Making Decisions with Incomplete and Inaccurate Information

by

Vijay Menon

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2021

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: Ioannis Caragiannis
Professor
Department of Computer Science, Aarhus University

Supervisor: Kate Larson
Professor
Cheriton School of Computer Science, University of Waterloo

Internal Members: Shai Ben-David
Professor
Cheriton School of Computer Science, University of Waterloo

Eric Blais
Associate Professor
Cheriton School of Computer Science, University of Waterloo

Internal-External Member: Stanko Dimitrov
Associate Professor
Department of Management Sciences, University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

The contents of this thesis are based on the following papers that I have co-authored.

1. Text and results in Chapter 2 are based on

   [ML18] Vijay Menon and Kate Larson. "Robust and Approximately Stable Marriages Under Partial Information". In: *Proceedings of the Fourteenth International Conference on Web and Internet Economics (WINE)*. 2018, pp. 341–355

2. Text and results in Chapter 3 are based on

   [ML19] Vijay Menon and Kate Larson. "Mechanism Design for Locating a Facility Under Partial Information". In: *Proceedings of the Twelfth International Symposium on Algorithmic Game Theory (SAGT)*. 2019, pp. 49–62

3. Text and results in Chapter 4 are based on

   [MML21] Thomas Ma, Vijay Menon, and Kate Larson. "Improving Welfare in One-Sided Matching Using Simple Threshold Queries". In: *Proceedings of the Thirteeth International Joint Conference on Artificial Intelligence (IJCAI)*. 2021

4. Text and results in Chapter 5 are based on

   [ML20] Vijay Menon and Kate Larson. *Algorithmic Stability in Fair Allocation of Indivisible Goods Among Two Agents*. 2020. arXiv: 2007.15203 [cs.GT]

Abstract

From assigning students to public schools to arriving at divorce settlements, there are many settings where preferences expressed by a set of stakeholders are used to make decisions that affect them. Due to its numerous applications, and thanks to the range of questions involved, such settings have received considerable attention in fields ranging from philosophy to political science, and particularly from economics and, more recently, computer science.

Although there exists a significant body of literature studying such settings, much of the work in this space make the assumption that stakeholders provide complete and accurate preference information to such decision-making procedures. However, due to, say, the high cognitive burden involved or privacy concerns, this may not always be feasible. The goal of this thesis is to explicitly address these limitations. We do so by building on previous work that looks at working with incomplete information, and by introducing solution concepts and notions that support the design of algorithms and mechanisms that can handle incomplete and inaccurate information in different settings.

We present our results in two parts. In Part I we look at decision-making in the presence of incomplete information. We focus on two broad themes, both from the perspective of an algorithm or mechanism designer. Informally, the first one studies the following question: *Given incomplete preferences, how does one design algorithms that are 'robust', i.e., ones that produce solutions that are "good" with respect to the underlying complete preferences?* We look at this question in context of two well-studied problems, namely, *i)* (a version of) the two-sided matching problem and *ii)* (a version of) the facility location problem, and show how one can design approximately-robust algorithms in such settings. Following this, we look at the second theme, which considers the following question: *Given incomplete preferences, how can one ask the agents for some more information in order to aid in the design of 'robust' algorithms?* We study this question in the context of the one-sided matching problem and show how even a very small amount of extra information can be used to get much better outcomes overall.

In Part II we turn our attention to decision-making in the presence of inaccurate information and look at the following question: *How can one design 'stable' algorithms, i.e., ones that do not produce vastly different outcomes as long as there are only small inaccuracies in a stakeholder's report of their preferences?* We study this in the context of fair allocation of indivisible goods among two agents and show how, in contrast to popular fair allocation algorithms, there are alternative algorithms that are fair and approximately-stable.

Acknowledgements

I would first like to thank my advisor, Kate Larson. Kate has always been supportive and encouraging, has always balanced my pessimism when it comes to my work with her optimism, and has always given me the freedom to pursue my own interests. I am grateful to have had the opportunity to work with her over the last several years.

I would also like to thank the members of my committee, Shai Ben-David, Eric Blais, Ioannis Caragiannis, and Stanko Dimitrov, for reading my thesis, for providing feedback, and for insightful questions.

Finally, I am grateful to my friends for all the good times and their companionship over the years, and thank my family for their support.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

There are numerous settings where an algorithm interacts with a set of stakeholders and asks their preferences over a set of outcomes in order to make decisions that affect them. For instance, centralized algorithms, that ask the students to submit a ranking over the schools they would like to attend, are used in many cities to assign students to public schools [AS03; APR05; Abd+05]; in many countries, medical residents are matched to hospitals using algorithms that ask the residents to submit their preferences over the hospitals and vice-versa [Rot84; Rot86; Man16; NRM; CAR]; such algorithms are also used to fairly divide different types of resources (e.g., compute resources [Gho+11]). Perhaps unsurprisingly, then, and because of the interdisciplinary nature of the questions involved, the design of such decision-making systems have been extensively studied by mathematicians, philosophers, political scientists, economists, and, more recently, computer scientists. Collectively, the study of formal frameworks to analyse and aggregate preferences in collective-decision making systems has been referred to as *social choice theory*, and its birth can be dated back to the 18th century French philosopher and mathematician Condorcet [Con85]. The more recent field of *computational social choice* (also commonly referred to as algorithmic economics, computational economics, algorithmic game theory), under which the work in this thesis falls, aims to provide a computational lens into such settings.

Despite the existence of a large body of work across disciplines that studies various collective decision-making settings, there are certain issues that have not received much attention. One issue, highlighted in this thesis, is that a disproportionate amount of the work on such problems assume that stakeholders—which could be humans, software agents, or a combination of both (and so henceforth are referred to as *agents*)—are capable, and willing, to provide *complete* and *accurate* preference information. While a formal definition of what it means for preferences to be *complete* and *accurate* differs based on the decision-making setting, for now it is enough to think of the former in terms of a setting where agents are asked to provide a ranking over all the candidates in, say, an election. As for the latter, for now it is enough to think of it in terms of a setting where agents are expected to indicate their exact *utility*—i.e., their value, which is, say, a positive integer—for different outcomes.

The assumption that agents provide complete and accurate preference information is reasonable in some settings. For instance, consider a simplified scenario where an agent is trying to decide which house to rent among the n they have had a viewing for, and their only concern is whether there is a grocery store within m kilometres from the house. Assuming that n is small (say, $n = 7$) and that the agent has access to information on the distance to the nearest grocery store, it is reasonable to believe that they will be able provide a ranking over all the houses (with potentially some houses being tied).

However, in many scenarios it is unrealistic to assume this will be the case. Indeed, in the setting mentioned above, the houses might have different monthly rents, they might have slightly different amenities, and the agent might care about not just the distance to the nearest grocery store, but also whether that store is accessible by public transit. Given this, ranking all the houses might no longer be feasible since it not only requires the agent to compile all the relevant information for all the houses—which in turn places a considerable cognitive burden on them—but might also require them to attribute numerical values to how much they value the different amenities and the ease of access to the grocery store. More broadly, an agent may not be able to provide complete and accurate preference information because *i)* there are just too many options to evaluate and they do not have much information or the expertise to evaluate them all, *ii)* they have some privacy concerns and hence do not want to reveal their complete preferences, and *iii)* determining their preferences exactly and accurately places too high a cognitive burden on them.

The goal of this thesis is to contribute to the theoretical foundations for the design of decision-making systems that are capable of handling incomplete and inaccurate information. The hope is that if there exists a solid foundation for the design of decision-making systems that can handle more realistic assumptions on the kind of information it has access to, then the general ideas from the field of (computational) social choice will see wider adoption. So, with this hope, we make progress towards our goal by

i) considering several settings where assuming availability of complete or accurate preference information is unrealistic and in turn using them to highlight different issues that arise when working with incomplete or inaccurate information

ii) expanding or complementing existing work that looks at how to deal with, and design algorithms for, scenarios where agents' preferences are incomplete

iii) introducing and analysing new solution concepts and notions that aid in the design of algorithms that provide certain guarantees even in the presence of incomplete or inaccurate preferences.

Before we delve more into these points and the technical overview of the work in this thesis, we present a brief, and informal, introduction to some basic terms and concepts that will be useful for understanding the rest of this chapter. A more formal treatment of all the terms and definitions is presented in the respective chapters where they are used.

