e_i} \mid H_e] = p \Pr[u_i \notin Z] \qquad \text{for any } e_i \in B. \tag{5.22}$$

We can therefore bound the sum in (5.19) over B as follows:

$$\begin{aligned}
\sum_{e_i \in B} \mathbb{E}[h_{e_i} \mid H_e] &= \sum_{e_i \in B} \frac{g_{e_i} \Pr[H_{e_i} \mid H_e]}{p \Pr[v \notin Z] \Pr[u_i \notin Z]} \\
&= \sum_{e_i \in B} \frac{g_{e_i}}{\Pr[v \notin Z]} && \text{By (5.22).} \\
&\leq \frac{g_v}{\Pr[v \notin Z]} \\
&\leq \frac{1 - q_v^P + \varepsilon^3}{\Pr[v \notin Z]} && \text{Observation 5.3.11.} \\
&\leq \frac{1 - q_v^P + \varepsilon^3}{1 - \min\{q_v^P + \varepsilon^3, 1 - \varepsilon\}}. && \text{Since } \Pr[v \in Z] \leq \min\{q_v^P + \varepsilon^3, 1 - \varepsilon\} \text{ by Claim 5.3.6.}
\end{aligned}$$

Since both the nominator and the denominator are $\approx 1 - q_v^P$, the sum is upper bounded by ≈ 1. To formalize this, consider two scenarios: (i) $q_v^P - \varepsilon^3 \geq 1 - \varepsilon$, and (ii) $q_v^P - \varepsilon^3 < 1 - \varepsilon$. In the former, we have

$$\frac{1 - q_v^P + \varepsilon^3}{1 - \min\{q_v^P + \varepsilon^3, 1 - \varepsilon\}} \overset{(i)}{=} \frac{1 - q_v^P + \varepsilon^3}{1 - (1 - \varepsilon)} \overset{(i)}{\leq} \frac{1 - (1 - \varepsilon + \varepsilon^3) + \varepsilon^3}{\varepsilon} = \frac{\varepsilon}{\varepsilon} = 1.$$

In the latter case,

$$\begin{aligned}
\frac{1 - q_v^P + \varepsilon^3}{1 - \min\{q_v^P + \varepsilon^3, 1 - \varepsilon\}} &\overset{(ii)}{=} \frac{1 - q_v^P + \varepsilon^3}{1 - q_v^P - \varepsilon^3} \leq \frac{1 - (1 - \varepsilon + \varepsilon^3) + \varepsilon^3}{1 - (1 - \varepsilon + \varepsilon^3) - \varepsilon^3} \\
&= \frac{\varepsilon}{\varepsilon(1 - 2\varepsilon^2)} \leq 1 + \varepsilon,
\end{aligned}$$

where the last inequality holds for any $\varepsilon < 0.36$. Therefore overall, we get

$$\sum_{e_i \in B} \mathbb{E}[h_{e_i} \mid H_e] \leq 1 + \varepsilon. \tag{5.23}$$

Incorporating the bounds (5.21) and (5.23) into (5.19) we get that $\mathbb{E}[h_v \mid H_e] \leq 1 + \varepsilon + p\varepsilon^5 \leq 1 + 2\varepsilon$. $\square$

We are now ready to prove Claim 5.3.19 via a concentration bound.

Proof of Claim 5.3.19. By Chebyshev's inequality, and the bound $\mathbb{E}[h_v \mid H_e] \leq 1 + 2\varepsilon$ of Claim 5.3.20, we get that

$$\Pr_{\mathcal{G},Z}[h_v > (1 + 2\varepsilon) + \varepsilon \mid H_e] \leq \frac{\mathrm{Var}_{\mathcal{G},Z}[h_v \mid H_e]}{\varepsilon^2}. \tag{5.24}$$

For brevity, we do not write the subscript $\mathcal{G}, Z$ for our probabilistic statements for the rest of the proof when it is clear. Since $h_v = \sum_{i=1}^{k} h_{e_i}$, by definition of variance we have

$$\mathrm{Var}[h_v \mid H_e] = \sum_{i=1}^{k} \sum_{j=1}^{k} \mathrm{Cov}[h_{e_i}, h_{e_j} \mid H_e].$$

By definition, if h_{e_i} and h_{e_j} are independent with respect to the randomization of $\mathcal{G}$ and Z, and conditioned on H_e, then $\mathrm{Cov}_{\mathcal{G},Z}[h_{e_i}, h_{e_j} \mid H_e] = 0$. But this does not hold for all h_{e_i} and h_{e_j}. As in the proof of Claim 5.3.20 consider the following partitioning of $\{e_1, \ldots, e_k\}$:

$$A = \{e_i \mid d_P(u_i, u) < \lambda(\Delta, \varepsilon)\}, \qquad B = \{e_i \mid d_P(u_i, u) \geq \lambda(\Delta, \varepsilon)\}.$$

With this partitioning, we can rewrite the equation above for variance as:

$$\begin{aligned}
\mathrm{Var}[h_v \mid H_e] &= \sum_{e_i \in A} \sum_{j=1}^{k} \mathrm{Cov}[h_{e_i} h_{e_j} \mid H_e] + \sum_{e_i \in B} \sum_{e_j \in A} \mathrm{Cov}[h_{e_i} h_{e_j} \mid H_e] \\
&+ \sum_{e_i \in B} \sum_{e_j \in B} \mathrm{Cov}[h_{e_i} h_{e_j} \mid H_e] \\
&\leq 2 \sum_{e_i \in A} \sum_{j=1}^{k} |\, \mathrm{Cov}[h_{e_i} h_{e_j} \mid H_e]| + \sum_{e_i \in B} \sum_{e_j \in B} \mathrm{Cov}[h_{e_i} h_{e_j} \mid H_e]. \qquad (5.25)
\end{aligned}$$

We will bound the two sums over A differently. Before that, let us prove a simple upper bound on the covariance of any two edges e_i, e_j:

$$\begin{aligned}
\mathrm{Cov}[h_{e_i} h_{e_j} \mid H_e] &= \mathbb{E}_{\mathcal{G},Z}[h_{e_i} h_{e_j} \mid H_e] - \mathbb{E}_{\mathcal{G},Z}[h_{e_i} \mid H_e]\mathbb{E}[h_{e_j} \mid H_e] \\
&\leq \mathbb{E}_{\mathcal{G},Z}[h_{e_i} h_{e_j} \mid H_e] \\
&\leq \frac{g_{e_i}}{p\varepsilon^2} \cdot \frac{g_{e_j}}{p\varepsilon^2} \qquad \text{By (5.3.17)}. \qquad (5.26)
\end{aligned}$$

Bounding the sums over A. We have

$$
\begin{aligned}
2\sum_{e_i\in A}\sum_{j=1}^{k}|\operatorname{Cov}[h_{e_i}h_{e_j}\mid H_e]| &\le 2\sum_{e_i\in A}\sum_{j=1}^{k}\frac{g_{e_i}}{p\varepsilon^2}\cdot\frac{g_{e_j}}{p\varepsilon^2} && \text{By (5.26).}\\
&\le 2\sum_{e_i\in A}\sum_{j=1}^{k}\varepsilon^3\Delta^{-\lambda(\Delta,\varepsilon)}g_{e_j} && g_{e_i}\le p^2\varepsilon^7\Delta^{-\lambda(\Delta,\varepsilon)} \text{ by (5.3.10).}\\
&= 2\varepsilon^3\Delta^{-\lambda(\Delta,\varepsilon)}\sum_{e_i\in A}\sum_{j=1}^{k}g_{e_j}\\
&= 2\varepsilon^3\Delta^{-\lambda(\Delta,\varepsilon)}|A|g_v\\
&\le 2\varepsilon^3\Delta^{-\lambda(\Delta,\varepsilon)}|A|\\
&\le 4\varepsilon^3. && \text{Since } |A|\le 2\Delta^{\lambda(\Delta,\varepsilon)} \text{ by (5.20).}
\end{aligned}
\tag{5.27}
$$

Bounding the sum over B. Let us for each $e_i \in B$ use D_i to denote the set of edges $e_j \in B$ where $\operatorname{Cov}[h_{e_i}, h_{e_j} \mid H_e] \neq 0$. We claim that for each $e_i \in B$, $|D_i| \le \Delta^{\lambda(\Delta,\varepsilon)}$. To prove this, observe that for all $e_i, e_j \in B$, we have $d_P(u_i, u) \ge \lambda(\Delta, \varepsilon)$ and $d_P(u_j, u) \ge \lambda(\Delta, \varepsilon)$ by definition of B. Moreover, since $(u, v), (v, u_i), (v, u_j) \in N$, we have $d_P(u, v) \ge \lambda(\Delta, \varepsilon)$, $d_P(u_i, v) \ge \lambda(\Delta, \varepsilon)$, and $d_P(u_j, v) \ge \lambda(\Delta, \varepsilon)$ by Observation 5.3.3. Therefore among $\{v, u, u_i, u_j\}$ only the pair u_i, u_j may have $d_P(u_i, u_j) < \lambda(\Delta, \varepsilon)$. If this is not the case and $d_P(u_i, u_j) \ge \lambda(\Delta, \varepsilon)$, then based on Claim 5.3.6 events H_{e_i} and H_{e_j}, and consequently, h_{e_i} and h_{e_j} would be independent conditioned on H_e and thus $\operatorname{Cov}(h_{e_i}, h_{e_j} \mid H_e) = 0$. This means that indeed for any e_i and any $e_j \in D_i$, $d_P(u_i, u_j) \le \lambda(\Delta, \varepsilon)$. Since the maximum degree of P is Δ, there are at most

$\Delta^{\lambda(\Delta,\varepsilon)}$ such vertices, implying indeed that

$$|D_i| \le \Delta^{\lambda(\Delta,\varepsilon)} + 1 \overset{(5.20)}{\le} 2\Delta^{\lambda(\Delta,\varepsilon)} \qquad \text{for any } e_i \in B. \tag{5.28}$$

We therefore have:

$$\begin{aligned}
\sum_{e_i \in B} \sum_{e_j \in B} \mathrm{Cov}[h_{e_i} h_{e_j} \mid H_e] &= \sum_{e_i \in B} \sum_{e_j \in D_i} \mathrm{Cov}[h_{e_i} h_{e_j} \mid H_e] \\
&\le \sum_{e_i \in B} \sum_{e_j \in D_i} \frac{g_{e_i}}{p\varepsilon^2} \frac{g_{e_j}}{p\varepsilon^2} && \text{By (5.26).} \\
&\le \frac{1}{p^2\varepsilon^4} \sum_{e_i \in B} g_{e_i} \left(\sum_{e_j \in D_i} g_{e_j} \right) \\
&\le \frac{1}{p^2\varepsilon^4} \sum_{e_i \in B} g_{e_i} \left(\sum_{e_j \in D_i} p^2 \varepsilon^7 \Delta^{-\lambda(\Delta,\varepsilon)} \right) && \text{By Observation 5.3.10.} \\
&\le \frac{p^2 \varepsilon^7 \Delta^{-\lambda(\Delta,\varepsilon)}}{p^2 \varepsilon^4} \sum_{e_i \in B} g_{e_i} |D_i| \\
&\le 2\varepsilon^3 \sum_{e_i \in B} g_{e_i} && \text{By (5.28) } |D_i| \le 2\Delta^{\lambda(\Delta,\varepsilon)}. \\
&\le 2\varepsilon^3 g_v \le 2\varepsilon^3. && \text{Since } \mathbf{g} \text{ is a valid fractional matching.}
\end{aligned} \tag{5.29}$$

Incorporating (5.27) and (5.29) into (5.25) we get that $\mathrm{Var}[h_v \mid H_e] \le 4\varepsilon^3 + 2\varepsilon^3 = 6\varepsilon^3$. Replacing back to equation (5.24) we get that $\Pr[h_v > 1 + 3\varepsilon \mid H_e] \le 6\varepsilon^3/\varepsilon^2 = 6\varepsilon$. $\square$

5.3.4 Putting Everything Together

In this section we prove using the stated bounds above that $\mathbf{x}$ as constructed satisfies the fractional matching constraints (5.3-5.5), satisfies (5.6), i.e. has expected weight at least $(1 - O(\varepsilon))\textsc{opt}$, and that it is non-zero only on the edges of $\mathcal{Q}$. This as already described in Observation 5.3.1 completes the proof of Theorem 6 that subgraph Q guarantees a $(1 - \varepsilon)$-approximation.

Fractional matching constraints (5.3) and (5.4). For constraint (5.3) that $x_v \leq 1$ for any vertex v, consider two scenarios: If v is matched via a matching edge of Z (the matching constructed in Section 5.3.2 on P), then on all edges $e \in N$ we set $x_e = 0$ by construction of $\mathbf{h}$ (5.9) and thus $x_v = 1$. On the other hand, if v is unmatched in Z, then we still have $x_v \leq 1$ due to construction (5.10) of $\mathbf{x}$ based on $\mathbf{h}$ which guarantees $\mathbf{x} \leq \frac{1}{1+3\varepsilon}\mathbf{h}$ and in addition $x_v = 0$ if $h_v \geq 1 + 3\varepsilon$.

The constraint (5.4) that $x_e \geq 0$ for all edges e is easy to confirm. For edges in P, the value of x_e is either 0 or 1. For edges in N, since $\mathbf{f}$ is non-negative, so are $\mathbf{g}$, $\mathbf{h}$, and $\mathbf{x}$.

Blossom inequalities (5.5). The blossom constraint (5.5) that $x(U) \leq \frac{|U|-1}{2}$ for all odd size $U \subseteq V$ with $|U| \leq 1/\varepsilon$ follows for the following reason. There are two types of edges that form $\mathbf{x}$ by construction: Those in set P, and those in N. For any edge $e \in P$, the value of x_e is simply integral. For any $e \in N$, we have

$$x_e \overset{(5.10)}{<} h_e \overset{\text{Observation 5.3.17}}{\leq} p\varepsilon^5 \Delta^{-\lambda(\Delta,\varepsilon)} \leq p\varepsilon^5 \leq \varepsilon^5. \tag{5.30}$$

Now suppose for contradiction that there is a subset of size $\leq 1/\varepsilon$ for which the blossom constraint (5.5) is violated, and let U be the smallest such subset. If there is an edge $e = (u, v) \in P$ whose both endpoints are in U and $x_e = 1$, then one can confirm that subset $U \setminus \{u, v\}$ should also violate the blossom inequality contradicting that U is the smallest. On the other hand, for all edges e with both endpoints in U we have $x_e \leq \varepsilon^5$ by (5.30). Since there are at most $|U|^2$ edges inside U and $|U| \leq 1/\varepsilon$, we have $x(U) \leq |U|^2\varepsilon^5 \leq \varepsilon^{-2}\varepsilon^5 = \varepsilon^3 < 1 < \frac{|U|-1}{2}$, contradicting the fact that the blossom inequality is violated. So all blossom inequalities of size up to $1/\varepsilon$ must be satisfied.

Fractional matching x is non-zero only on $\mathcal{Q}$. For any edge $e \in P$, if $x_e > 0$ then $e \in Z$ and by Claim 5.3.6, $e \in \mathcal{P}$ i.e. e is realized. Since $P \subseteq Q$, then $e \in \mathcal{Q}$. On the other hand, for any edge $e \in N$, if $x_e > 0$ then we should have $h_e > 0$ by construction of $\mathbf{x}$ and to have $h_e > 0$ we should have $g_e > 0$ and $f_e > 0$. By construction of $\mathbf{h}$, if $h_e > 0$ then e must be realized, and by construction of $\mathbf{f}$, if $f_e > 0$ then $e \in S \subseteq Q$. Combination of these imply $e \in \mathcal{Q}$. Therefore overall, if for any edge e, $x_e > 0$ then $e \in \mathcal{Q}$ and so $\mathbf{x}$ is a fractional matching of only the edges in $\mathcal{Q}$.

Expected weight of x. By Claim 5.3.6 part 2, we have $\mathbb{E}[w(Z)] \geq (1-2\varepsilon)\chi(P)$ and thus $\mathbb{E}[\sum_{e\in P} x_e w_e] \geq (1-2\varepsilon)\chi(P)$. On the other hand, by Claim 5.3.18 $\mathbb{E}[\sum_{e\in N} x_e w_e] \geq (1-15\varepsilon)w(\mathbf{g})$ and $\mathbb{E}[w(\mathbf{g})] \geq (1-\varepsilon)\chi(N)$ by Claim 5.3.12. Combining all of these, we get

$$\mathbb{E}[w(\mathbf{x})] = \mathbb{E}\left[\sum_{e \in E} x_e w_e\right] = \mathbb{E}\left[\sum_{e \in P} x_e w_e\right] + \mathbb{E}\left[\sum_{e \in N} x_e w_e\right]$$
$$\geq (1-2\varepsilon)\chi(P) + (1-15\varepsilon)(1-\varepsilon)\chi(N)$$
$$\geq (1-16\varepsilon)(\chi(P)+\chi(N)) \overset{\text{Obs 5.3.2}}{\geq} (1-16\varepsilon)(1-\varepsilon)\textsc{opt} \geq (1-17\varepsilon)\textsc{opt}.$$

And thus our construction of $\mathbf{x}$ satisfies $\mathbb{E}[w(\mathbf{x})] \geq (1-O(\varepsilon))\textsc{opt}$ required by (5.6).

Combination of the properties above as shown before in Observation 5.3.1 proves Theorem 6, which is our main result.

5.4 The Weighted Vertex-Independent Matching Lemma

In this section, we turn to prove Lemma 5.3.4 which was used in Section 5.3. We restate the lemma below and for simplicity of notation, drop the primes in symbols such as $G', \mathcal{G}', \Delta'$ as stated in Section 5.3 and use $G, \mathcal{G}, \Delta$ instead.

Lemma 5.3.4. (restated)**.** *Let $G = (V, E, w)$ be an edge-weighted base graph with maximum degree Δ. Let $\mathcal{G}$ be a random subgraph of G that includes each edge $e \in E$ independently with some fixed probability $p \in (0, 1]$. Let $\mathcal{A}(H)$ be any (possibly randomized) algorithm that given any subgraph H of G, returns a (not necessarily maximum weight) matching of H. For any $\varepsilon > 0$ there is a randomized algorithm $\mathcal{B}$ to construct a matching $Z = \mathcal{B}(\mathcal{G})$ of $\mathcal{G}$ such that*

1. *For any vertex v, $\Pr_{\mathcal{G}\sim G,\mathcal{B}}[v \in Z] \leq \Pr_{\mathcal{G}\sim G,\mathcal{A}}[v \in \mathcal{A}(\mathcal{G})] + \varepsilon^3$.*

2. *$\mathbb{E}[w(Z)] \geq (1-\varepsilon)\mathbb{E}[w(\mathcal{A}(\mathcal{G}))]$.*

3. *For any vertex-subset* $\{v_1, v_2, \ldots\} \subseteq V$ *such that for all* i, j, $d_G(v_i, v_j) \geq \lambda$ *where* $\lambda = O(\varepsilon^{-24} \log \Delta \cdot \text{poly}(\log \log \Delta))$, *events* $\{v_1 \in Z\}, \{v_2 \in Z\}, \{v_3 \in Z\}, \ldots$ *are all independent with respect to both the randomizations used in algorithm* $\mathcal{B}$ *and in drawing* $\mathcal{G}$.

Outline of the proof. To prove this lemma, we need to design an algorithm $\mathcal{B}(\mathcal{G})$ that satisfies all three properties. If we only had the first two properties to satisfy, we could simply use algorithm $\mathcal{A}$. The problem however, becomes challenging when we need to, in addition, satisfy the third property regarding the independence between the events $\{v_1 \in Z\}, \{v_2 \in Z\}, \{v_3 \in Z\}, \ldots$ for vertices $v_1, v_2, \ldots$, that are pair-wise far enough from each other. To ensure that our algorithm meets this condition, similar to what we did in Chapter 4 for the unweighted variant of the lemma, we show that it can be implemented efficiently in the LOCAL model of computation (whose formal description follows).

The LOCAL model is a standard distributed computing model which consists of a network (graph) of processors with each processor having its own tape of random bits. Computation proceeds in synchronous rounds and in each round, processors can send unlimited size messages to each of their neighbors. Thus, to transmit a message from a node u to node v, we require at least $d(u, v)$ rounds. For the same reason, if an algorithm terminates within r-rounds of LOCAL, the output of any two nodes that have distance at least $2r$ from each other would be independent, which is essentially how we guarantee our independence property.

For simplicity, we explain our algorithm in a sequential setting in Algorithm 4, and later describe how it can be simulated in the LOCAL model. We define a recursive algorithm $\mathcal{B}_r(\mathcal{G})$ that given a parameter r, as the depth of recursion, and a subgraph of G, denoted by $\mathcal{G}$ outputs a matching of this graph. We give an informal overview of the algorithm in Section 5.4.1, and formally state it Section 5.4.2.

5.4.1 Overview of the Algorithm

We define a recursive algorithm $\mathcal{B}_r(\mathcal{G})$ that given a parameter r, as the depth of recursion, and a subgraph of G, denoted by $\mathcal{G}$ outputs a matching of this graph. We then set our algorithm $\mathcal{B}(\mathcal{G}) := \mathcal{B}_t(\mathcal{G})$ for a number $t = O(\varepsilon^{-20})$. For $r = 0$, algorithm $\mathcal{B}_0(\mathcal{G})$ simply returns an empty matching. For any $r > 0$, the idea is to use the matching constructed in $\mathcal{B}_{r-1}(\mathcal{G})$ and transform it to a one that is sufficiently heavier in expectation. However, this transformation needs to be in a way that the probability of a vertex being matched in $\mathcal{B}_r(\mathcal{G})$ is not significantly higher than $\Pr_{\mathcal{G}\sim G,\mathcal{A}}[v \in \mathcal{A}(\mathcal{G})]$. A useful observation here is that we do not need to ensure that for any given subgraph $\mathcal{G}$ algorithm $\mathcal{B}(\mathcal{G})$ gives a large enough matching while the probability of a vertex being matched in the algorithm is not greater than $\Pr_{\mathcal{G}'\sim G,\mathcal{A}}[v \in \mathcal{A}(\mathcal{G}')] + \varepsilon^3$, rather we need this to hold in expectation over realization of $\mathcal{G}$. We strongly use this observation in the design of our algorithm by drawing several (ε^{-12}) other random realization of G and simultaneously constructing a matching for each one. This way, we have the freedom of matching a vertex with a high probability in an instance, in the expense of the vertex being matched with a lower probability in another instance. Similarly, we might construct a relatively low-weight matching for an instance but compensate it by finding a relatively heavier matching in another one. More precisely, in $\mathcal{B}(\mathcal{G})$, we have $\alpha = \varepsilon^{-12} + 1$ random realizations of G, denoted by $\mathcal{G}_1, \ldots, \mathcal{G}_\alpha$, where $\mathcal{G}_1 = \mathcal{G}$, and our goal is to construct matchings $M'_1, \ldots, M'_\alpha$ for them simultaneously. Roughly speaking, since our input subgraph $\mathcal{G}$ is itself a random realization of G and that all these subgraphs are drawn from the same distribution, we achieve our goal if our algorithm performs as desired in average over these α realizations.

Below we provide a definition which we will use to refer to our subgraphs and their corresponding matching.

Definition 5.4.1 (profiles). *We say $((\mathcal{G}_1, M_1), \ldots, (\mathcal{G}_k, M_k))$ is a profile of size k, iff for any $i \in [k]$, $\mathcal{G}_i$ is a subgraph of G and M_i is a matching on $\mathcal{G}_i$.*

To construct matchings $M'_1, \ldots, M'_\alpha$ for subgraphs $\mathcal{G}_1, \ldots, \mathcal{G}_\alpha$ in algorithm $\mathcal{B}_r(\mathcal{G})$, we start by running $\mathcal{B}_r(\mathcal{G}_i)$ for any $i \in [\alpha]$, and obtain matchings $M_1, \ldots, M_\alpha$ as a result. In the other words, we start from profile $((\mathcal{G}_1, M_1), \ldots, (\mathcal{G}_\alpha, M_\alpha))$ and want to transform it to $((\mathcal{G}_1, M'_1), \ldots, (\mathcal{G}_\alpha, M'_\alpha))$ such that $\mathbb{E}[w(M'_i)]$ is sufficiently greater than $\mathbb{E}[w(M_i)]$ for a random $i \in [\alpha]$, while the constraints in the second and third properties of Lemma 5.3.4 are not violated. To get this, we use an idea similar to finding augmenting paths in the classic weighted matching algorithms. However, ours rather than being a path, is a structure that consists of multiple paths in graphs $\mathcal{G}_1, \ldots, \mathcal{G}_\alpha$. We call this structure a *multi-walk* and formally define it in Definition 5.4.2. Similar to how augmenting paths are used, we will use this structure to flip the membership of some edges in their corresponding matchings with the goal of increasing the expected size of the matchings. However, note that if we naively choose the multi-walks with the sole purpose of increasing the average size of the matchings, we might violate the second property of lemma, as it might lead to some vertices being matched with an undesirably large probability. Further, these multi-walks should not include vertices that are further than a threshold since otherwise we might violate the third property of the lemma. To overcome the first issue, after probability of a vertex v being matched in our algorithm reaches a threshold, we mark it as *saturated*. When a vertex is saturated, our algorithm ensures that while augmenting the matchings (using multi-walks), it does not increase the number of matchings in which this vertex is matched. Having these constrains narrows down our choices of augmenting structures (multi-walks) significantly. However, we give a constructive proof (using Algorithm 5), and show that this narrow set includes a subset that can be used to increase the average size of our matchings sufficiently.