1.1 Some Basic Terms and Concepts

All the work in this thesis deals with settings where a set of agents provide an algorithm with their *preferences*, and the overarching goal in all of these situations is for the algorithm to select an outcome that is *socially desirable*. In particular, we consider the following four settings, namely, *i) two-sided matching*, *ii) one-sided matching*, *iii) fair allocation of indivisible goods*, and *iv) facility location*. Each of these settings are well-studied in the (computational) social choice literature and enable us to highlight different aspects of working with incomplete and inaccurate preference information. For instance, motivated by applications in assigning students to public schools, the one-sided matching setting allows us to think of scenarios where an algorithm, when presented with incomplete preferences, can ask the agents to provide some more information in order for it to pick a better outcome; facility location is arguably the simplest setting that allows us to talk about incentive issues that arise when working with incomplete preference information; and motivated by observations we encountered on a popular fair division website, the setting of fairly allocating indivisible objects allows us to highlight the challenges when working with inaccurate preferences. Below we provide a very broad description of these settings.

i) Two-sided matching: In this setting, there are two disjoint sets of agents U and W, and each agent in a set provides their preferences over agents in the other set. The goal is to find a *matching*—which in turn is a collection of pairs, one from each set, such that no agent is part of more than one pair—that is *socially desirable*.

ii) One-sided matching: In this setting, there is a set of agents and a set of objects, and each agent provides their preferences over the objects. The goal is again to find a *socially desirable* matching such that each agent is assigned at most one object.

iii) Fair allocation of indivisible goods: In this setting, there is a set of agents and a set of indivisible goods, and each agent provides their preferences over the goods. The goal is to find a *fair* (which in this case is what is considered to be socially desirable) assignment (a.k.a. division, allocation), where an assignment is a partition of the goods among the agents.

iv) Facility location: In this setting, there is a set of agents who have preferences for where they want a facility (like, say, a public school) to be located, and the goal is to pick a location that is *socially desirable*.

Given that we consider several settings, naturally, the definitions of preferences, the outcome space, and what is socially desirable, etc., will vary depending on the context. Nevertheless, below we informally introduce some of these terms and related definitions. As mentioned previously, the formal definitions appear in the chapters.

Preferences: In order to understand how much an agent likes a particular outcome (a.k.a. alternative), the algorithm requires some information. There are two common approaches to elicit this information.

i) A common approach is to ask the agents to submit *ordinal preferences*, which typically entails that the agents submit a linear or weak order over the set of outcomes. For an agent i and outcomes a, b, we typically use $a \succ_i b$ (respectively, $a \succeq_i b$) to denote that agent i strictly prefers (respectively, weakly prefers) outcome a over b.

ii) The second common approach is to ask the agents to submit *cardinal preferences*, which entails that the agents submit their utility (a.k.a. value, which can be real number or integer) for each outcome. For an agent i, we use v_i to denote their utility function (a.k.a. valuation function), and often times we are interested in the cardinal preferences that are *consistent* with the given ordinal preferences. Here, by consistent we mean that, for outcomes a and b, $a \succeq_i b \Leftrightarrow v_i(a) \geq v_i(b)$.

Matching: Given two disjoint sets U and W, throughout, we use the term matching to refer to a bijection $\mu : U \to W$.

Stable matchings: Although, depending on the application, there are several properties that can be considered as 'desirable' in a two-sided matching setting, a large fraction of the work on two-sided matching, including Chapter 2 of this thesis, focuses on finding *stable matchings*. Given two disjoint sets of agents U and W, where each agent in a set specifies a linear order over agents in the other set, a matching μ is said to be stable if there is no *blocking pair*, i.e., a pair (m, w), where $m \in U$ and $w \in W$, such that $w \succ_m \mu(m)$ and $m \succ_w \mu(w)$ (i.e., m prefers w over their partner in the matching and vice-versa).

Pareto optimality and Pareto optimal matchings: In general an outcome is called Pareto optimal if no agent can be made better off without making another agent worse-off. Although, just like in two-sided matching settings, there are several properties that can be considered as 'desirable' in a one-sided matching setting, the most well-studied one, and one we consider in Chapter 4, is *Pareto optimal matchings.* Given a set of agents $\mathcal{N} = \{a_1, \ldots, a_n\}$ and objects $\mathcal{H} = \{h_1, \ldots, h_m\}$, where agents specify their preferences over objects, a matching μ is Pareto optimal if for all matchings μ',

$$(\exists a_i \in \mathcal{N}, \mu'(a_i) \succ_i \mu(a_i)) \Rightarrow (\exists a_j \in \mathcal{N}, \mu'(a_j) \prec_j \mu(a_j))\,.$$

In words, a matching μ is Pareto optimal if there is no other matching μ' where all the agents are assigned an object that they prefer at least as much as in the original matching and there is at least one agent who strictly prefers their assignment in the new matching.

Social welfare: In settings where the agents express cardinal preferences, social welfare refers to the sum of utilities that the agents have for an outcome. Finding the *optimal (social) welfare* refers to the goal of finding the outcome that maximizes social welfare. Note that an outcome with the optimal social welfare is also Pareto optimal.

Equipped with these informal definitions, in the next section we present an overview of the structure of the thesis and our results.

1.2 Overview of the Thesis

The results in this thesis are presented in two parts. The first part consists of three chapters and focusses on making decisions with incomplete information in different settings. The second part consists of a chapter that looks at making decisions with inaccurate information.

1.2.1 Part I: Making decisions with incomplete information

As mentioned previously, there are several reasons why an agent might not be able, or willing, to provide complete preferences. (Throughout, we often use the terms 'incomplete preferences' and 'partial preferences' interchangeably.) In this part of the thesis we look at three different settings, one in each chapter, and focus on two broad themes, both from the perspective of an algorithm or mechanism designer. Informally, the first one deals with the following question: *Given incomplete preferences, how does one design algorithms that* *are 'robust', i.e., ones that produce solutions that are "good" with respect to the underlying complete preferences?* Chapters 2 and 3 directly deal with this question.

Chapter 2: Two-Sided Matching under Partial Information

In Chapter 2 we consider the well-studied stable marriage (SM) problem. In this problem, we are given two disjoint sets U and W, where each agent in U and W provides some preferences over agents in the other set, and the goal is usually to match the agents in U to the ones in W so that the resulting matching is stable. While the most basic version of this problem assumes that the agents provide strict linear orders over agents in the other set, we consider a setting where they are allowed to submit strict weak orders.[1]

Although the issue of incomplete preferences has been previously studied in the context of SM, the approach in most of them has been to either work with *weakly-stable matchings*, which are stable matchings that arise as a result of an arbitrary linear extension of the submitted partial orders, or to look at *super-stable matchings*, which are matchings that are stable with respect to all the possible linear extensions of the submitted partial orders. In the case of the former, one key issue is that we often do not really know how "good" a particular weakly-stable matching is with the respect to the underlying true orders of the agents, and in the case of the latter they often do not exist. Therefore, in this chapter, we move away from these extremes and instead try to find a middle-ground when it comes to working with incomplete information. In particular, we look at answering the following two questions: *i)* How can a designer generate matchings that are robust and what is a possible measure of how "good" a matching is with respect to the underlying complete preferences? *ii)* What is the trade-off between the amount of missing information and the "quality" of solution one can get?

With the goal of resolving these questions through a simple and prior-free approach, we look at matchings that minimize the maximum number of blocking pairs with respect to all the possible underlying true orders as a measure of "goodness" or "quality", and subsequently provide results on finding such matchings. In particular, we first restrict our attention to matchings that have to be stable with respect to at least one of the completions (i.e., weakly-stable matchings) and show that in this case arbitrarily filling-in the missing information and computing the resulting stable matching can give a non-trivial approximation factor for

[1] A *strict linear order* is a binary relation $\succ$ that is irreflexive, transitive, and connected—meaning, for two agents a, b, where $a \neq b$, either $a \succ b$ or $b \succ a$. A *strict weak order* a binary relation $\succ$ that is irreflexive, transitive, and where for agents a, b, c, if a and b are incomparable—meaning, neither $a \succ b$ nor $b \succ a$—and b and c are incomparable, then a and c are incomparable.

our problem in certain cases. We complement this result by showing that, even under severe restrictions on the preferences of the agents, the factor obtained is asymptotically tight in many cases. We then investigate a special case, where only agents on one side provide strict weak orders and all the missing information is at the bottom of their preference orders, and show that in this special case the negative result mentioned above can be circumvented in order to get a much better approximation factor; this result, too, is tight in many cases. Finally, we move away from the restriction on weakly-stable matchings and show a general hardness of approximation result.

Chapter 3: Mechanism Design for Locating a Facility under Partial Information

Continuing with the theme of finding 'robust' solutions, we move to Chapter 3 where we study the facility location problem. Informally, in the simplest version of this problem, a planner wants to locate a facility on the real line and there are n agents, each with a preferred location x_i for the public facility. The goal, then, is to see if it is possible to design mechanisms—which can be essentially thought of as algorithms with incentive built-in—that can minimize some objective function (like, for instance, minimize the sum of distances for the agents to the facility), perhaps even approximately, and at the same time incentivize the agents to report their preferred locations *truthfully*—meaning, no agent i obtains a higher utility by reporting an $x_i' \neq x_i$.