5.4.2 Algorithm $\mathcal{B}(\mathcal{G})$

We start by providing some definitions that will be used in the Algorithm.

Definition 5.4.2 (multi-walks)**.** *We define $W = ((s_1, e_1), \ldots, (s_l, e_l))$ to be a multi-walk of length l of profile $P = ((\mathcal{G}_1, M_1), \ldots, (\mathcal{G}_k, M_k))$ iff it satisfies the following conditions.*

- *For any $i \in [l]$, we have $s_i \in [k]$, and e_i is an edge in subgraph $\mathcal{G}_{s_i}$.*
- *$(e_1, \ldots, e_k)$ is a walk in graph G.*
- *W contains distinct elements, e.g., for any i and j, we have $(s_i, e_i) \neq (s_j, e_j)$.*

Given a profile $P = ((\mathcal{G}_1, M_1), \ldots, (\mathcal{G}_j, M_k))$ *and a multi-walk* $W = ((s_1, e_1), \ldots, (s_l, e_l))$, *we say* $P \oplus W = ((\mathcal{G}_1, M_1'), \ldots, (\mathcal{G}_j, M_k'))$ *is the result of applying* W *on* P *iff for any* $i \in [k]$, M_i' *is constructed as follows:*

$$M_i' = M_i \cup \{e_j \,|\, i = s_j \text{ and } e_j \notin M_i\} \backslash \{e_j \,|\, i = s_j \text{ and } e_j \in M_i\}.$$

Definition 5.4.3 (alternating multi-walks)**.** *A multi-walk* $W = ((s_1, e_1), \ldots, (s_k, e_k))$ *of profile* $P = ((\mathcal{G}_1, M_1), \ldots, (\mathcal{G}_\alpha, M_\alpha))$ *is an alternating multi-walk iff it satisfies the two following conditions. First, for any* $i \in [k-1]$ *we have* $\mathbf{1}(e_i \in M_{s_i}) + \mathbf{1}(e_{i+1} \in M_{s_{i+1}}) = 1$, *and second,* $P \oplus W$ *is a profile. We further define* $g(W, P)$, *the gain of applying alternating multi-walk* W *on* P, *as*

$$g(W, P) = \sum_{(i,e') \in W} (\mathbf{1}(e' \notin M_i) - \mathbf{1}(e' \in M_i)) w(e').$$

Given a vertex $v \in V$ and an alternating multi-walk $W = ((s_1, e_1), \ldots, (s_l, e_l))$ of profile $P = ((\mathcal{G}_1, M_1), \ldots, (\mathcal{G}_k, M_k))$, we define $d_{W,v}$ and $\bar{d}_{W,v}$ as follows:

$$d_{W,v} = |\{i : v \in e_i, \text{ and } e_i \in M_{s_i}\}| \text{ and } \bar{d}_{W,v} = |\{i : v \in e_i, \text{ and } e_i \notin M_{s_i}\}|. \tag{5.31}$$

Definition 5.4.4 (applicable multi-walks)**.** *Given a multi-walk* $W = ((s_1, e_1), \ldots, (s_l, e_l))$ *of profile* P, *and a subset of vertices* V_s, *we say* W *is* applicable *with respect to a set of vertices* V_s *iff it is alternating and for any* $v \in V_s$ *it satisfies* $d_{W,v} \geq \bar{d}_{W,v}$.

To prove Lemma 5.3.4, we design an algorithm $\mathcal{B}$ that given a random realization of G outputs a matching Z and show that it satisfies the desired properties of the lemma. In 4, we provide a recursive algorithm $\mathcal{B}_r(\mathcal{G})$ that given an integer number r and a realization $\mathcal{G}$ of G outputs a matching of $\mathcal{G}$. We set $\mathcal{B}(\mathcal{G}) = \mathcal{B}_t(\mathcal{G})$ for $t = c_t\varepsilon^{-20}$ where c_t is a constant number. (We fix the value of c_t later.)

Algorithm 6. $\mathcal{B}_r(\mathcal{G})$

1 If $r = 0$, return an empty matching.
2 Set $\alpha \leftarrow \varepsilon^{-12} + 1$, $l \leftarrow 3\varepsilon^{-3}$.
3 For any $i \in [\alpha]$, construct $\mathcal{G}_i$ as follows. We set $\mathcal{G}_1 := \mathcal{G}$, and for any $1 < r$ subgraph $\mathcal{G}_i$ includes any edge $e \in G$ independently with probability p.
4 Define profile $P := ((\mathcal{G}_1, M_1), \dots, (\mathcal{G}_\alpha, M_\alpha))$ where $M_i := \mathcal{B}_{r-1}(\mathcal{G}_i)$.
5 Call a vertex v *saturated* iff $\Pr_{\mathcal{G}'\sim G,\mathcal{B}}[v \in Z_{r-1}] \le \Pr_{\mathcal{G}\sim G,\mathcal{A}}[v \in \mathcal{A}(\mathcal{G}')] + \varepsilon^3 - 1/\alpha$, and *unsaturated* otherwise.
6 Let $\mathcal{W}_a$ be the set of alternating multi-walks of P that are applicable with respect to the set of saturated vertices.
7 Construct the weighted hyper-graph $H = (V, E_H)$ as follows. For any multi-walk W in set $\mathcal{W}_a$ with length at most l, H contains a hyper-edge between vertices in W with weight $g(W, P)$.
8 $M_H \leftarrow \mathsf{ApproxMatching}(H)$. // See Proposition 5.4.18 for the $\mathsf{ApproxMatching}()$ algorithm.
9 Iterate over all hyper-edges in M_H, apply their corresponding multi-walks on P, and let $P' := ((\mathcal{G}_1, M'_1), \dots, (\mathcal{G}_\alpha, M'_\alpha))$ be the final profile.
10 Return matching M'_1.

Observation 5.4.5. *For any r, matchings $M'_1, \dots, M'_\alpha$ in Algorithm $\mathcal{B}_r(\mathcal{G})$ are random variables that are drawn from the same distribution.*

Proof. This is due to the fact that matchings $M_1, \dots, M_\alpha$ are independent random variables from the same distribution, and that to obtain $M'_1, \dots, M'_\alpha$, based on these matchings, algorithm does not treat them differently. □

Before proceeding to the proof of the three properties let us prove the following lemma about alternating multi-walks.

Lemma 5.4.6. *Consider profile* $P = ((\mathcal{G}_1, M_1), \ldots, (\mathcal{G}_j, M_\alpha))$ *and a multi-walk of this profile* $W = ((s_1, e_1), \ldots, (s_k, e_k))$ *with* $e_i = (u_i, u_{i+1})$ *for any* $i \in [k]$*. If* W *is an alternating multi-walk, then it satisfies the following properties:*

1. *For any* $v \in V$*, if* $v \notin \{u_1, u_{k+1}\}$*, then we have* $d_{W,v} = \bar{d}_{W,v}$*.*

2. *If* $e_1 \in M_{s_1}$*, then we have* $d_{W,u_1} \geq \bar{d}_{W,u_1}$*. Also, if* $e_1 \notin M_{s_1}$*, we have* $d_{W,u_1} \leq \bar{d}_{W,u_1}$*.*

3. *If* $e_1 \in M_{s_1}$ *and* $e_k \in M_{s_k}$*, then* W *is applicable with respect to any subset of* V*.*

Proof. Observe that for any $i > 1$, we have $u_i \in e_i$ and $u_i \in e_{i-1}$. Consider an arbitrary vertex $v \in V$. Since W is alternating, for any $1 < j \leq k$ that $v = u_j$, we either have $e_{j-1} \in M_{s_{j-1}}$ and $e_j \notin M_{s_j}$ or $e_{j-1} \notin M_{s_{j-1}}$ and $e_j \in M_{s_j}$. This implies:

$$d_{W,u_1} = |\{i : 1 < i \leq k, v = u_i\}| + \mathbf{1}(v = u_1, e_1 \in M_{s_1}) + \mathbf{1}(v = u_{k+1}, e_k \in M_{s_k}),$$

and

$$\bar{d}_{W,u_1} = |\{i : 1 < i \leq k, v = u_i\}| + \mathbf{1}(v = u_1, e_1 \notin M_{s_1}) + \mathbf{1}(v = u_{k+1}, e_k \notin M_{s_k}).$$

Note that if $v \notin \{u_1, u_{k+1}\}$, then we have

$$\begin{aligned} d_{W,v} &= |\{i : v \in e_i, \text{ and } e_i \in M_{s_i}\}| = |\{i : 1 < i \leq k, v = u_i\}| \\ &= |\{i : v \in e_i, \text{ and } e_i \notin M_{s_i}\}| = \bar{d}_{W,v}, \end{aligned}$$

which completes the proof of the first item. To prove the second item, note that if $e_1 \in M_{s_1}$, then we have $\mathbf{1}(u_1 = u_1, e_1 \in M_{s_1}) = 1$ and $\mathbf{1}(u_1 = u_1, e_1 \notin M_{s_1}) = 0$, which gives us

$$\mathbf{1}(u_1 = u_1, e_1 \in M_{s_1}) + \mathbf{1}(u_1 = u_{k+1}, e_k \in M_{s_k}) \geq \mathbf{1}(u_1 = u_1, e_1 \notin M_{s_1}) + \mathbf{1}(u_1 = u_{k+1}, e_k \notin M_{s_k}),$$

and results in $d_{W,u_1} \geq \bar{d}_{W,u_1}$. A similar argument shows that if $e_1 \notin M_{s_1}$, then $d_{W,u_1} \leq \bar{d}_{W,u_1}$ holds.

Since multi-walks are not directed the second claim of the lemma can also be interpreted as follows. If $e_k \in M_{s_k}$ then, $d_{W,u_{k+1}} \geq \bar{d}_{W,u_{k+1}}$. Combining this with the first claim of the lemma, we obtain that if $e_1 \in M_{s_1}$, and $e_k \in M_{s_k}$, then for any $v \in V$, we have $d_{W,v} \geq \bar{d}_{W,v}$. By definition of applicable multi-walks, this means that if $e_1 \in M_{s_1}$, and $e_k \in M_{s_k}$ then multi-walk W is applicable with respect to any subset of V. This completes the proof the lemma. $\square$

5.4.3 Lemma 5.3.4 Property 1: Matching Probabilities

In this section our goal is to prove that Algorithm $\mathcal{B}(\mathcal{G})$ satisfies the first property of Lemma 5.3.4 as follows.

Lemma 5.4.7. *For any vertex $v \in V$, we have* $\Pr_{\mathcal{G}\sim G,\mathcal{B}}[v \in Z] \leq \Pr_{\mathcal{G}\sim G,\mathcal{A}}[v \in \mathcal{A}(\mathcal{G})] + \varepsilon^3$.

Proof. We will prove a stronger claim which is for any $v \in V$, and any $r \leq t$, we have

$q_{r,v} \leq q_v^{\mathcal{A}} + \varepsilon^3$, where

$$q_{r,v} := \Pr_{\mathcal{G} \sim G, \mathcal{B}}[v \in \mathcal{B}_r(\mathcal{G})], \quad \text{and} \quad q_v^{\mathcal{A}} := \Pr_{\mathcal{G} \sim G, \mathcal{B}}[v \in \mathcal{A}(\mathcal{G})].$$

We use proof by induction. The claim obviously holds for $r = 0$. For any $r > 0$, we assume that $q_{r-1,v} \leq q_v^{\mathcal{A}} + \varepsilon^3$ holds and obtain $q_{r,v} \leq q_v^{\mathcal{A}}$. Draw a random realization of G and denote it by $\mathcal{G}$ (i.e. $\mathcal{G} \sim G$). Consider matchings $M_i, \ldots, M_\alpha$, and $M'_i, \ldots, M'_\alpha$ from algorithm $\mathcal{B}_r(\mathcal{G})$, and let us define

$$\rho_{r,v} := |\{i : v \in M_i\}|/\alpha, \quad \text{and} \quad \rho'_{r,v} = |\{i : v \in M'_i\}|/\alpha.$$

We claim that $q_{r-1,v} = \rho_{r,v}$ and $q_{r,v} = \rho'_{r,v}$ hold. The former is due to the fact that any $i \in [\alpha]$, M_i is the result of running algorithm $\mathcal{B}_{r-1}$ on a random realization of G which by definition is equal to $q_{r-1,v}$. For the latter, note that we have $M'_i = \mathcal{B}_r(\mathcal{G})$ and by Observation 5.4.5, we know that matchings $M'_i, \ldots, M'_\alpha$ are drawn from the same distribution. As a result, we get

$$|\{i : v \in M'_i\}| = \alpha \Pr[v \in \mathcal{B}_r(\mathcal{G})],$$

which implies $q_{r,v} = \rho'_{r,v}$.