Although this problem has been extensively studied, the assumption in all of this work is that the agents are always precisely aware of their preferred locations on the real line. However, this might not always be the case, and so in this chapter we deal with the scenario where agents only provide coarse information—namely, that for each agent i their preferred location lies in some interval K_i. While almost all of mechanism design deals with the case when agents provide complete preference information, we explore the design of robust mechanisms given such incomplete information when considering two well-studied objective functions. This in turn raises some technical questions since notions like *truthfulness*, or more formally the usual equilibrium solution concepts, are no longer well-defined in this setting. Therefore, we consider two natural equilibrium solution concepts for this setting, namely, *i)* very weak dominance and *ii)* minimax dominance. We show that under the former solution concept, there are no mechanisms that do better than a naive mechanism which always, irrespective of the information provided by the agents, outputs the same location. However, when using the latter, weaker, solution concept, we show that one can do significantly better, and we provide upper and lower bounds on the performance of mechanisms when considering different objective functions of interest. Furthermore, it also turns out that our mechanisms can be viewed as extensions to the classical optimal mechanisms in that they perform optimally when agents precisely specify their preferences.

Chapter 4: Improving Welfare in One-sided Matching using Threshold Queries

While Chapters 2 and 3 deal with the case when the designer has to work with the given incomplete information, it might be possible to request agents to provide more information to improve the final outcome. This is precisely the kind of setting considered in Chapter 4 and here we explore the second theme, which considers the following question: *Given incomplete preferences, how can one ask the agents for some more information in order to design algorithms that produce solutions that are "good" with respect to the underlying complete preferences?*

In particular, here we study one-sided matching problems where n agents have preferences over m objects and each of them need to be assigned at most one object. Most work on such problems assume that the agents only have ordinal preferences and usually the goal in these is to compute a matching that satisfies some notion of economic efficiency like Pareto optimality. However, agents may have some preference intensities or cardinal utilities that, e.g., indicate that they like an an object much more than another object. We first show how not taking these into account can result in a significant loss in social welfare. While one way to potentially account for these is to directly ask the agents for this information, such an elicitation process is cognitively demanding. Therefore, we focus on learning more about their cardinal preferences using simple threshold queries which ask an agent if their utility for an object is greater than a certain value, and use this in turn to come up with algorithms that produce a matching that, for a particular economic notion X, satisfies X and also achieves a good approximation to the optimal welfare among all matchings that satisfy X. We focus on several notions of economic efficiency, and look at both adaptive and non-adaptive algorithms. Overall, our results show how one can improve welfare by even non-adaptively asking the agents for just one bit of extra information per object.

1.2.2 Part II: Making decisions with inaccurate information

In situations where agents are asked for cardinal preferences, attributing exact numerical values to outcomes can be a cognitively demanding task, and so it is not hard to imagine scenarios where an agent might make "mistakes" while reporting their preferences—meaning, instead of reporting x, they might report x' due to the inherent difficulty in attributing numerical values to preferences. The second part of the thesis deals with a such a setting.

Chapter 5: Algorithmic Stability in Fair Allocation of Indivisible Goods

In Chapter 5 we look at the problem of allocating a set of indivisible goods among a set of agents. The main constraint is that the goods need to be divided among the agents in a 'fair' way (for specific definitions of what it means to be 'fair'). Most of the literature on this problem assume that the agents provide cardinal preferences. However, given the difficulty in translating preferences to numerical values, ideally we would like algorithms to not be overly sensitive to the exact numerical values provided.

Therefore, towards this end, and in particular to reduce the impact of "small" or "innocuous" mistakes, we propose a notion of algorithmic stability for scenarios where cardinal preferences are elicited. Informally, our definition captures the idea that an agent should not experience a large change in their utility as long as they make "small" or "innocuous" mistakes while reporting their preferences. We study this notion in the context of fair and efficient allocations of indivisible goods among two agents, and show that it is impossible to achieve exact stability along with even a weak notion of fairness and even approximate efficiency. As a result, we propose two relaxations to stability, namely, approximate-stability and weak-approximate-stability, and show how popular fair division algorithms that guarantee fair and efficient outcomes perform poorly with respect to these relaxations. This leads us to the explore the possibility of designing new algorithms that are more stable. Towards this end we present a general characterization result for a notion of fairness called pairwise maximin share allocations, and in turn use it to design an algorithm that is approximately-stable and guarantees a pairwise maximin share and Pareto optimal allocation for two agents. Finally, we present a simple framework that can be used to modify existing fair and efficient algorithms in order to ensure that they also achieve weak-approximate-stability.

Part I

Making Decisions with Incomplete Information

Chapter 2

Two-Sided Matching Under Partial Information

2.1 Introduction

In this chapter we begin with the theme of working with incomplete preferences. In particular, we are interested in designing 'robust' algorithms—informally, algorithms that when given access to incomplete preferences provide some guarantees with respect to the underlying complete preferences. We look at this question in the context of *two-sided matching* problems. Broadly, two-sided matching problems model scenarios where there are two disjoint sets of agents that want to be matched with each other. Each agent in a set has preferences over agents in the other set, and the goal is to match the agents in some socially desirable way. Given their numerous applications, e.g., in matching students to dormitories (known as the Stable Roommates problem (SR) [Irv85]), residents to hospitals (known as the Hospital-Resident problem (HR) [Man16]) etc., such problems have been extensively studied (see the books by Gusfield and Irving [GI89] and Manlove [Man13] for a survey on two-sided matching problems) and they are ubiquitous in practice.

While the class of two-sided matching problems is large, the focus of this chapter is on a specific two-sided matching problem called the *Stable Marriage problem* (SM), first introduced by Gale and Shapley [GS62]. In SM we are given two disjoint sets U and W (colloquially referred to as the set of men and women) and each agent in one set specifies a strict linear order over the agents in the other set. The objective, then, is to find a *stable matching*, i.e., a collection of pairs (m, w), where $m \in U$ and $w \in W$, such that each man/woman is part of at most one pair and where there is no unmatched pair (m', w') such that m' prefers w' over their partner in the matching and vice-versa. A pair which satisfies the latter criterion is called a *blocking pair*.

Although the assumption that the agents will be able to specify strict linear orders is not unreasonable in small markets, in general, as the markets get larger, it may not be feasible for an agent to determine a complete ordering over all the alternatives. Furthermore, there may arise situations where agents may simply be unwilling to provide their complete preferences due to, say, privacy concerns. Thus, it is natural for a designer to allow agents the flexibility to specify partial orders, and so here we assume that the agents submit strict weak orders[1] (i.e., strict partial orders where the incomparability relation is transitive) that are consistent with their underlying true strict linear orders. Although the issue of partially specified preferences has received attention previously, we argue below that certain aspects have not been addressed sufficiently.

In particular, the common approach to the question of what constitutes a "good" matching in such a setting has been to either work with *weakly-stable matchings*, which are stable matchings that arise as a result of an arbitrary linear extension of the submitted partial orders, or to look at *super-stable matchings*, which are matchings that are stable with respect to all the possible linear extensions of the submitted partial orders [Irv94; Ras+14]. Although these are useful notions and approaches that are analogous to weakly-stability and super-stability have been studied in other contexts like, for instance, voting, where one talks about possible and necessary winners (e.g., see [XC11; Bau+12]), and in the context of one-sided matchings (e.g., see [AWX15; Azi+19]), one key issue with weak-stability is that we often do not really know how "good" a particular weakly-stable matching is with the respect to the underlying true preferences of the agents. And in the case of super-stability, one key issue that such matchings often do not exist for many instances. Furthermore, we believe that it is in the interest of the market-designer to understand how robust or "good" a matching is with respect to the underlying true preferences of the agents, for, if otherwise, issues relating to instability and market unravelling can arise since the matching that is output by a mechanism can be arbitrarily bad with respect to these true preferences.

Hence, in this chapter we propose to move away from the extremes of working with either arbitrary weakly-stable matchings or super-stable matchings, and to find a middle-ground when it comes to working with partial preference information. To this end, our aim here is to answer two questions from the perspective of a market-designer: *i)* How should one handle

[1] All our negative results naturally hold for the case when the agents are allowed to specify strict partial orders. As for our positive results, most of them can be extended for general partial orders, although the resulting bounds will be worse.

partial information so as to be able to provide guarantees with respect to the underlying true preference orders? *ii)* What is the trade-off between the amount of missing information and the quality of a matching that one can achieve? We discuss these in more detail in the following sections.

2.1.1 Working with partial information

When agents do not submit full preference orderings, there are several possible ways to cope with the missing information. One approach is to assume that there exists some underlying distribution from which the agents' true preferences are drawn (e.g., see [Haz+12]), and then use this information to find a "good" matching—which is, say, the one with the least number of blocking pairs in expectation. However, the success of such an approach crucially depends on having access to information about the underlying preference distributions which may not always be available. Therefore, here we make no assumptions on the underlying preference distributions and instead adopt a prior-free and absolute-worst-case approach where we assume that any of the linear extensions of the given strict partial orders can be the underlying true order, and we aim to provide solutions that *perform well* with respect to all of them. We note that similar worst-case approaches have been looked at previously, for instance, by Chiesa, Micali, and Allen-Zhu [CMA12] in the context of auctions.

The objective we concern ourselves with here is that of minimizing the number of blocking pairs, which is well-defined and has been considered previously in the context of matching problems (for instance, see [ABM05; BMM10]). In particular, for a given instance $\mathcal{I}$ our goal is to return a matching μ_{opt} that has the best worst case—i.e., a matching that has the minimum maximum 'regret' after one realises the true underlying preference orders. We refer to μ_{opt} as the minimax optimal solution.