We prove our induction step for the cases of $q_{r-1,v} \leq q_v^{\mathcal{A}} + \varepsilon^3 - 1/\alpha$, and $q_{r-1,v} > q_v^{\mathcal{A}} + \varepsilon^3 - 1/\alpha$ separately. We first show that if $q_{r-1,v} \leq q_v^{\mathcal{A}} + \varepsilon^3 - 1/\alpha$, (i.e., v is not saturated), then $\rho_{r,v} \geq \rho'_{r,v} - 1/\alpha$ holds, which can be interpreted as

$$q_v^{\mathcal{A}} + \varepsilon^3 - 1/\alpha \geq q_{r-1,v} \geq q_{r,v} - 1/\alpha,$$

and as a result $q_v^{\mathcal{A}} + \varepsilon^3 \geq q_{r,v}$. Let W_H denote the set of multi-walks corresponding to edges in M_H constructed in $\mathcal{B}_r(\mathcal{G})$. Since M_H is a matching, for any vertex v, there exists at most one multi-walk $W \in W_H$ that contains vertex v. In addition, since W is alternating, we have $|d_{W,v} - \bar{d}_{W,v}| \leq 1$, where $d_{W,v}$ and $\bar{d}_{W,v}$ are defined as

$$d_{W,v} = |\{i : v \in e_i, \text{ and } e_i \in M_{s_i}\}| \text{ and } \bar{d}_{W,v} = |\{i : v \in e_i, \text{ and } e_i \notin M_{s_i}\}|.$$

Since after applying a multi-walk W on a profile, membership of the edges in W flips in their corresponding matchings, we get $|\{i : v \in M_i\}| \geq |\{i : v \in M'_i\}| - 1$ which means $\rho_{r,v} \geq \rho'_{r,v} - 1/\alpha$. We now consider the case of $q_{r-1,v} \geq q_v^{\mathcal{A}} + \varepsilon^3 - 1/\alpha$, (i.e., v is saturated) and show that in this case, $\rho_{r,v} \geq \rho'_{r,v}$ holds. Due to W being applicable with respect to the set of saturated vertices, by Definition 5.4.4, it satisfies $d_{W,v} \geq \bar{d}_{W,v}$. This directly yields $\rho_{r,v} \geq \rho'_{r,v}$, and as a result $q_{r-1,v} \geq q_{r,v}$. Based on the induction hypothesis, we have $q_{r-1,v} \leq q_v^{\mathcal{A}} + \varepsilon^3$ which implies $q_{r,v} \leq q_v^{\mathcal{A}} + \varepsilon^3$ and completes the proof. □

5.4.4 Lemma 5.3.4 Property 2: Expected Weight of the Matching

In this section, our goal is to prove $\mathbb{E}[w(Z)] \geq (1 - \varepsilon)$, where $Z = \mathcal{B}_t(\mathcal{G})$ for $t = c_t\varepsilon^{-20}$. We will fix the value of the constant c_t later in this section.

We start by Lemma 5.4.8 concerning the relation between the expected weight of the matching and the weight of matching M_H on hyper-graph H in the algorithm. For any r, let $M_{H,r}$ denote the matching M_H in algorithm $\mathcal{B}_r(\mathcal{G})$.

Lemma 5.4.8. *For any $0 < r \leq t$, we have*

$$\mathbb{E}_{\mathcal{G}\sim G}[\mathcal{B}_r(\mathcal{G})] = \mathbb{E}_{\mathcal{G}\sim G}[\mathcal{B}_{r-1}(\mathcal{G})] + \mathbb{E}[w(M_{H,r})]/\alpha.$$

Proof. Consider algorithm $\mathcal{B}_r(\mathcal{G})$ where $\mathcal{G}$ is a random realization of G. To prove this lemma, we will show

$$\sum_{i\in\alpha} w(M_i') - \sum_{i\in\alpha} w(M_i) = w(M_{H,r}). \tag{5.32}$$

By Algorithm 4, we have $\mathcal{B}_r(\mathcal{G}) = M_1'$. Moreover, Observation 5.4.5 states that matchings $M_1', \ldots, M_\alpha'$ are all drawn from the same distribution which implies

$$\mathbb{E}\left[\sum_{i\in\alpha} w(M_i')\right] = \alpha\mathbb{E}_{\mathcal{G}\sim G}[\mathcal{B}_r(\mathcal{G})].$$

Similarly, since matchings $M_1, \ldots, M_\alpha$ are all drawn from the same distribution as $\mathcal{B}_{r-1}(\mathcal{G})$ we have

$$\mathbb{E}\left[\sum_{i\in\alpha} w(M_i)\right] = \alpha\mathbb{E}_{\mathcal{G}\sim G}[\mathcal{B}_{r-1}(\mathcal{G})].$$

Consequently, to prove the lemma, it suffices to prove Equation 5.32 holds. Let W_H denote the set of multi-walks corresponding to edges in M_H constructed in $\mathcal{B}_r(\mathcal{G})$. Since the weight of each edge in H is equal to the gain of its corresponding multi-walk, we can write

$$w(M_H) = \sum_{W \in W_H} g(W, P) = \sum_{W \in W_H} \sum_{(i,e) \in W} (\mathbf{1}(e \notin M_i) - \mathbf{1}(e \in M_i))w(e). \tag{5.33}$$

Note that profile P' is the result of iteratively applying the set of multi-walks W_H on profile P. However, since M_H is a matching, and as a result multi-walks in W_H are vertex disjoint, gain of a multi-walk is not affected by the multi-walks applied before that. Moreover, since different multi-walks concern different vertices of the graph, we can assume w.l.o.g, that we apply all of them at the same time. Let us define for any $i \in [\alpha]$,

$$E_{i,1} = \bigcup_{W \in W_H} \{e \,|\, (i,e) \in W \text{ and } e \notin M_i\}, \text{ and } E_{i,2} = \bigcup_{W \in W_H} \{e \,|\, (i,e) \in W \text{ and } e \in M_i\}.$$

By Definition 5.4.1, for any $i \in [\alpha]$, we have $M'_i = M_i \cup E_{i,1} \backslash E_{i,2}$. This implies

$$w(M'_i) - w(M_i) = \sum_{e \in E_{i,1}} w_e - \sum_{e \in E_{i,2}} w_e = \sum_{W \in W_H} \sum_{(j,e) \in W, j=i} (\mathbf{1}(e' \notin M_j) - \mathbf{1}(e' \in M_j))w(e'),$$

and as a result

$$\sum_{i \in [\alpha]} w(M'_i) - w(M_i) = \sum_{W \in W_H} \sum_{(j,e) \in W} (\mathbf{1}(e' \notin M_j) - \mathbf{1}(e' \in M_j))w(e').$$

Combining this with Equation 5.33 results in Equation 5.32 and completes the proof.

□

For any $r \leq t$, let $Z_r := \mathcal{B}_r(\mathcal{G})$. Given Lemma 5.4.8, to prove the second property, it suffices to show that for any r having $\mathbb{E}[w(Z_r)] < (1-\varepsilon)\mathbb{E}[w(\mathcal{A}(\mathcal{G}))]$ results in $\mathbb{E}[w(M_{H,r})] \geq \alpha\mathbb{E}[w(\mathcal{A}(\mathcal{G}))]/t$. Based on Lemma 5.4.8, this implies

$$\mathbb{E}[w(Z_t)] \geq \sum_{r<t} \mathbb{E}[w(M_{H,r})]/\alpha \geq \min(t\alpha\mathbb{E}[w(\mathcal{A}(\mathcal{G}))]/(t\alpha), (1-\varepsilon)\mathbb{E}[w(\mathcal{A}(\mathcal{G}))])$$
$$= (1-\varepsilon)\mathbb{E}[w(\mathcal{A}(\mathcal{G})),$$

which is equivalent to the second property of Lemma 5.3.4. To achieve this, in Lemma 5.4.9 (stated below), we prove that having $\mathbb{E}[w(Z)] < (1-\varepsilon)\mathbb{E}[w(\mathcal{A}(\mathcal{G}))]$ results in $\mathbb{E}[w(M_{H,r})] = \Omega(\varepsilon^8\mathbb{E}[w(\mathcal{A}(\mathcal{G}))])$, which can be interpreted as $\mathbb{E}[w(M_{H,r})] \geq c\varepsilon^8\mathbb{E}[w(\mathcal{A}(\mathcal{G}))]$ for a constant number c. By setting

$$c_t = \frac{\varepsilon^{-12}+1}{c\varepsilon^{-12}},$$

we get

$$\mathbb{E}[w(M_{H,r})] = c\varepsilon^8\mathbb{E}[w(\mathcal{A}(\mathcal{G}))] = \frac{(\varepsilon^{-12}+1)\mathbb{E}[w(\mathcal{A}(\mathcal{G}))]}{c_t\varepsilon^{-20}}.$$

Recall that we have $t = c_t\varepsilon^{-20}$, and $\alpha = \varepsilon^{-12}+1$, which gives us $\mathbb{E}[w(M_{H,r})] \geq \alpha\mathbb{E}[w(\mathcal{A}(\mathcal{G}))]/t$. Therefore, to prove the second property of Lemma 5.3.4, it only suffices to prove the following lemma.

Lemma 5.4.9. *For any $r \leq t$, if $\mathbb{E}[w(Z)] < (1-\varepsilon)\mathbb{E}[w(\mathcal{A}(\mathcal{G}))]$ holds, then we have $\mathbb{E}[w(M_{H,r})] = \Omega(\varepsilon^8\mathbb{E}[w(\mathcal{A}(\mathcal{G}))])$.*

Proof. To prove this, we will construct a subgraph H' of H which max-degree 2α such that

$$\mathbb{E}\left[\sum_{e\in H'} w(e)\right] \geq \alpha\varepsilon^2\mathbb{E}[w(\mathcal{A}(\mathcal{G}))].$$

First, note that H is a hyper-graph of rank $l = 3\varepsilon^{-3}$ since each edge is between the vertices of a path of length at most l in G. Using Lemma 5.4.15, we know that subgraph H' (and as a result hyper-graph H) has a matching of weight $\sum_{e\in H'} w(e)/(2l\alpha)$ which is in expectation equal to $\varepsilon^5\mathbb{E}[w(\mathcal{A}(\mathcal{G}))]/6$. Moreover, $M_{H,r}$ is constructed by $\mathsf{ApproxMatching}(H)$ which by Proposition 5.4.18 returns an $O(l)$-approximation of the maximum weight matching of H. Thus, we get

$$\mathbb{E}[w(M_{H,r})] = \Omega(\varepsilon^8\mathbb{E}[w(\mathcal{A}(\mathcal{G}))]).$$

Before proceeding to the construction of H' in Algorithm 5, let us provide some definitions. Given a profile $P = ((\mathcal{G}_1, M_1), \ldots, (\mathcal{G}_j, M_k))$, we say $W = ((s_1, e_1), \ldots, (s_a, e_a))$, an alternating multi-walk of P, is *expandable* by $W' = ((s'_1, e'_1), \ldots, (s'_b, e'_b))$ iff either W_1 or W_2, defined below, is an alternating multi-walk:

$$W_1 = ((s_1, e_1), \ldots, (s_a, e_a), (s'_1, e'_1), \ldots, (s'_b, e'_b)),$$

$$W_2 = ((s'_1, e'_1), \ldots, (s'_b, e'_b), (s_1, e_1), \ldots, (s_a, e_a)).$$

If W is expandable by W' either one of W_1 and W_2 that is an alternating multi-walk is the result of expanding W by W'. (If both are alternating multi-walks, we pick one arbitrarily.) Similarly, we say W is expandable by a path or a cycle $p = (e'_1, \ldots, e'_b)$ in graph G_i iff W is expandable by $((i, e'_1), \ldots, (i, e'_b))$, and the result of expanding W by p is similar to expanding W by $((i, e'_1), \ldots, (i, e'_b))$.

Below we state Algorithm 5 which given profile P and the set of saturated vertices V_s outputs hyper-graph H'. Note that both P and V_s are from algorithm $\mathcal{B}_r(\mathcal{G})$ by which $M_{H,r}$ is constructed.