More precisely, let $\mathcal{I} = (p_U, p_W)$ denote an instance, where $p_U = \{p_{u_1}, \cdots, p_{u_n}\}$, $p_W = \{p_{w_1}, \cdots, p_{w_n}\}$, $U = \{u_i\}_{i \in \{1,2,\cdots,n\}}$ and $W = \{w_i\}_{i \in \{1,2,\cdots,n\}}$ are the set of men and women respectively, and p_i is the strict partial order submitted by agent i. Additionally, let $C(p_i)$ denote the set of linear extensions of p_i, C be the Cartesian product of the $C(p_i)$s, i.e., $C = \bigtimes_{i \in U \cup W} C(p_i)$, $bp(\mu, c)$ denote the set of blocking pairs that are associated with the matching μ according to some linear extension $c \in C$, and $\mathcal{M}_{\mathcal{I}}$ denote the set of all possible matchings with respect to $\mathcal{I}$. Then, the matching μ_{opt} that we are interested in is defined as

$$\mu_{opt} = \underset{\mu \in \mathcal{M}_{\mathcal{I}}}{\arg\min} \max_{c \in C} |bp(\mu, c)| \,.$$

While we are aware of just one work by Drummond and Boutilier [DB13] who consider the minimax regret approach in the context of stable matchings (they consider it mainly in the context of preference elicitation; see Section 2.1.4 for more details), the approach, in general, is perhaps reminiscent, for instance, of the works of Hyafil and Boutilier [HB04] and Lu and Boutilier [LB11] who looked at the minimax regret solution criterion in the context of mechanism design for games with type uncertainty and preference elicitation in voting protocols, respectively.

Remark: In the usual definition of a minimax regret solution, there is a second term which measures the 'regret' as a result of choosing a particular solution. That is, in the definition above, it would usually be $\mu_{opt} = \arg\min_{\mu \in S} \max_{c \in C} |bp(\mu, c)| - |bp(\mu_c, c)|$, where μ_c is the optimal matching (with respect to the objective function $|bp()|$) for the linear extension c. This is not included in the definition above because $|bp(\mu_c, c)| = 0$ as every instance of the marriage problem with linear orders has a stable solution (which by definition has zero blocking pairs). Additionally, the literature on stable matchings sometimes uses the term 'regret' to denote the maximum cost associated with a stable matching, where the cost of a matching for an agent is the rank of its partner in the matching and the maximum is taken over all the agents (for instance, see [Man+02]). However, here the term regret is used in the context of the minimax regret solution criterion.

2.1.2 Measuring the amount of missing information

For the purposes of understanding the trade-off between the amount of missing information and the "quality" of solution one can achieve, we need a way to measure the amount of missing information in a given instance, and for this here we adopt the following. For a given instance $\mathcal{I}$, the amount of missing information, δ, is the average fraction of pairwise comparisons one cannot infer from the given strict partial orders. That is, we know that if every agent submits a strict linear order over n alternatives, then we can infer $\binom{n}{2}$ comparisons from it. Now, instead, if an agent i submits a strict partial order p_i, then we denote by δ_i the fraction of these $\binom{n}{2}$ comparisons one cannot infer from p_i (this is the "missing information" in p_i). Our δ here is defined as

$$\delta = \frac{1}{2n} \sum_{i \in U \cup W} \delta_i. \tag{2.1}$$

Although, given a strict partial order p_i, it is straightforward to calculate δ_i, we will nevertheless assume throughout that δ is part of the input. Hence, our definition of an instance will be modified the following way to include the parameter for missing information: $\mathcal{I} = (\delta, p_U, p_W)$.

Remark: $\delta = 0$ denotes the case when all the preferences are strict linear orders. Also, for an instance with n agents on each side, the least value of δ when the amount of missing information is non-zero is $\frac{1}{2n}\frac{1}{\binom{n}{2}}$ (this happens in the case where there is only one agent with just one pairwise comparison missing). However, despite this, in the interest of readability, we sometimes just write statements of the form "for all $\delta > 0$". Such statements need to be understood as being true for only realizable or valid values of δ that are greater than zero.

2.1.3 Our contributions

The focus in this chapter is on computing the minimax optimal matching, i.e., a matching that, when given an instance $\mathcal{I}$, minimizes the maximum number of blocking pairs with respect to all the possible linear extensions (see Section 2.2.1 for a formal definition of the problem). Towards this end, we make the following contributions.

- First, we formally define the problem and show that it is equivalent to the problem of finding a matching that has the minimum number of *super-blocking pairs* (i.e., man-woman pairs where each of them weakly-prefers the other over their current partners). Although an optimal answer to our question might involve matchings that have man-woman pairs such that each of them strictly prefers the other over their partners, we start by focusing on matchings that do not have such pairs. Given the fact that any matching with no such pairs are weakly-stable, through this setting we address the question: *Given an instance, is it possible to find weakly-stable matching that performs well, in terms of minimizing the number of blocking pairs, with respect to all the linear extensions of the given strict partial orders?* We show that by arbitrarily filling-in the missing information and computing the resulting stable matching, one can obtain a non-trivial approximation factor (i.e., one that is $o(n^2)$) for our problem for many values of δ. We complement this result by showing that, even under severe restrictions on the preferences of the agents, the factor obtained is asymptotically tight in many cases.

- By assuming a special structure on the agents' preferences—one where strict weak orders are specified by just one set of agents and all the missing information is at the bottom of their preference orders—we show that there is an $O(n)$-approximation algorithm for our problem. The proof of this is via finding a 2-approximation for another problem (see Problem 3) that might be of independent interest.

- In Section 2.4 we remove the restriction to weakly-stable matchings and show a general hardness of approximation result for our problem. Following this, we discuss one possible approach that can lead to a near-tight approximation guarantee.

2.1.4 Related work

Starting with the work of Gale and Shapley [GS62], two-sided matching problems have been extensively studied, due, in part, to their numerous applications [GI89; RS90; Man13]. For instance, it is used to match medical residents to hospitals [Rot84; Rot86; Man16] in many countries [NRM; CAR], to match students to universities [Rom98; BB04], etc. We refer the reader to the books by Gusfield and Irving [GI89] and Manlove [Man13] for a survey on two-sided matching problems.

In this vast literature, there are several papers that have looked at problems relating to the topic of chapter, and that broadly deal with missing preference information or uncertainty in preferences. We discuss them below.

Drummond and Boutilier [DB13] used the minimax regret solution criterion in order to drive preference elicitation strategies for matching problems. While their paper discusses about computing robust matchings subject to a minimax regret solution criteria, their focus was on providing an NP-completeness result and heuristic preference elicitation strategies for refining the missing information. In contrast, in addition to focusing on understanding the exact trade-offs between the amount of missing information and the solution "quality", our main concern here is to derive approximation algorithms for computing such robust matchings.

Rastegari et al. [Ras+14] studied a partial information setting in labour markets. However, again, the focus of this paper was different. They looked at pervasive-employer-optimal matchings, which are matchings that are employer-optimal (see [Ras+14] for the definitions) with respect to all the underlying linear extensions. In addition, they also discussed how to identify, in polynomial time, if a matching is employer-optimal with respect to some linear extension.

Aziz et al. [Azi+16] looked at the stable matching problem in settings where there is uncertainty about the preferences of the agents. They considered three different models of uncertainty and primarily studied the complexity of computing the stability probability of a given matching and the question of finding a matching that will have the highest probability of being stable. In contrast to their work, here we do not make any underlying distributional assumptions about the preferences of the agents and instead take an absolute worst-case approach, which in turn implies that our results hold irrespective of the underlying distribution on the completions.

Finally, we also briefly mention another line of research which deals with partial information settings and goes by the name of *interview minimization* (see, for instance, the papers by Rastegari et al. [Ras+13] and Drummond and Boutilier [DB14]). One of the main goals in this line of work is to come with a matching that is stable (and possibly satisfying some other desirable property) by conducting as few 'interviews' (which in turn helps the agents in refining their preferences) as possible. We view this work as an interesting, orthogonal, direction from the one we pursue here.

2.2 Preliminaries

Let U and W be two disjoint sets. The sets U and W are colloquially referred to as the set of men and women, respectively, and $|U| = |W| = n$. We assume that each agent in U and W has a strict linear order (i.e., a ranking without ties) over the agents in the other set, but this strict linear order may be unknown to the agents or they may be unwilling to completely disclose the same. Hence, each agent in U and W specifies a strict partial order over the agents in the other set (which we refer to as their *preference order*) that is *consistent* with their underlying true orders. For an agent i, their preference order p_i is consistent with their underlying strict linear order if agent i prefers a to b in p_i only if it prefers a to b according to the underlying linear order.

Throughout, we use p_U and p_W, respectively, to denote the collective preference orders of all the men and women (a.k.a. *profile* of preferences of men and women). For a strict partial order p_i associated with agent i, we denote the set of linear extensions associated with p_i by $C(p_i)$ and denote by C the Cartesian product of the $C(p_i)$s, i.e., $C = \bigtimes_{i \in U \cup W} C(p_i)$. We refer to the set C as "the set of all completions" where the term *completion* refers to an element in C. Also, throughout, we denote strict preferences by $\succ$ and use $\succeq$ to denote the relation 'weakly-prefers'. So, for instance, we say that an agent i strictly prefers a to b and denote this by $a \succ_i b$ and use $a \succeq_i b$ to denote that either i strictly prefers a to b or finds them incomparable. As mentioned in the introduction, we restrict our attention to the case when the strict partial orders submitted by the agents are strict weak orders over the set of agents in the other set.

Remark: Strict weak orders are defined to be strict partial orders where incomparability is transitive. Hence, although the term *tie* is used to mean indifference, it is convenient to think of strict weak orders as rankings with ties. Therefore, throughout, we will use the terms 'ties' and 'incomparabilities' interchangeably, and whenever we say that agent c finds a and b to be tied, we mean that c finds a and b to be incomparable.

An instance $\mathcal{I}$ of the stable marriage problem (SM) is defined as $\mathcal{I} = (\delta, p_U, p_w)$, where δ denotes the amount of missing information in that instance and this in turn, as defined in (2.1), is the average number of pairwise comparisons that are missing from the instance, and p_U and p_W are as defined above. We use $\mathcal{M}_\mathcal{I}$ to denote the set of all possible matchings with respect to $\mathcal{I}$, where a matching $\mu \in \mathcal{M}$ is a set of disjoint pairs (m, w), where $m \in U$ and $w \in W$.

Given an instance $\mathcal{I}$, the goal is usually to come up with a matching $\mu \in \mathcal{M}_\mathcal{I}$ that is stable. There are different notions of stability that have been proposed and below we define two of them that are relevant to our results here: *i)* weak-stability and *ii)* super-stability. However, before we look at their definitions we introduce the following terminology that will be used throughout this chapter. (Note that in the definitions below we implicitly assume that in any matching μ all the agents are matched. This is so because of the standard assumption that is made in the literature on SM—i.e., the stable marriage problem where every agent has a strict linear order over all the agents in the other set—that an agent always prefers to be matched to some agent than to remain unmatched.)

Definition 1 (blocking pair/obvious blocking pair)**.** Given an instance $\mathcal{I}$ and a matching $\mu \in \mathcal{M}_\mathcal{I}$, (m, w) is said to be a blocking pair associated with μ if $w \succ_m \mu(m)$ and $m \succ_w \mu(w)$. The term blocking pair is usually used in situations where the preferences of the agents are strict linear orders, so in cases where the preferences of the agents have missing information, we refer to such a pair as an obvious blocking pair.

Definition 2 (super-blocking pair)**.** Given an instance $\mathcal{I}$ where the agents submit partial preference orders and a matching $\mu \in \mathcal{M}_\mathcal{I}$, we say that (m, w), where $\mu(m) \neq w$, is a super-blocking pair with respect to μ if $w \succeq_m \mu(m)$ and $m \succeq_w \mu(w)$.

Given the definitions above we can now define weak-stability and super-stability.

Definition 3 (weakly-stable matching)**.** Given an instance $\mathcal{I}$ and matching $\mu \in \mathcal{M}_\mathcal{I}$, μ is said to be weakly-stable with respect to $\mathcal{I}$ if it does not have any obvious blocking pairs. When the preferences of the agents are strict linear orders, such a matching is just referred to as a stable matching.

Definition 4 (super-stable matching)**.** Given an instance $\mathcal{I}$ and matching $\mu \in \mathcal{M}_\mathcal{I}$, μ is said to be super-stable with respect to $\mathcal{I}$ if it does not have any super-blocking pairs.

2.2.1 Problem definitions

As mentioned in the introduction, we are interested in finding the minimax optimal matching where the objective is to minimize the number of blocking pairs, i.e., to find, for an instance $\mathcal{I}$, a matching that has the minimum maximum number of blocking pairs with respect to all the completions of the preferences. This is formally defined below.

Problem 1 (δ-minimax-matching)**.** Given a $\delta \in [0, 1]$ and an instance $\mathcal{I} = (\delta', p_U, p_W)$, where $\delta' \leq \delta$ is the amount of missing information and p_U, p_W are the preferences submitted by men and women, respectively, compute μ_{opt}, where

$$\mu_{opt} = \underset{\mu \in \mathcal{M}_{\mathcal{I}}}{\arg\min} \max_{c \in C} |bp(\mu, c)| \,. \tag{2.2}$$

Although the problem defined above is our main focus, for the rest of this chapter we will be talking in terms of the following problem which concerns itself with finding an approximately super-stable matching (i.e., a super-stable matching with the minimum number of super-blocking pairs). As we will see below, this is because both the problems are equivalent.

Problem 2 (δ-min-bp-super-stable-matching)**.** Given a $\delta \in [0, 1]$ and an instance $\mathcal{I} = (\delta', p_U, p_W)$, where $\delta' \leq \delta$ is the amount of missing information and p_U, p_W are the preferences submitted by men and women, respectively, compute μ^{SS}_{opt}, where

$$\mu^{SS}_{opt} = \underset{\mu \in \mathcal{M}_{\mathcal{I}}}{\arg\min} |\text{super-bp}(\mu)| \,, \tag{2.3}$$

and super-bp(μ) is the set of super-blocking pairs associated with μ for the instance $\mathcal{I}$.

Below we show that both the problems are equivalent. However, before that we prove the following lemma.

Lemma 1. *Given an instance $\mathcal{I} = (\delta, p_U, p_W)$, let $\mu \in \mathcal{M}_{\mathcal{I}}$, α denote the maximum number of blocking pairs associated with μ for any completion of $\mathcal{I}$, and β denote the number of super-blocking pairs associated with μ for the instance $\mathcal{I}$. Then, $\alpha = \beta$.*

Proof. First, it is easy to see that if there are α blocking pairs associated with μ for a completion, then there are at least as many super-blocking pairs associated with μ. Therefore, $\alpha \leq \beta$.

Next, we will show that if β is the number of super-blocking pairs associated with μ, then $\mathcal{I}$ has at least one completion such that it has β number of blocking pairs associated with μ. To see this, for each $m_i \in U$ and for each $w_j \in W$ such that (m_i, w_j) is a super-blocking pair, do the following:

- if m_i finds w_j incomparable to $\mu(m_i)$, then construct a new partial order p'_{m_i} for m_i such that it is the same as p_{m_i} except for the fact that in p'_{m_i} we have that m_i strictly prefers w_j over $\mu(m_i)$.
- if w_j finds m_i incomparable to $\mu(w_j)$, then construct a new partial order p'_{w_j} for w_j such that it is the same as p_{w_j} except for the fact that in p'_{w_j} we have that w_j strictly prefers m_i over $\mu(w_j)$.

Once the above steps are done, if there still exists any agent whose preference order is partial, then complete it arbitrarily. Now, consider this instance $\mathcal{I}'$ that is obtained and observe that every (m_i, w_j) which was a super-blocking pair associated with μ in $\mathcal{I}$ forms a blocking pair in μ with respect to $\mathcal{I}'$. Therefore, this completion has β blocking pairs, and since the maximum number of blocking pairs in any completion is α, we have that $\beta \leq \alpha$. Combining this with the case above, we have that $\alpha = \beta$. □

Given the lemma above, we can now show that the problems are equivalent.

Proposition 2. *For any $\delta \in [0, 1]$, the δ-minimax-matching and δ-min-bp-super-stable-matching problems are equivalent.*

Proof. Let $\mathcal{I} = (\delta', p_U, p_W)$ be an arbitrary instance, where $\delta' \leq \delta$, and $\mu \in \mathcal{M}_{\mathcal{I}}$. We show that μ is an optimal solution for the δ-minimax-matching problem if and only if it is an optimal solution for the δ-min-bp-super-stable-matching problem.

($\Rightarrow$) Let us suppose that μ is not an optimal solution for the δ-min-bp-super-stable-matching problem. This implies that there exists some other μ' such that $|\text{super-bp}(\mu')| < |\text{super-bp}(\mu)|$. However, from Lemma 1 we know that the maximum number of blocking pairs associated with μ' for any completion with respect to $\mathcal{I}$ is equal to $|\text{super-bp}(\mu')|$, which in turn contradicts the fact that μ was optimal for the δ-minimax-matching problem.

($\Leftarrow$) We can prove this analogously as in the case above. □

For the rest of this chapter, we assume that we are always dealing with instances which do not have a super-stable matching since the existence of such a matching can be checked in polynomial-time [Irv94, Theorem 3.4]. So, now, in the context of the δ-min-bp-super-stable-matching problem, it is easy to show that if the number of super-blocking pairs k in the optimal solution is a constant, then we can solve it in polynomial-time. We show this below. Later, in Section 2.4, we will see that the problem is NP-hard, even to approximate.