Algorithm 7. Constructing subgraph H' given profile $P := ((\mathcal{G}_1, M_1), \ldots, (\mathcal{G}_\alpha, M_\alpha))$ and V_s.

```
Define H' to be a hyper-graph with vertex set V that initially does not have any
  edges.
For any i ∈ [α], let M_i^A := A(G_i), and E'_i := {e ∈ G_i | 1(e ∈ M_i) + 1(e ∈ M_i^A) = 1}
  // E'_i contains an edge if it is in exactly one of M_i and M_i^A.
Let V_r := {v ∈ V_s : |{i : v ∈ M_i^A}| > |{i : v ∈ M_i}|}.
Remove an edge e from E'_i iff e ∈ M_i^A and at least one of its end-points is in V_r.
Let G'_i := (V, E'_i).
while there exists an i ∈ α, where E'_i ≠ ∅, do
    Let W be an empty multi-walk.
    Pick a maximal path or a cycle p from G'_i.
    If W is expandable by p, expand W by p, and and remove all the edges of p
      from E'_j.
    while there exists a subgraph G'_j that contains a maximal path or a cycle p by
      which W is expandable, do
        Expand W by p and remove all the edges of p from E'_j.
    Add W to 𝒲.
for any W ∈ 𝒲, do
    Pick an integer number x between 0 and l/4 − 1 uniformly at random.
    Decompose W = ((s_1, e_1), ..., (s_k, e_k)) to smaller multi-walks W_1, ..., W_a by
      removing any element (s_i, e_i) from the multi-walk iff e_i ∉ M_{s_i} and either
      i mod (l/4) = x or i mod (l/4) = x + 1 hold.
    If W_1 is expandable by W_a, expand W_1 by W_a, and set W_a to be an empty
      multi-walk.
    For any multi-walk W' ∈ {W_1, ... W_a}, add an edge to hyper-graph H' between
      the vertices in W' with weight g(W').
Return H'.
```

To complete the proof of Lemma 5.4.9, we need to show that hyper-graph H' outputted by Algorithm 5, has the three following properties.

1. The maximum degree of hyper-graph H' is upper-bounded by 2α.

2. hyper-graph H' is a subgraph of hyper-graph H.

3. We have $\mathbb{E}[\sum_{e \in H'} w(e)] \geq \alpha\varepsilon^2 \mathbb{E}[w(\mathcal{A}(\mathcal{G}))]$.

For the first property of H' first observe that any hyper-edge $e \in H'$ represents a multi-walk W_e in P. For any vertex v, if $v \in e$, then W_e contains an element (i, e') where $v \in e'$ and $e' \in \mathcal{G}'_i$. Moreover, in the algorithm, after using (i, e') in construction of a multi-walk, we remove e' from subgraph $\mathcal{G}'_i$. (see Line 11 of Algorithm 5.) We also know that degree of each vertex in $\mathcal{G}'_i$ is at most two. This gives us an upper-bound of 2α for degree of each vertex in H'.

To prove the second property, let us first recall that based on Line 7 of Algorithm 4, hyper-graph H has a hyper-edge for any multi-walk of length at most l in set $\mathcal{W}_a$ (which is defined as the set of multi-walks of P that are applicable with respect to the set of saturated vertices). To prove this property, it suffices to show that any hyper-edge in H' also represent a multi-walk of length at most l in $\mathcal{W}_a$. Since in both graphs H and H', weight of each edge is set to be the gain of its corresponding multi-walk, we do not need to consider the edge-weights in our proof. Consider a multi-walk W' from Line 17 of Algorithm 5. Since any edge in H' represents a multi-walk described in this line of the algorithm, to complete the proof we only need to show that W' is a multi-walk of lenght at most l in $\mathcal{W}_a$. Clearly, the length of this multi-walk is at most l due to Line 15 of Algorithm 5. Moreover, Lemma 5.4.14 states that W' is an alternating multi-walk and is applicable with respect to the saturated vertices, which implies $W' \in \mathcal{W}_a$, and completes the proof of this property.

To give a lower-bound for $\mathbb{E}[\sum_{e\in H'} w(e)]$ we will prove that

$$\mathbb{E}\left[\sum_{e\in H'} w(e)\right] \geq \alpha((1-3\varepsilon^3)\mathbb{E}[w(\mathcal{A}(\mathcal{G}))] - \mathbb{E}[w(Z)]),$$

which considering $\mathbb{E}[w(Z)] < (1-\varepsilon)\mathbb{E}[w(\mathcal{A}(\mathcal{G}))]$ in the statement of lemma results in:

$$\mathbb{E}\left[\sum_{e\in H'} w(e)\right] \geq \alpha(\varepsilon - 3\varepsilon^3)\mathbb{E}[w(\mathcal{A}(\mathcal{G}))].$$

For a small enough ε that satisfies $\varepsilon^2 > \varepsilon - 3\varepsilon^3$ we can write this as

$$\mathbb{E}\left[\sum_{e\in H'} w(e)\right] \geq \alpha\varepsilon^2\mathbb{E}[w(\mathcal{A}(\mathcal{G}))],$$

which is equivalent to the third property of H'. For any $e \in H'$,let W_e be the multi-walk in Line 17 of Algorithm 5 represented by e. By definition of $g(W_e, P)$, and the fact that for any $(i, e') \in W_e$, if $e' \notin M_i$, then $e' \in M_i^{\mathcal{A}}$ we get:

$$\begin{aligned} w(e) = g(W_e, P) &= \sum_{(i,e')\in W_p} (\mathbf{1}(e' \notin M_i) - \mathbf{1}(e' \in M_i))w(e') \\ &= \sum_{(i,e')\in W_p} (\mathbf{1}(e' \in M_i^{\mathcal{A}}) - \mathbf{1}(e' \in M_i))w(e'). \end{aligned}$$

Observe that based on *Algorithm* 5, for any $i \in [\alpha]$ and any edge $e' \in M_i$, there exists an edge $e \in H'$ such that $(i, e') \in W_e$. Similarly, for any $i \in [\alpha]$ and any edge $e' \in M_i^{\mathcal{A}}$, there exists an edge $e \in H'$ such that $(i, e') \in W_e$ unless e' is removed in Line 4 of the algorithm or (i, e') is removed in Line 15 of the algorithm. Based on Lemma 5.4.16 we know that probability of e' being removed in Line 4 is upper-bounded by ε^3. Moreover, it is easy to see that probability of (i, e') being removed in Line 15 is upper-bounded by $4/l = 4\varepsilon^3/3$. This means that with probability of at least $1 - 3\varepsilon^3$, for any $i \in [\alpha]$ and any edge $e' \in M_i^{\mathcal{A}}$, there exists an edge $e \in H'$ such that $(i, e') \in W_e$. This implies

$$\begin{aligned}\mathbb{E}\left[\sum_{e \in H'} w(e)\right] &= \sum_{i \in \alpha}\left(\sum_{e' \in M_i^{\mathcal{A}}} (1 - 3\varepsilon^3) w(e') - \sum_{e' \in M_i} w(e')\right) \\ &= \sum_{i \in \alpha}\big((1 - 3\varepsilon^3) w(M_i^{\mathcal{A}}) - w(M_i)\big).\end{aligned}$$

Since matchings $M_1, \ldots, M_\alpha$ are drawn from the same distribution, and similarly, matchings $M_1^{\mathcal{A}}, \ldots, M_\alpha^{\mathcal{A}}$ are drawn from the same distribution, for any $i \in [\alpha]$ we have $\mathbb{E}[w(M_i)] = \mathbb{E}[w(Z_r)]$ and $\mathbb{E}[w(M_i^{\mathcal{A}})] = \mathbb{E}[w(\mathcal{A}(\mathcal{G}))]$. This gives us

$$\mathbb{E}\left[\sum_{e \in H'} w(e)\right] = \alpha\big((1 - 3\varepsilon^3)\mathbb{E}[w(\mathcal{A}(\mathcal{G}))] - \mathbb{E}[w(Z_r)]\big),$$

and concludes the proof of this Lemma. □

Lemma 5.4.10. *Consider multi-walks $\{W_1, \ldots, W_a\}$ in Line 17 of Algorithm 5. If there exists an $i \in [a]$, where W_i is not applicable with respect to set V_s, then W is not applicable with respect to this set either.*

Proof. We use proof by contradiction. We assume that $W = ((s_1, e_1), \ldots, (s_k, e_k))$ is an alternating multi-walk applicable with respect to set V_s while there exists an $i \in [a]$ where W_i is not applicable with respect to this set. We then show that this leads to a contradiction. If W_i is not applicable with respect to V_s, then either it is not alternating, or there exists a vertex $v \in V_s$ for which $d_{W_i,v} < \bar{d}_{W_i,v}$. By Lemma 5.4.6, if W is alternating then any $v \in V$ that satisfies $d_{W_i,v} < \bar{d}_{W_i,v}$ is an end-point of W_i. Therefore, to obtain a contradiction, it suffices to prove that W is alternating, and that if $v \in V_s$ is an end-point of W, then $d_{W_i,v} \geq \bar{d}_{W_i,v}$.

We first prove our claim for the case of $1 < i < a$. By construction, in this case, W_i is a subsequence of W, i.e., $W_i = ((s_x, e_x), \dots, (s_y, e_y))$ for $1 < x < y < k$, and as a result it is an alternating multi-walk. We will show that in this case, multi-walk W is applicable with respect to any subset of V. Based on Lemma 5.4.6, to get this, it suffices to show that $e_x \in M_{s_x}$ and $e_y \in M_{s_y}$ hold. Since W_i is a result of decomposing W, we know that elements (s_{x-1}, e_{x-1}) and (s_{y+1}, e_{y+1}) are removed in Line 15 of the algorithm. As a result we have $e_{x-1} \notin M_{s_{x-1}}$ and $e_{y-1} \notin M_{s_{y-1}}$. Combining this with the fact that W in alternating, we get $e_x \in M_{s_{x-1}}$ and $e_y \in M_{s_y}$.

To complete the proof, it remains to show that for any $i \in \{1, a\}$, multi-walk W_i is alternating, and that any vertex v which is an end-point of W_i satisfies $d_{W_i,v} \geq \bar{d}_{W_i,v}$. For any $i \in [k]$, let $e_i = (u_i, u_{i+1})$ which means that for any $i > 1$, we have $u_i \in e_{i-1}$ and $u_i \in e_i$. Consider the multi-walks W_1 and W_a in Line 15 of the algorithm. We assume w.l.o.g. that during the decomposing of W to shorter multi-walks, it is decomposed to at least two multi-walks and as a result $1 < a$. At this point of the algorithm, we have $W_1 = ((s_1, e_1), \dots, (s_x, e_x))$ and $W_a = ((s_y, e_y), \dots, (s_k, e_k))$ for some $1 \leq x < y \leq k$. Note that both W_1 and W_k are alternating multi-walks due to being subsequences of W. Moreover, similar to the previous case, we can argue that $e_x \in M_{s_x}$ and $e_y \in M_{s_y}$ due to the fact that elements (s_{x+1}, e_{x+1}) and (s_{y+1}, e_{y+1}) are removed during the decomposition process. If we also have $e_1 \in M_{s_x}$ and $e_k \in M_{s_x}$ then W_1 is not expandable by W_a and both these multi-walks are applicable with respect to any set of vertices due to the third item of Lemma 5.4.6. Therefore, we focus on the case that either $e_k \notin M_{s_x}$ or $e_1 \notin M_{s_x}$ holds. Let us assume w.l.o.g. that we have $e_k \notin M_{s_x}$. It is easy to see that if $u_1 \notin V_s$ then W_1 is applicable with respect to V_s. We claim that in this case of $e_k \notin M_{s_x}$, if $u_1 \in V_s$, then we have $u_1 = u_{k+1}$ and $e_1 \in M_{s_x}$ as otherwise W does meet the condition $d_{W_i,v} \geq \bar{d}_{W_i,v}$ which is necessary for W being applicable with respect to set V_s. This implies that W_1 is expandable by W_a since $((s_1, e_1), \ldots, (s_x, e_x), (s_y, e_y), \ldots, (s_k, e_k))$ is an alternating multi-walk. As a result to complete the proof we only need to show that the result of expanding W_1 by W_a is applicable with respect to V_s. Indeed in this case, this multi-walk is applicable with respect to any set of vertices due to $e_x \in M_{s_x}$ and $e_y \in M_{s_y}$ and the third item of Lemma 5.4.6. Thus, the proof of the this lemma is concluded. □

Lemma 5.4.11. *The while loop in Line 6 of Algorithm 5 terminates and $\mathcal{W}$ constructed by that is a set of alternating multi-walks.*

Proof. It is easy to see that if the loop terminates $\mathcal{W}$ only contains alternating multi-walks since any multi-walk W added to this set is the result of iteratively expanding an empty multi-walk by a set of paths and cycles. Recall that by definition, an empty multi-walk is alternating and the result of expanding an alternating multi-walk by a path or a cycle is also an alternating multi-walk. The while loop terminates when for any $i \in [\alpha]$, we have $E'_i = \emptyset$, thus to complete the proof, it suffices to show that each iteration of the loop terminates and that in each one, we remove at least one edge from one of the subgraphs $\mathcal{G}'_1, \ldots, \mathcal{G}'_\alpha$. We consider an arbitrary iteration of the loop, and show that in Line 9, edges of p are removed from $\mathcal{G}'_i$. This happens iff W is expandable by p. Multi-walk W is empty at this point of the algorithm (and as a result is an alternating multi-walk) and $p = (e_1, \ldots, e_k)$ is a maximal (nonempty) path or a cycle chosen from an arbitrary $\mathcal{G}'_i$ in Line 8. As an application of Lemma 5.4.12, we get that W is expandable by p. As a result of this, in Line 9 of the algorithm edges of p are removed from E'_i. To conclude that the while loop terminates we also have to show that each of its iterations terminate. It is easy to see since the loop nesting in this while loop obviously terminates as well. □

Lemma 5.4.12. *Let $p = (e'_1, \ldots, e'_b)$ be a a maximal connected-component (a path or a cycle) in graph ${\mathcal{G}'}_i$ (defined in Algorithm 5), and let $W = ((s_1, e_1), \ldots, (s_a, e_a))$ be an alternating multi-walk of profile $P' = ((\mathcal{G}_1, M_1), \ldots, (\mathcal{G}_\alpha, M_\alpha))$, such that for any $j \in [b]$, we have $(i, e'_j) \notin W$ and for any $j \in [a]$, we have $e_j \in E'_{s_j}$. If the first vertex of W is the same as the last vertex of p and $\mathbf{1}(e_1 \in M_{s_1}) + \mathbf{1}(e'_b \in M_i) = 1$, then W is expandable by p.*

Proof. First, let us note that any maximal connected-component in graph ${\mathcal{G}'}_i$ is a path or a cycle since we have $E'_i \subset (M_i \cup M_i^{\mathcal{A}})$, and as a result the degree of each vertex in $\mathcal{G}'_i$ is at most two. (Recall that, M_i and $M_i^{\mathcal{A}}$ are both matchings of graph $\mathcal{G}_i$.) To prove that W is expandable by p we will show that $W_p = ((i, e'_1), \ldots, (i, e'_b), (s_1, e_1), \ldots, (s_a, e_a))$ is an alternating multi-walk. First, W_p is a multi-walk since $(e'_1, \ldots, e'_b, e_1, \ldots, e_a)$ is a walk in G and it also contains distinct elements as for any $j \in [b]$, $(i, e'_j) \notin W$ holds.

By Definition 5.4.3, to prove that W_p is alternating, we first need to show that for any two consecutive elements in W_p, e.g., (s''_1, e''_1) and (s''_2, e''_2), we have $\mathbf{1}(e''_1 \in M_{s''_1}) + \mathbf{1}(e''_2 \in M_{s''_2}) = 1$. If both these elements are in W this simply holds due to W being an alternating multi-walk itself. Moreover, if exactly one of them is in W, we get this as a result of $\mathbf{1}(e_1 \in M_{s_1}) + \mathbf{1}(e'_b \in M_i) = 1$ (in the statement of lemma). Therefore, we need to focus on showing that for any $j \in [b-1]$, we have $\mathbf{1}(e_j \in M_i) + \mathbf{1}(e_{j+1} \in M_i) = 1$. Since $E'_i \subset (M_i \cup M_i^{\mathcal{A}})$ and by the fact that M_i and $M_i^{\mathcal{A}}$ are matchings of graph $\mathcal{G}_i$, if $e_i \in M_i$ then $e_{i+1} \notin M_i$. Similarly, if $e_i \notin M_i$ then $e_i \in M_i^{\mathcal{A}}$ which gives us $e_{i+1} \notin M_i^{\mathcal{A}}$ and $e_{i+1} \in M_i$.

As the second condition in Definition 5.4.3, we need to show that $P \Delta W_p = ((\mathcal{G}_1, M'_1), \ldots, (\mathcal{G}_k, M'_k))$ is a profile, where for any $j \in [\alpha]$ we have

$$M'_j = M_j \cup \{e \,|\, (j, e) \in W_p \text{ and } e \notin M_j\} \backslash \{e \,|\, (j, e) \in W_p \text{ and } e \in M_j\}. \tag{5.34}$$

By Definition 5.4.1, to prove that $P \Delta W_p$ is a profile, it only suffices to show that for any $j \in [\alpha]$, M'_j is a matching in $\mathcal{G}_j$. This simply holds for any $j \neq i$ due to W being an alternating multi-walk itself, thus we only need to show that M_i is a matching in $\mathcal{G}_i$. To achieve this, we consider any two edges $\{e, e'\} \subset M'_i$ and show that e and e' are not adjacent in $\mathcal{G}_i$. If neither one of these edges is in p, then for W to be an alternating multi-walk these edges cannot be adjacent. Moreover, it is easy to see that if both edges are in p, they are not adjacent either. Thus, we assume that exactly one of the edges is in p. W.l.o.g., we assume $e \in p$ and $e' \notin p$. We consider two cases of $e' \in G'_i$ and $e' \notin G'_i$. In the first case, e and e' are not adjacent since p is a maximal component of G'_i and as a result is not connected to edges that are not in p (including e'). In the case of $e' \notin G'_i$, we claim that e' is in both M_i and $M_i^{\mathcal{A}}$ which means it cannot be adjacent to any edge in G'_i including e. To prove this claim, note that by Equation 5.34, we have $M'_i \subset (M_i \cup \{e'' \,|\, (i, e'') \in W_p\})$ and by the statement of lemma for any $(i, e'') \in W_p$ we have $e'' \in E'_i$. Moreover, by definition of $\mathcal{G}'_i$, we know $E'_i \subset (M_i \cup M_i^{\mathcal{A}})$. Putting these facts together results in the following equation:

$$M'_i \subset (M_i \cup \{e'' \,|\, (i, e'') \in W_p\}) \subset (M_i \cup E'_i) \subset (M_i \cup M_i^{\mathcal{A}}).$$

Recall that $\mathcal{G}'_i$ contains an edge iff it is in $(M_s \cup M_a^{\mathcal{A}})$ but not in $(M_s \cap M_a^{\mathcal{A}})$. As a result since e is in M'_i but it is not in $\mathcal{G}'_i$, then it is in $(M_s \cap M_a^{\mathcal{A}})$. This completes the proof of our lemma since we obtained that W_p is an alternating multi-walk. □

Claim 5.4.13. *In Line 5 of Agorithm 5, for any $v \in V_s$, we have $r_v \geq g_v$ where $g_v = |\{i : v \in (M_i^{\mathcal{A}} \cap E'_i)\}|$ and $r_v = |\{i : v \in (M_i \cap E'_i)\}|$.*

Proof. We use proof by contradiction. Let $v \in V_s$ be a vertex with $r_v < g_v$. It is easy to see that we have $v \notin V_r$ since in Line 4, for any $i \in [\alpha]$, we remove any edge in E'_i which has at least one end-point in V_r. As a result, in Line 5, for any $u \in V_r$ we have $d_{v,g} = 0$. Due to $v \notin V_r$, we get $|\{i : v \in M_i^{\mathcal{A}}\}| \leq |\{i : v \in M_i\}|$. Observe that for any $v \notin V_r$, we have

$$|\{i : v \in (M_i^{\mathcal{A}} \cap E'_i)\}| = |\{i : v \in M_i^{\mathcal{A}}\}| - |\{i : v \in (M_i^{\mathcal{A}} \cap M_i)\}|, \text{ and}$$

$$|\{i : v \in (M_i \cap E'_i)\}| = |\{i : v \in M_i\}| - |\{i : v \in (M_i^{\mathcal{A}} \cap M_i)\}|.$$

This gives us $r_v - g_v = |\{i : v \in M_i\}| - |\{i : v \in M_i^{\mathcal{A}}\}|$, which implies $r_v \geq g_v$ and completes our proof. $\square$

Lemma 5.4.14. *Any multi-walk in line 17 of Algorithm 5 which is represented by an edge in hyper-graph H' is applicable with respect to the vertices in V_s.*

Proof. By Lemma 5.4.10, to prove this, it suffices to show that any $W \in \mathcal{W}$ constructed in the algorithm is applicable with respect to V_s. Recall that, by Definition 5.4.4, a multi-walk W of profile P is applicable with respect to V_s iff it is alternating and it satisfies $d_{W,v} \geq \bar{d}_{W,v}$ for any $v \in V_s$. Based on Lemma 5.4.11, W is an alternating multi-walk thus it remains to show that for any $v \in V_s$, we have $d_{W,v} \geq \bar{d}_{W,v}$.

We use proof by contradiction. We start by assuming that there exists a vertex $v \in V_s$ and a multi-walk $W' \in \mathcal{W}$ where $d_{W',v} < \bar{d}_{W',v}$ and then show that it results in a contradiction. Let $W = ((s_1, e_1), \ldots, (s_k, e_k))$ be the first multi-walk for which we have $d_{W,v} \neq \bar{d}_{W,v}$. By Lemma 5.4.6, this implies that vertex v is an endpoint of this multi-walk. W.l.o.g., let us assume that we have $e_1 = (v, u_2)$. Consider subgraphs $\mathcal{G}'_1, \ldots, \mathcal{G}'_\alpha$ in the algorithm when W is added to $\mathcal{W}$. Due to the condition of the while loop in Line 6 of the algorithm the following holds at this point of the algorithm. There does not exist a $\mathcal{G}'_i$ that contains a maximal path p with which W is expandable. By Lemma 5.4.12, this implies that any maximal path $p = (e'_1, \ldots, e'_a)$ in any subgraph $\mathcal{G}'_i$ that ends in vertex v (i.e., $e'_a = (u'_a, v)$) satisfies $\mathbf{1}(e_1 \in M_{s_1}) = \mathbf{1}(e'_a \in M_i)$. We consider both cases of $e_1 \in M_{s_1}$ and $e_1 \in M_{s_1}$ and prove prove the lemma for each one independently.

Let us assume that $e_1 \in M_{s_1}$. In this case, by Item 2 of Lemma 5.4.6, we have

$d_{W,v} \geq \bar{d}_{W,v}$ which means $W \neq W'$. We will show that in this case, any multi-walk W'' added to set $\mathcal{W}$ in the next iterations satisfies $d_{W'',v} \geq \bar{d}_{W'',v}$ which contradicts the existence of W'. Consider a maximal connected component (a path or a cycle) $p = (e'_1, \ldots, e'_a)$ in $\mathcal{G}_i$ for an arbitrary $i \in [\alpha]$, and define $W_p = ((i, e'_1), \ldots, (i, e'_a))$. By Lemma 5.4.12 W_p is an alternating multi-walk. Moreover, by Item 1 of Lemma 5.4.6 if v is not an end-point of p (which also includes the case that p is a cycle) then we have $d_{W,v} = \bar{d}_{W_p,v}$. Further, if p is a path and v is one of its end-points, i.e., $e'_a = (u'_a, v)$, as mentioned above we have $\mathbf{1}(e_1 \in M_{s_1}) = \mathbf{1}(e'_a \in M_i)$, which means $e'_a \in M_i$. As a result of this and by invoking the second item of Lemma 5.4.6, we get that $d_{W_p,v} \geq \bar{d}_{W_p,v}$. Note that any multi-walk W'' constructed in the next iterations consists of a set of maximal connected components. Since all the remaining connected components satisfy $d_{W_p,v} \geq \bar{d}_{W_p,v}$, we also have $d_{W'',v} \geq \bar{d}_{W'',v}$. This contradicts the existence of multi-walk W' with $d_{W',v} < \bar{d}_{W',v}$.