Proposition 3. *An exact solution to the δ-min-bp-super-stable-matching problem can be computed in $O(n^{2(k+1)})$ time, where k is the number of super-blocking pairs in the optimal solution.*

Proof. We will describe the algorithm below whose main idea is based on the following observation.

For an instance $\mathcal{I}$, consider its optimal solution μ_{opt} and let the k super-blocking pairs associated with μ_{opt} be $B = \{(m_1, w_1), \cdots, (m_k, w_k)\}$. Next, for each such pair (m_i, w_i), put m_i (respectively, w_i) at the end of w_i's (respectively, m_i's) preference list (i.e., in $\mathcal{I}$, make every other man (woman), except those involved in another blocking pair with w_i (m_i), rank better than m_i (w_i)). If either of them are involved in multiple blocking pairs in B, then make those partners as incomparable at the end of the preference list. Let us call the resulting instance $\mathcal{I}'$. Notice that μ_{opt} is super-stable with respect to $\mathcal{I}'$ as the pairs in B are no longer blocking and no new blocking pairs are created because of our manipulations to the preference list.

Given the above observation, we can now describe the exponential algorithm.

- Initially $j = 1$. Given a j, try out every possible set of pairs of size j to see if they are the right blocking pairs.

- For each set generated in the previous step, modify the original instance $\mathcal{I}$ to $\mathcal{I}'$ as described above and see if $\mathcal{I}'$ has a super-stable matching (as mentioned previously, this can be done in polynomial time [Irv94, Theorem 3.4]). If yes, then return the super-stable matching as that is the solution. Otherwise, if none of the sets of size j result in a "yes", then go back to step 1 and try again with the next value of j.

Now, it is easy to see that we end up with the optimal solution this way since we try all possible sets of blocking pairs. As for the time, we know that for each j we have at most $(n^2)^j$ choices of sets and for each set we need at most $2n^2$ time to do the necessary manipulations to the instance and to check for super-stability. Hence, the total time required is $\sum_{j=1}^{k} 2n^{2j+2} = O(n^{2(k+1)})$. ∎

2.3 Investigating Weakly-Stable Matchings

In this section we focus on situations where obvious blocking pairs are not permitted in the final matching. In particular, we explore the space of weakly-stable matchings and ask whether it is possible to find weakly-stable matchings that also provide good approximations to the δ-min-bp-super-stable-matching problem (and thus the δ-minimax-matching problem).

2.3.1 Using weakly-stable matchings to approximate the δ-min-bp-super-stable-matching problem

It has previously been established that a matching is weakly-stable if and only if it is stable with respect to at least one completion [Man+02, Section 1.2]. Therefore, given this, one immediate question that arises in the context of approximating the δ-min-bp-super-stable-matching problem is: *What if we just fill in the missing information arbitrarily and then compute a stable matching associated with such a completion?* This is the question we consider here, and we show that weakly-stable matchings do give a non-trivial (i.e., one that is $o(n^2)$, as any matching has only $O(n^2)$ super-blocking pairs) approximation bound for our problem for certain values of δ. The proof of the following theorem is through a simple application of the Cauchy-Schwarz inequality.

Theorem 4. *For any $\delta > 0$ and an instance $\mathcal{I} = (\delta', p_U, p_W)$ where $\delta' \leq \delta$, any weakly-stable matching with respect to $\mathcal{I}$ gives an $O\left(\min\left\{n^3\delta, n^2\sqrt{\delta}\right\}\right)$-approximation for the δ-min-bp-super-stable-matching problem.*

Proof. Let μ be a weakly-stable matching associated with $\mathcal{I}$. By the definition of weakly-stable matchings we know that μ does not have any obvious blocking pairs. This implies that for every super-blocking pair (m, w) associated with μ, either m finds w incomparable to his partner $\mu(m)$ or w finds m incomparable to her partner $\mu(w)$. If it is the former then we refer to the super-blocking pair (m, w) as one that is associated with m and if not we say that it is associated with w. Next, let us suppose that there are d agents who have a blocking pair associated with them and let b_i denote the number of super-blocking pairs associated with agent i. So, now, the number of super-blocking pairs, $|\text{super-bp}(\mu)|$, associated with μ can be written as

$$|\text{super-bp}(\mu)| = \sum_{i=1}^{d} b_i \leq \sum_{i=1}^{d} (\ell_i - 1) = \sum_{i=1}^{d} \ell_i - d, \tag{2.4}$$

where ℓ_i refers to the length of largest tie associated with agent i and the inequality follows from the definition of an association of a super-blocking pair with an agent.

Additionally, for each $i \in \{1, \ldots, d\}$, we know that at least $\binom{\ell_i}{2}$ pairwise comparisons are missing with respect to i (since i has a tie of length ℓ_i). Therefore, using the Cauchy-Schwarz inequality, we have that

$$\sum_{i=1}^{d} \binom{\ell_i}{2} = \frac{1}{2} \sum_{i=1}^{d} \ell_i^2 - \ell_i \geq \frac{1}{2} \left[\frac{1}{d} \left(\sum_{i=1}^{d} \ell_i \right)^2 - \sum_{i=1}^{d} \ell_i \right]. \tag{2.5}$$

Also, since the total amount of missing information δ' in the instance $\mathcal{I}$ is less than or equal to δ and since each $\ell_i \geq 2$ (as it is a weak order and a tie, if it exists, is of length at least 2) we have that

$$d \leq \sum_{i=1}^{d} (\ell_i - 1) \leq \sum_{i=1}^{d} \binom{\ell_i}{2} \leq \delta(2n) \binom{n}{2} \leq \delta n^3. \tag{2.6}$$

Now, using (2.6) and the fact that d is also upper-bounded by $2n$ (as there are only $2n$ agents in the instance), we have that $d \leq \min\{2n, \delta n^3\}$. Therefore, using (2.5) and again using the fact that δ is the maximum amount of missing information, we have,

$$\frac{1}{2} \left[\frac{1}{d} \left(\sum_{i=1}^{d} \ell_i \right)^2 - \sum_{i=1}^{d} \ell_i \right] \leq \sum_{i=1}^{d} \binom{\ell_i}{2} \leq \delta(2n) \binom{n}{2}. \tag{2.7}$$

This in turn implies that if we solve for $\sum_{i=1}^{d} \ell_i$, we have,

$$\sum_{i=1}^{d} \ell_i \leq \frac{1}{2} \left(d + \sqrt{d^2 + 8dn^2(n-1)\delta} \right) < d + \sqrt{d^2 + 8dn^2(n-1)\delta}.$$

So, now, we can use the fact that $d \leq \min\{2n, \delta n^3\}$ to see that

$$\sum_{i=1}^{d} \ell_i - d \leq \min\{4n^3\delta, 5n^2\sqrt{\delta}\}.$$

Finally, this along with (2.4) gives our result since the number of super-blocking pairs in the optimal solution is at least 1 (since, as mentioned in Section 2.2.1, we are only considering instances that do not have a super-stable matching). □

2.3.2 Can we do better when restricted to weakly-stable matchings?

While Theorem 4 established an approximation bound for the δ-min-bp-super-stable-matching problem when considering only weakly-stable matchings, it was simply based on arbitrarily filling-in the missing information. Therefore, there remains the question as to whether one can be clever about handling the missing information and as a result obtain improved approximation bounds. In this section we consider this question and show that for many values of δ the approximation factor obtained in Theorem 4 is asymptotically the best one can achieve when restricted to weakly-stable matchings.

Theorem 5. *For any $\delta \in [\frac{16}{n^2}, \frac{1}{4}]$, if there exists an α-approximation algorithm for δ-min-bp-super-stable-matching that always returns a matching that is weakly-stable, then $\alpha \in \Omega\left(n^2\sqrt{\delta}\right)$. Moreover, this result is true even if we allow only one side to specify ties and also insist that all the ties need to be at the top of the preference order.*

Proof. At a very high-level, the key idea in this proof is to create an instance $\mathcal{I}$ such that if we insist on there being no obvious blocking pairs, then this results in some kind of a "cascading effect", thus in turn causing a very sharp blow-up in the number of super-blocking pairs. With this intuition, we first construct an instance $\mathcal{I}$ as shown in Figure 2.1, where ties appear only on the women's side. Furthermore, we define the following:

- $y = \frac{n\sqrt{\delta}}{2}$, $z = \frac{n}{2y}$ (for simplicity we assume that y and z are integers; we can appropriately modify the proof if that is not the case)
- $b_j = \frac{n}{2} + jy + 1, \forall j \in [0, \ldots z]$, $B_i = \{b_{i-1}, \ldots, b_i - 1\}, \forall i \in [1, \ldots z]$
- $F = \{1, \ldots, \frac{n}{2}\}, S = \{\frac{n}{2} + 1, \ldots, n\}$
- $W_X\,(M_X)$: for some set X, place all the women (men) with index in X in the increasing order of their indices
- $W^T_X\,(M^T_X)$: for some set X, place all the women (men) with index in X as tied
- $[\ldots]$: place all the remaining alternatives in some strict order.