Now we consider the case of $e_1 \notin M_{s_1}$. We will show that this assumption results in equation $|\{i : v \in (M_i^{\mathcal{A}} \cap E'_i)\}| < |\{i : v \in (M_i \cap E'_i)\}|$ for vertex v, which contradicts the statement of Claim 5.4.13. First, we show that if $e_1 \notin M_{s_1}$ then any multi-walk $W'' \in \mathcal{W}$ satisfies $d_{W'',v} \leq \bar{d}_{W'',v}$. Let us consider a path or cycle $p = (e'_1, \ldots, e'_a)$ in graph $\mathcal{G}_i$ for an arbitrary $i \in [\alpha]$, and define $W_p = ((i, e'_1), \ldots, (i, e'_a))$. Similar to what we used in the proof of the previous case, if v is not an end-point of W_p (which also includes the case of p being a cycle), then by Lemma 5.4.6, we have $d_{W_p,v} \geq \bar{d}_{W_p,v}$. Moreover, if p is a path and v is an end-point in this path, i.e., $e'_a = (u'_a, v)$, we have $\mathbf{1}(e_1 \in M_{s_1}) = \mathbf{1}(e'_a \in M_i)$. Since in this case we have $e_1 \notin M_{s_1}$, we get $e'_a \notin M_i$. As a result of this, Item 2 in Lemma 5.4.6 gives us $d_{W_p,v} \leq \bar{d}_{W_p,v}$. Based on an argument that we used for the previous case, this implies that any mutli-walk W'' that we add to $\mathcal{W}$ in the next iterations satisfies $d_{W'',v} \geq \bar{d}_{W'',v}$. Moreover, due to the assumption that W is the first multi-walk that for any W'' that is added to this set before W we have $d_{W'',v} = \bar{d}_{W'',v}$. We also have $d_{W,v} < \bar{d}_{W,v}$ as a result of assumption $e_1 \notin M_{s_1}$ and the second item of Lemma 5.4.6. This gives us the following equation:

$$\sum_{W \in \mathcal{W}} (\bar{d}_{W,v} - d_{W,v}) = \sum_{W \in \mathcal{W}} (|\{(i,e) \in W : v \in e, e \notin M_i\}| - |\{(i,e) : v \in e, e \in M_i\}|) > 0. \tag{5.35}$$

where the first equality is due to the definition of $\bar{d}_{W,v}$ and $d_{W,v}$. Further, based on Lemma 5.4.11, we know that the while loop in Line 6 of Algorithm 5 terminates. When this loop terminates, there is no $j \in [\alpha]$ where $\mathcal{G}'_j$ contains at least one edge. This means that for any $e \in E'_j$ element (e,i) is in exactly one of the multi-walks in $\mathcal{W}$. Also, note that by construction, $E'_j \subset (M^{\mathcal{A}}_j \cup M_j)$. As a results we get the following equations for vertex v:

$$|\{i : v \in (M_i \cap E'_i)\}| = \sum_{W \in \mathcal{W}} |\{(i,e) \in W : v \in e, e \in M_i\}|, \text{ and}$$

$$|\{i : v \in (M^{\mathcal{A}}_i \cap E'_i)\}| = \sum_{W \in \mathcal{W}} |\{(i,e) \in W : v \in e, e \notin M_i\}|.$$

Combining this with Equation 5.35, we get:

$$|\{i : v \in (M^{\mathcal{A}}_i \cap E'_i)\}| - |\{i : v \in (M_i \cap E'_i)\}| > 0$$

which is in contradiction with the following equation by Claim 5.4.13 for any $v \in V_s$:

$$|\{i : v \in (M_i^{\mathcal{A}} \cap E_i')\}| \leq |\{i : v \in (M_i \cap E_i')\}|.$$

□

Lemma 5.4.15. *Any weighted hyper-graph $K = (G, E_K)$ of max-degree Δ and rank r has a matching with weight at least $\frac{1}{r\Delta} \sum_{e \in E_K} w(e)$.*

Proof. We construct a matching M_K using an iterative greedy algorithm and show that its weight is at least $\frac{1}{2\Delta} \sum_{e \in E_K} w(e)$. At the beginning all the edges are alive. In each iteration, we add an edge e to M_K which has the maximum weight among the alive edges and kill all its neighboring edges (that are not already killed by another vertex). Note that each edge e in M_K kills at most $r\Delta - 1$ other edges with weight smaller than $w(e)$, which means $\sum_{e \in M_K} w(e) \geq \frac{1}{r\Delta} \sum_{e \in E_K} w(e)$.

□

Lemma 5.4.16. *Given that an edge $e = (u_1, u_2)$ exists in $M_i^{\mathcal{A}}$ defined in Algorithm 5, probability of this edge being removed in Line 4 of the algorithm is upper-bounded by ε^3.*

Proof. Note that $e = (u_1, u_2)$ is removed in Line 4 of the algorithm iff $e \in M_i^{\mathcal{A}}$ and there exists a vertex $v \in \{u_1, u_2\}$ which is saturated and satisfies $|\{j : v \in M_j^{\mathcal{A}}\}| < |\{j : v \in M_j\}|$. Let I_e be an indicator random variable for the event of e being removed from $\mathcal{G}_i'$ in Line 4 of the algorithm. Moreover, let us define $g_v := |\{j : v \in M_j^{\mathcal{A}}\}|$ and

$r_v := |\{j : v \in M_j\}|$. We have

$$\Pr[I_e] \leq \Pr[g_{u_1} > r_{u_1} \mid u_1 \in M_i^{\mathcal{A}}] + \Pr[g_{u_2} > r_{u_2} \mid u_2 \in M_i^{\mathcal{A}}]. \tag{5.36}$$

Thus, it suffices to show that, $\Pr[g_v > r_v \mid v \in M_i^{\mathcal{A}}] \leq \varepsilon^3/2$ holds for any vertex $v \in \{u_1, u_2\}$. We have

$$\Pr[g_v > r_v \mid v \in M_i^{\mathcal{A}}] \leq \Pr[g_{v,-i} + 1 > r_{v,-i}] \leq \Pr[g_{v,-i} \geq r_{v,-i}] \tag{5.37}$$

where $g_{v,-i} := |\{j : j \neq i \text{ and } v \in M_j^{\mathcal{A}}\}|$ and $r_{v,-i} := |\{j : j \neq i \text{ and } v \in M_j\}|$. Recall that by definition of saturated vertices in Line 5 of Algorithm 4, for any saturated vertex v and $i \in [\alpha]$, we have $\Pr[v \in M_i] - \Pr[v \in M_i^{\mathcal{A}}] \geq \varepsilon^3 - 1/\alpha$ and as a result $\mathbb{E}[r_{v,-i}] - \mathbb{E}[g_{v,-i}] \geq (\alpha - 1)(\varepsilon^3 - 1/\alpha)$. To complete the proof, we show

$$\Pr[|g_{v,-i} - \mathbb{E}[g_{v,-i}]| > (\alpha - 1)\varepsilon^4] \leq \varepsilon^{-4}, \text{ and } \Pr[|r_v - \mathbb{E}[r_v]| > (\alpha - 1)\varepsilon^4] \leq \varepsilon^{-4}.$$

Note that $g_{v,-i}$ and $r_{v,-i}$ are both sum of independent Bernoulli random variables as for any a and b, $\mathcal{G}'_a$ and $\mathcal{G}'_b$ are independent random variables. Therefore, to bound $\Pr[|g_{v,-i} - \mathbb{E}[g_{v,-i}]| > (\alpha - 1)\varepsilon^4]$ and $\Pr[|r_{v,-i} - \mathbb{E}[r_{v,-i}]| > (\alpha - 1)\varepsilon^4]$ we can use Chebyshev's inequality which states for any k, $\Pr[|r_{v,-i} - \mathbb{E}[r_{v,-i}]| > \mathrm{Var}(r_v)^{1/2}k] \leq k^{-2}$. Observe that $\mathrm{Var}(r_{v,-i}) < (\alpha - 1)$ and $\mathrm{Var}(g_{v,-i}) < (\alpha - 1)$. Based on Algorithm 4, we have $\alpha - 1 = \varepsilon^{-12}$. This implies that

$$\begin{aligned}
\Pr[|r_{v,-i} - \mathbb{E}[r_{v,-i}]| > (\alpha-1)\varepsilon^4] &= \Pr[|r_{v,-i} - \mathbb{E}[r_{v,-i}]| > \varepsilon^{-8}] \\
&= \Pr[|r_{v,-i} - \mathbb{E}[r_{v,-i}]| > (\alpha-1)^{1/2}\varepsilon^{-2}]] \\
&\leq \Pr[|r_{v,-i} - \mathbb{E}[r_{v,-i}]| \geq \operatorname{Var}(r_v)^{-1/2}\varepsilon^{-2}]] \\
&\leq \varepsilon^4.
\end{aligned}$$

We can similarly show that $\Pr[|g_{v,-i} - \mathbb{E}[g_{v,-i}]| > (\alpha-1)\varepsilon^4] \leq \varepsilon^4$. Moreover, since $\mathbb{E}[r_{v,-i}] - \mathbb{E}[g_{v,-i}] \geq (\alpha-1)(\varepsilon^3 - 1/\alpha)$, if $g_{v,-i} \geq r_{v,-i}$ then, we either have $g_{v,-i} \geq \mathbb{E}[g_{v,-i}] + (\alpha-1)(\varepsilon^3 - 1/\alpha)/2$ or $r_{v,-i} \leq \mathbb{E}[r_{v,-i}] - (\alpha-1)(\varepsilon^3 - 1/\alpha)/2$. For a small enough ε, we have $(\varepsilon^3 - 1/\alpha)/2 \geq \varepsilon^4$, and

$$\Pr[g_{v,-i} \geq r_{v,-i}] \leq \Pr[|r_{v,-i}-\mathbb{E}[r_{v,-i}]| > (\alpha-1)\varepsilon^4]+\Pr[|g_{v,-i}-\mathbb{E}[g_{v,-i}]| > (\alpha-1)\varepsilon^4] \leq 2\varepsilon^4.$$

Combining this with Equation 5.37 and Equation 5.36 results in $\Pr[I_e] \leq 4\varepsilon^4$ which for a small enough ε, gives us $\Pr[I_e] \leq \varepsilon^3$. □

5.4.5 Lemma 5.3.4 Property 3: Independence

In this section our goal is to prove the following lemma.

Lemma 5.4.17. *For any $0 \leq r \leq t$, it is possible to simulate algorithm $\mathcal{B}_r(\mathcal{G})$ in $O(\varepsilon^{-24} \log \Delta \operatorname{poly}(\log\log \Delta))$ rounds of* LOCAL.

Proof. We will show that for any $r \leq t$, algorithm $\mathcal{B}_r(\mathcal{G})$ can be implemented in

$$x_r := cr\varepsilon^{-4} \log \Delta \operatorname{poly}(\log \log \Delta)$$

rounds of LOCAL for a large enough constant c. Since we have $t = c_t \varepsilon^{-20}$ for a constant c_t, this implies that $\mathcal{B}(\mathcal{G}) = \mathcal{B}_t(\mathcal{G})$ can be simulated in $O(\varepsilon^{-24} \log \Delta \operatorname{poly}(\log \log \Delta))$ rounds. To prove this claim, we use proof by induction. As the base case, $\mathcal{B}_0(\mathcal{G})$ can be simply implemented in $O(1)$ rounds as it only returns an empty matching. As the induction step, for any $r > 1$, we assume that our claim holds for $\mathcal{B}_{r-1}(\mathcal{G})$, and prove that it holds for $\mathcal{B}_r(\mathcal{G})$ too.

Graph G is the underlying graph in our LOCAL simulation of $\mathcal{B}_r(\mathcal{G})$, and there is a processor on each $v \in V$. The initial information that each node v holds is as follows. Its incident neighbors in graphs G and $\mathcal{G}$, $\Pr_{\mathcal{G} \sim G, \mathcal{A}}[v \in \mathcal{A}(\mathcal{G})]$, and parameters ε, r and Δ (maximum degree of G). Observe that other than $\mathcal{G}$, the rest of the initial information is independent of the realization of $\mathcal{G}$ and the randomization of the algorithm. Thus, if two vertices are not adjacent in G, they initially do not share any information that is correlated with the randomization of the algorithm or the realization of $\mathcal{G}$. As a result, to prove our lemma, we only need to show that using this initialization, we can implement our algorithm in the desired number of rounds. To prove our claim, we go over Algorithm 4 line by line, and investigate the number of rounds that we need to simulate each one in the LOCAL model. The first two lines obviously take $O(1)$ round since no communication is needed for initializing the variables.

In Line 3 and Line 4 of the algorithm, the goal is to construct profile P. First, to construct subgraphs $\mathcal{G}_2, \ldots, \mathcal{G}_\alpha$, for any edge $e \in G$, we only need its end-points to communicate and hold the information about realization of e in these subgraphs. This can be done in $O(1)$. Moreover, by the induction step for any i, algorithm $\mathcal{B}_{r-1}(\mathcal{G}_\rangle)$ can be simulated in x_{r-1} rounds. Further, $\mathcal{B}_{r-1}(\mathcal{G}_1) \ldots, \mathcal{B}_{r-1}(\mathcal{G}_\alpha)$, can be constructed in parallel. As a result this line of the algorithm takes $x_{r-1} + O(1)$ rounds.

To simulate Line 5 of the algorithm, we show that any vertex v can compute $\Pr_{\mathcal{G}' \sim G, \mathcal{A}}[v \in \mathcal{B}_{r-1}(\mathcal{G}')]$ and determine whether it is saturated or not after x_{r-1} rounds of the algorithm. First, note that $\Pr_{\mathcal{G}' \sim G, \mathcal{A}}[v \in \mathcal{B}_{r-1}(\mathcal{G}')]$ is just a function of G. Moreover, by the induction step, $\mathcal{B}_{r-1}(\mathcal{G}')$ can be implemented in x_{r-1} rounds of LOCAL, which implies that $\Pr_{\mathcal{G}' \sim G, \mathcal{A}}[v \in \mathcal{B}_{r-1}(\mathcal{G}')]$ is a function of x_{r-1}-hop of vertex v in graph G. This is a piece of information that vertex v can gather in x_{r-1} rounds. Therefore, considering that initially each vertex holds the value of $\Pr_{\mathcal{G} \sim G, \mathcal{A}}[v \in \mathcal{A}(\mathcal{G})]$ and ε, vertex v can determine whether it is saturated or not by evaluating the following inequality.

$$\Pr_{\mathcal{G} \sim G, \mathcal{B}}[v \in Z_{r-1}] \leq \Pr_{\mathcal{G} \sim G, \mathcal{A}}[v \in \mathcal{A}(\mathcal{G})] + \varepsilon^3 - 1/\alpha.$$

This only adds an extra $O(1)$ to the round complexity of the LOCAL algorithm since each vertex can gather the necessary information during the $x_{r-1} + O(1)$ that our algorithm has already run from the beginning of the algorithm.

In Line 6 and Line 7, the goal is to construct the hyper-graph H, which has a hyper-edge between the vertices of any multi-walk of length at most $l = 3\varepsilon^{-3}$ of P in set $\mathcal{W}_a$. Recall that $\mathcal{W}_a$ is the set of alternating multi-walks of P that are applicable with respect to the set of saturated vertices. To achieve this, first, each vertex gathers all the information about the vertices in its l-hop and finds the alternating multi-walks of length at most l that contain this vertex. In this way, each vertex knows all the edges of H to which it belongs. This can obviously be done in $O(l)$ rounds.

Line 8 of the algorithm is about $\mathsf{ApproxMatching}(H)$ which as mentioned before uses an algorithm by Harris [30] stated below.

Proposition 5.4.18 ([30, Theorem 1.2])**.** *Given a hyper-graph of rank r and a constant $\delta \in (0, 1/2)$, there is an $\tilde{O}(\log \Delta + r)$-round algorithm in the* LOCAL *model to get an $O(r)$-approximation to maximum weight matching with probability at least $1 - 1/\delta$. Here the $\tilde{O}$ notation hides* $\text{poly} \log \log \Delta$ *and* $\text{poly} \log r$ *factors.*

Based on this proposition, to analyze the round complexity of $\mathsf{ApproxMatching}(H)$, we first need to give an upper-bound for the maximum degree of H which is the maximum number of hyper-edges in H that any single vertex v can belong to. In hyper-graph H, we have a hyper-edge between the vertices of any alternating hyper-walk $w = ((s_1, e_1), \dots, (s_k, e_k))$ of length at most l in profile P. By definition of multi-walks, $p = (e_1, \dots, e_k)$ should be a walk in graph G. In a graph of maximum degree Δ, there are at most $l\Delta^l$ distinct walks of length at most l that contain vertex v. Further, for any $i \in [k]$, we have $s_i \in [\alpha]$ which means that there are at most α possible choices for any s_i. Thus, in graph H, there are at most $l(\Delta\alpha)^l$ edges that contain any arbitrary vertex v, and as a result maximum degree of H is upper-bounded by $l(\Delta\alpha)^l$. Moreover, rank of hyper-graph H is simply upper-bounded by l since the rank of a hyper-graph is the maximum number of vertices that any edge contains. In the case of graph H this is bounded by l since each edge is between vertices of a walk of length at most l. Putting these together, and plugging in the value of variables $l = 3\varepsilon^{-3}$ and $\alpha = \varepsilon^{-12} + 1$, we obtain the following upper-bound for the round complexity of $\mathsf{ApproxMatching}(H)$:

$$\tilde{O}(\log{(\Delta\alpha)}^{2l} + l) = O(l\log{(\Delta\alpha)}\,\mathrm{polylog}\,(l)\,\mathrm{poly}\log\log{(l(\Delta\alpha)^l)})$$
$$= O(\varepsilon^{-4}\log{(\Delta)}\,\mathrm{poly}\log\log{(\Delta)}).$$

We can set the constant c in a way that the number of rounds needed here is upper-bounded by $c\varepsilon^{-4}\log{(\Delta)}\,\mathrm{poly}\log\log{(\Delta)}/2$.

Finally, in Line 11 we need to apply a set of multi-walks of length at most l (constructed in previous rounds) on profile P. This can be easily done in $O(\varepsilon^{-3})$-rounds since we have $l = 3\varepsilon^{-3}$. To sum up, The overall round complexity of the algorithm which we denote by R_r is as follows:

$$\begin{aligned}
R_r =& O(1) + x_{r-1} + O(l) + O(1) + O(l) + c\varepsilon^{-4}\log{(\Delta)}\,\mathrm{poly}\log\log{(\Delta)}/2 \\
=& c(r-1)\varepsilon^{-4}\log\Delta\,\mathrm{poly}(\log\log\Delta) + O(\varepsilon^{-4}) + c\varepsilon^{-4}\log\Delta\,\mathrm{poly}(\log\log\Delta) \\
=& x_r + O(\varepsilon^{-4}) - c\varepsilon^{-4}\log\Delta\,\mathrm{poly}(\log\log\Delta)/2.
\end{aligned}$$

Let $c_0\varepsilon^{-4}$ be an upper-bound for what we denote in our round complexity as $O(\varepsilon^{-4})$ where c_0 is constant. We can set the constant c to be large enough to satisfy

$$c_0\varepsilon^{-4} - c\varepsilon^{-4}\log\Delta\,\mathrm{poly}(\log\log\Delta)/2 \leq 0.$$

This gives us $R_r \leq x_r$, and concludes our proof. □

Bibliography

[1] Amine Allouah and Omar Besbes. Prior-independent optimal auctions. In *Proceedings of the 2018 ACM Conference on Economics and Computation, Ithaca, NY, USA, June 18-22, 2018*, page 503, 2018.

[2] Noga Alon, Ankur Moitra, and Benny Sudakov. Nearly Complete Graphs Decomposable into Large Induced Matchings and Their Applications. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 1079–1090, 2012.

[3] Sepehr Assadi, MohammadHossein Bateni, Aaron Bernstein, Vahab S. Mirrokni, and Cliff Stein. Coresets meet EDCS: algorithms for matching and vertex cover on massive graphs. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1616–1635, 2019.

[4] Sepehr Assadi and Aaron Bernstein. Towards a Unified Theory of Sparsification for Matching Problems. In *2nd Symposium on Simplicity in Algorithms, SOSA@SODA 2019, January 8-9, 2019 - San Diego, CA, USA*, pages 11:1–11:20, 2019.

[5] 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.

[6] 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.

[7] Lawrence M Ausubel, Paul Milgrom, et al. The lovely but lonely vickrey auction. *Combinatorial auctions*, 17:22–26, 2006.

[8] Soheil Behnezhad and Mahsa Derakhshan. Stochastic Weighted Matching: $(1-\varepsilon)$ Approximation. In *61st IEEE Annual Symposium on Foundations of Computer*

Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020, pages 1392–1403. IEEE, 2020.

[9] Soheil Behnezhad, Mahsa Derakhshan, Alireza Farhadi, MohammadTaghi Hajiaghayi, and Nima Reyhani. Stochastic Matching on Uniformly Sparse Graphs. In *Algorithmic Game Theory - 12th International Symposium, SAGT 2019, Athens, Greece, September 30 - October 3, 2019, Proceedings*, pages 357–373, 2019.

[10] Soheil Behnezhad, Mahsa Derakhshan, and MohammadTaghi Hajiaghayi. Stochastic Matching with Few Queries: $(1-\varepsilon)$ Approximation. In *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.

[11] 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.

[12] 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.

[13] Dimitri P Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 1999.

[14] Hedyeh Beyhaghi, Negin Golrezaei, Renato Paes Leme, Martin Pal, and Balasubramanian Siva. Improved approximations for free-order prophets and second-price auctions. *arXiv preprint arXiv:1807.03435*, 2018.

[15] Anand Bhalgat, Jon Feldman, and Vahab S. Mirrokni. Online allocation of display ads with smooth delivery. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, pages 1213–1221. ACM, 2012.

[16] Avrim Blum, Ioannis Caragiannis, Nika Haghtalab, Ariel D. Procaccia, Eviatar B. Procaccia, and Rohit Vaish. Opting into optimal matchings. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 2351–2363, 2017.

[17] 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.

[18] 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. *Operations Research*, 68(1):16–34, 2020.

[19] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities - A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

[20] L. Elisa Celis, Gregory Lewis, Markus M. Mobius, and Hamid Nazerzadeh. Buy-it-now or take-a-chance: Price discrimination through randomized auctions. *Manag. Sci.*, 60(12):2927–2948, 2014.

[21] Mahsa Derakhshan, Negin Golrezaei, and Renato Paes Leme. LP-based Approximation for Personalized Reserve Prices. In *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019*, page 589. ACM, 2019.

[22] Mahsa Derakhshan, David M. Pennock, and Aleksandrs Slivkins. Beating greedy for approximating reserve prices in multi-unit VCG auctions. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 1099–1118. SIAM, 2021.

[23] Peerapong Dhangwatnotai, Tim Roughgarden, and Qiqi Yan. Revenue maximization with a single sample. *Games Econ. Behav.*, 91:318–333, 2015.

[24] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American economic review*, 97(1):242–259, 2007.

[25] Jack Edmonds. Maximum matching and a polyhedron with 0, 1-vertices. *Journal of research of the National Bureau of Standards B*, 69(125-130):55–56, 1965.

[26] Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. Monotonicity testing over general poset domains. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 474–483, 2002.

[27] Mohsen Ghaffari. Distributed Maximal Independent Set using Small Messages. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 805–820, 2019.

[28] Ashish Goel, Michael Kapralov, and Sanjeev Khanna. On the communication and streaming complexity of maximum bipartite matching. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 468–485, 2012.

[29] Negin Golrezaei, Max Lin, Vahab Mirrokni, and Hamid Nazerzadeh. Boosted second price auctions: Revenue optimization for heterogeneous bidders. 2017.

[30] David G. Harris. Distributed local approximation algorithms for maximum matching in graphs and hypergraphs. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 700–724. IEEE Computer Society, 2019.

[31] Jason D. Hartline and Tim Roughgarden. Simple versus optimal mechanisms. *SIGecom Exch.*, 8(1), 2009.

[32] Michael Kapralov. Better bounds for matchings in the streaming model. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1679–1697, 2013.

[33] Renato Paes Leme, Martin Pál, and Sergei Vassilvitskii. A field guide to personalized reserve prices. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 1093–1102. ACM, 2016.

[34] Nathan Linial. Locality in Distributed Graph Algorithms. *SIAM J. Comput.*, 21(1):193–201, 1992.

[35] Takanori Maehara and Yutaro Yamaguchi. Stochastic Monotone Submodular Maximization with Queries. *CoRR*, abs/1907.04083, 2019.

[36] Jamie Morgenstern and Tim Roughgarden. On the pseudo-dimension of nearly optimal auctions. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 136–144, 2015.

[37] Michael Ostrovsky and Michael Schwarz. Reserve prices in internet advertising auctions: a field experiment. In *Proceedings 12th ACM Conference on Electronic Commerce (EC-2011), San Jose, CA, USA, June 5-9, 2011*, pages 59–60. ACM, 2011.

[38] Tim Roughgarden and Joshua R. Wang. Minimizing regret with multiple reserves. *ACM Trans. Economics and Comput.*, 7(3):17:1–17:18, 2019.

[39] Imre Z Ruzsa and Endre Szemerédi. Triple Systems With No Six Points Carrying Three Triangles. *Combinatorics (Keszthely, 1976), Coll. Math. Soc. J. Bolyai*, 18:939–945, 1978.

[40] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24. Springer Science & Business Media, 2003.

[41] 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.