Men

$$
\begin{aligned}
m_1 &: w_1 \succ W_{B_1} \succ \ldots \succ W_{B_z} \succ W_{F\setminus\{1\}}\\
m_2 &: w_1 \succ w_2 \succ [\ldots]\\
m_3 &: w_2 \succ w_3 \succ W_{F\setminus\{2,3\}} \succ W_S\\
m_4 &: w_2 \succ w_4 \succ W_{F\setminus\{2,4\}} \succ W_S\\
&\vdots\\
m_{\frac{n}{2}} &: w_2 \succ w_{\frac{n}{2}} \succ W_{F\setminus\{2,\frac{n}{2}\}} \succ W_S\\
m_{b_0} &: w_1 \succ w_{b_0} \succ W_{B_1\setminus\{b_0\}} \succ W_{S\setminus B_1} \succ W_{F\setminus\{1\}}\\
&\vdots\\
m_{b_1-1} &: w_1 \succ w_{b_1-1} \succ W_{B_1\setminus\{b_1-1\}} \succ W_{S\setminus B_1} \succ W_{F\setminus\{1\}}\\
m_{b_1} &: w_1 \succ w_{b_1} \succ W_{B_2\setminus\{b_1\}} \succ W_{S\setminus B_2} \succ W_{F\setminus\{1\}}\\
&\vdots\\
m_{b_2-1} &: w_1 \succ w_{b_2-1} \succ W_{B_2\setminus\{b_2-1\}} \succ W_{S\setminus B_2} \succ W_{F\setminus\{1\}}\\
&\vdots\\
m_{b_{z-1}} &: w_1 \succ w_{b_{z-1}} \succ W_{B_z\setminus\{b_{z-1}\}} \succ W_{S\setminus B_z} \succ W_{F\setminus\{1\}}\\
&\vdots\\
m_{b_z-1} &: w_1 \succ w_{b_z-1} \succ W_{B_z\setminus\{b_z-1\}} \succ W_{S\setminus B_z} \succ W_{F\setminus\{1\}}
\end{aligned}
$$

Women

$$
\begin{aligned}
w_1 &: m_2 \succ m_1 \succ [\ldots]\\
w_2 &: m_2 \succ M_{(F\cup S)\setminus\{1\}} \succ m_1\\
w_3 &: m_1 \succ m_3 \succ [\ldots]\\
w_4 &: m_1 \succ m_4 \succ [\ldots]\\
&\vdots\\
w_{\frac{n}{2}} &: m_1 \succ m_{\frac{n}{2}} \succ [\ldots]\\
w_{b_0} &: M^T_{B_1\setminus\{b_0\}} \succ M_{S\setminus B_1} \succ m_1 \succ m_{b_0} \succ M_{F\setminus\{1\}}\\
&\vdots\\
w_{b_1-1} &: M^T_{B_1\setminus\{b_1-1\}} \succ M_{S\setminus B_1} \succ m_1 \succ m_{b_1-1} \succ M_{F\setminus\{1\}}\\
w_{b_1} &: M^T_{B_1\setminus\{b_1\}} \succ M_{S\setminus B_1} \succ m_1 \succ m_{b_1} \succ M_{F\setminus\{1\}}\\
&\vdots\\
w_{b_2-1} &: M^T_{B_1\setminus\{b_2-1\}} \succ M_{S\setminus B_1} \succ m_1 \succ m_{b_2-1} \succ M_{F\setminus\{1\}}\\
&\vdots\\
w_{b_{z-1}} &: M^T_{B_1\setminus\{b_{z-1}\}} \succ M_{S\setminus B_1} \succ m_1 \succ m_{b_{z-1}} \succ M_{F\setminus\{1\}}\\
&\vdots\\
w_{b_z-1} &: M^T_{B_1\setminus\{b_z-1\}} \succ M_{S\setminus B_1} \succ m_1 \succ m_{b_z-1} \succ M_{F\setminus\{1\}}
\end{aligned}
$$

Figure 2.1: The instance $\mathcal{I}$ that is used in the proof of Theorem 5

Next, we will show that all the weakly-stable matchings associated with $\mathcal{I}$ have $O(n^2\sqrt{\delta})$ super-blocking pairs, whereas the optimal solution has exactly one super-blocking pair. To do this, first note that the optimal solution μ_{opt} associated with the instance is $\mu_{opt} = \{(m_1, w_1), (m_2, w_2), \ldots, (m_n, w_n)\}$, where (m_2, w_1) is the only super-blocking pair (and it is an obvious blocking pair). Also, it can be verified that the total amount of missing information in $\mathcal{I}$ is at most δ. So, next, we prove the following claim.

Claim 1. If μ is a weakly-stable matching associated with the instance $\mathcal{I}$, then $\forall i \in \{\frac{n}{2}+1, \ldots, n\}, \mu(m_i) \neq w_i$.

Proof. First, note that in any weakly-stable matching m_2 will always be matched to w_1 as otherwise it will result in an obvious blocking pair. Next, let us suppose that there exists an $i \in \{\frac{n}{2}+1, \ldots, n\}$ such that $\mu(m_i) = w_i$. Now, we will consider the following two cases and show that in both the cases this is impossible.

Case 1. m_1 is matched to a woman $w \in W_{F\setminus\{1\}}$ in μ: In this case one can see that (m_1, w_i) forms an obvious blocking pair.

Case 2. m_1 is matched to a woman $w \in W_S$ in μ: Note that if this is the case, then there is at least one $j \in S$ such that m_j is matched with a woman $w \in W_F$. Now, notice that (m_j, w_i) forms an obvious blocking pair. □

Given the claim above, consider a man m whose index is in some block B_j and let his index value be k. From the way the preferences are defined, it is easy to see that in any weakly-stable matching, m will be matched to a woman w whose index lies in the same block B_j (because otherwise it will result in an obvious blocking pair). At the same time, from the claim above we know that this w's index is not k. Now, let us consider the woman w_{b_j-1} who is the woman with the highest index value in B_j and let m' denote the man who is matched to w_{b_j-1} in a weakly-stable matching. From the observation above we know that m' has an index value in B_j. Additionally, given the way the preferences are defined for m' and using the fact that any woman w_p such that $p \in B_j$ finds all the men in $B_j \setminus \{p\}$ to be incomparable, one can see that m' forms $(|B_j| - 2)$ super-blocking pairs (with all the women in B_j except w_{b_j-1} and the one with the same index value as m'). Also, by using the same argument again, but with respect to w_{b_j-2}, we can show that partner of w_{b_j-2} in the matching forms at least $(|B_j| - 3)$ super-blocking pairs (with all women except w_{b_j-2}, w_{b_j-1}, and the one with the same index). Continuing this way we see that each block B_j contributes $O(|B_j|^2)$ super-blocking pairs. And so, since there are z blocks and $|B_j| = y$ for all j, we have that there are $O(n^2\sqrt{\delta})$ super-blocking pairs in any weakly-stable matching. □

2.3.3 The case of one-sided top-truncated preferences: An $O(n)$ approximation algorithm

Although Theorem 5 is an inherently negative result, in this section we consider an interesting restriction on the preferences of the agents and show how this negative result can be circumvented. In particular, we consider the case where only agents on one side are allowed to specify ties and all the ties need to be at the bottom. Such a restriction has been looked at previously in the context of matching problems and as noted by Irving and Manlove [IM08] is one that appears in practise in the Scottish Foundation Allocation Scheme (SFAS). Additionally, restricting ties to the bottom models a very well-studied class of preferences known as top-truncated preferences, which has received considerable attention in the context of voting (see, for instance, [Bau+12]).

Top-truncated preferences model scenarios where an agent is certain about their most preferred choices, but is indifferent among the remaining ones or is unsure about them.

More precisely, in our setting, the preference order submitted by, say, a woman w is said to be a top-truncated order if it is a linear order over a subset of U and the remaining men are all considered to be incomparable by w. In this section we consider one-sided top-truncated preferences, i.e., where only men or women are allowed to specify top-truncated orders, and show an $O(n)$-approximation algorithm for δ-min-bp-super-stable-matching under this setting. (Without loss of generality we assume throughout that only the women submit strict weak orders.) Although arbitrarily filling-in the missing information and computing the resulting weakly-stable matching can lead to an $O(n^2\sqrt{\delta})$-approximate matching even for this restricted case (see Appendix A.2 for an example), we will see that not all weakly-stable matchings are "bad" and that in fact the $O(n)$-approximate matching we obtain is weakly-stable.

However, in order to arrive at this result, we first introduce the following problem which might be of independent interest. (To the best of our knowledge, this has not been previously considered in the literature.) Informally, in this problem we are given an instance $\mathcal{I}$ and are asked if we can delete some of the agents to ensure that the instance, when restricted to the remaining agents, will have a perfect super-stable matching.

Problem 3 (min-delete-super-stable-matching)**.** Given an instance $\mathcal{I} = (\delta, p_U, p_W)$, where δ is the amount of missing information and p_U, p_W are the preferences submitted by men and women, respectively, compute the set D of minimum cardinality such that the instance $\mathcal{I}_{-D} = (\delta_{-D}, p_{U\setminus D}, p_{W\setminus D})$, where $\delta_{-D} = \frac{1}{|(U\cup W)\setminus D|}\sum_{i\in(U\cup W)\setminus D}\delta_i$, has a perfect super-stable matching (i.e., every agent in $(U \cup W) \setminus D$ is matched in a super-stable matching).

Below we first show a 2-approximation for the min-delete-super-stable-matching problem when restricted to the case of one-sided top-truncated preferences. Subsequently, we use this result in order to get an $O(n)$-approximation for our problem. However, before that, we introduce the following terminology which will be used throughout in this section.

- An instance $\mathcal{I}$ of the min-delete-super-stable-matching problem can also be thought of as the set of agents along with their preference lists. Initially for every agent this list has all the agents in the other set listed in some order. Now, during the course of our algorithm sometimes we use the operation "delete(a, b)" which removes agent a from b's list and b from a's. After such a deletion (or after a series of such deletions) our instance now refers to the set of agents along with their updated lists.

- We say that a matching μ is *internally super-stable* with respect to an instance $\mathcal{I}$ if μ is super-stable with respect to the instance that is obtained by only considering the matched agents in μ (i.e., consider $\mathcal{I}$ and remove all the agents who are not matched in μ from $\mathcal{I}$).

Procedure: proposeWith($A, \mathcal{I}$)

assign each agent $a \in A$ to be free
while some $a \in A$ is free **do**
  $b \leftarrow$ first agent on a's list
  if b is already engaged to agent p && b finds p and a incomparable **then**
    delete (a, b)
  else
    if b is already engaged to agent p, then assign p to be free
    assign a and b to be engaged
    for each agent c in b's list such that $a \succ_b c$, delete (c, b)
  end if
end while
for each man m **do**
  $w \leftarrow$ first woman on m's list
  if there exists m' such that w finds m and m' incomparable, then delete (m', w)
end for ▷ deletions in this loop only happen once and removes all the remaining ties
return $\mathcal{I}$ ▷ this returns the updated lists

Main:
Input: a one-sided top-truncated instance $\mathcal{I} = (\delta, p_U, p_W)$
$\mathcal{I}' \leftarrow$ proposeWith($U, \mathcal{I}$)
$\mathcal{I}' \leftarrow$ proposeWith($W, \mathcal{I}'$)
while there exists some exposed rotation $(m_1, w_1), (m_2, w_2), \ldots, (m_r, w_r)$ in $\mathcal{I}'$ **do**
  for all $i \in \{1, \ldots, r\}$, delete (m_i, w_i)
  $\mathcal{I}' \leftarrow$ proposeWith($U, \mathcal{I}'$)
  $\mathcal{I}' \leftarrow$ proposeWith($W, \mathcal{I}'$)
end while
$\mu \leftarrow$ for all men $m \in U$, match m with the only woman in his list
construct $G = (U \cup W, E)$, where $(m, w) \in E$ if (m, w) is super-blocking pair in μ w.r.t. $\mathcal{I}$
$D \leftarrow$ minimum vertex cover of G
for each $a \in D$ **do**
  $D \leftarrow D \cup \mu(a)$
end for
return (D, μ)

Algorithm 1: For the case of one-sided top-truncated preferences, the set D returned by the algorithm is a 2-approximation for the min-delete-super-stable-matching problem and the matching μ returned is an $O(n)$-approximation for δ-min-bp-super-stable-matching.

- We say that an instance $\mathcal{I}$ with no ties has an exposed rotation $\rho = (m_1, w_1)$,

$\ldots, (m_r, w_r)$ if, in $\mathcal{I}$, w_i is the first agent in m_i's list and w_{i+1} is the second agent in m_i's list (here $(i+1)$ is done modulo r).

Proposition 6. *Algorithm 1 is a polynomial-time 2-approximation algorithm for the min-delete-super-stable-matching problem when restricted to the case of one-sided top-truncated preferences.*

Proof. The main idea for Algorithm 1 is inspired by the work of Tan [Tan90] who looked at the problem of finding the maximum internally stable matching for the stable roommates problem (which is equivalent to the problem of finding the minimum number of agents to delete so that the rest of the agents will have a stable matching when the instance is just restricted to themselves). Informally, at a very high level, the key idea in Tan's algorithm was to show that some of the entries in each agent's list can be deleted by running the proposal-rejection sequence like in Gale-Shapley algorithm and through rotation eliminations, while at the same time maintaining at least one solution of the maximum size. As we will see below, this is essentially what we do here as well, adapting this idea as necessary for our case when there are ties on one side but only at the bottom.

Before we go on to the main lemmas, let us suppose that $\mathcal{I} = (\delta, p_U, p_W)$ is an arbitrary instance of the min-delete-super-stable-matching problem when restricted to the case of one-sided top-truncated preferences, where δ is the amount of missing information, p_U, p_W are the preference orders submitted by the men and women, respectively, and $|U| = |W| = n$. Also, let D_{opt} be the optimal solution for this instance. This in turn implies that we can form a perfect and internally super-stable matching of size $k = n - \frac{D_{opt}}{2}$, and that in fact k is the maximum size of any such matching (as otherwise D_{opt} cannot be optimal). Next, for now, let us assume the correctness of the following lemmas (note that all the instances we talk about in this section are restricted to the case of one-sided top-truncated preferences). Their proofs appear in Appendix A.1.

Lemma 7. *Let $\mathcal{I}_1$ denote some instance and $\mathcal{I}_2$ denote the instance returned by the procedure proposeWith$(A, \mathcal{I}_1)$, where the set A represents the proposing side. If there exists a matching of size t in $\mathcal{I}_1$ that is internally super-stable with respect to $\mathcal{I}$, then there exists a matching of size t in $\mathcal{I}_2$ that is internally super-stable with respect to $\mathcal{I}$.*

Lemma 8. *Let $\mathcal{I}_1$ denote some instance that does not contain any ties, $(m_1, w_1), (m_2, w_2), \ldots, (m_r, w_r)$ be a rotation that is exposed in $\mathcal{I}_1$, and $\mathcal{I}_2$ be the instance that is obtained by deleting the entries (m_i, w_i) for all $i \in \{1, \ldots, r\}$ from $\mathcal{I}_1$. If there exists a matching of size t in $\mathcal{I}_1$ that is internally super-stable with respect to $\mathcal{I}$, then there exists a matching of size t in $\mathcal{I}_2$ that is internally super-stable with respect to $\mathcal{I}$.*

Lemma 9. *If $\mathcal{I}_1$ is an instance that does not contain any exposed rotation, then the list of every man in $\mathcal{I}_1$ has only one woman and vice versa.*

Given the above lemmas and given the fact that the instance $\mathcal{I}$ has an internally super-stable matching of size $k = n - \frac{D_{opt}}{2}$, we can start with the instance $\mathcal{I}$ and repeatedly apply Lemma 7 and see that the instance $\hat{\mathcal{I}}$ that we obtain after line 18 of Algorithm 1 has a matching of size k that is internally super-stable matching with respect to $\mathcal{I}$. Next, starting with the instance $\hat{\mathcal{I}}$, we can again repeatedly apply Lemma 8 and see that the instance $\mathcal{I}'$ that remains after line 23 of Algorithm 1 also has a matching of size k that is internally super-stable matching with respect to $\mathcal{I}$. Additionally, from Lemma 9 we know that in this instance each man has only one woman in his list and vice versa. So, now, consider the matching μ that can be obtained by matching each man with the only woman in his list. Next, consider the bipartite graph $G = (V, E)$ where $V = U \cup W$ and $(m, w) \in E$ if (m, w) is a super-blocking pair with respect to the original instance $\mathcal{I}$, and consider a minimum vertex cover C of G. We show below that $k \leq n - \frac{|C|}{2}$.

Suppose this is false and that $\mathcal{I}$ has an internally super-stable matching of size greater than $n - \frac{|C|}{2}$. Now, from the discussion above, we know that this implies $\mathcal{I}'$ also has a matching of size greater than $n - \frac{|C|}{2}$ such that it is internally super-stable with respect to $\mathcal{I}$. This in turn implies that we can remove less than $|C|$ agents and have a matching that is internally super-stable with respect to $\mathcal{I}$. That is, we can remove less than $|C|$ agents and also at the same time ensure that for every $(m, w) \in E$, this matching has only one of m or w matched in it, for if otherwise it will not be internally super-stable with respect to $\mathcal{I}$. However, this implies that $|C|$ is not the size of the minimum vertex cover of G, and hence we have a contradiction.

Now, to get our approximation bound, consider the set D that is returned by the algorithm. We know that $D \leq 2|C| \leq 4(n - k) = 2D_{opt}$, where the first inequality arises because of lines 27–29 in Algorithm 1 and the second inequality uses the observation above that $k \leq n - \frac{|C|}{2}$. Finally, it is easy to see that all the steps can be done in polynomial-time (it is well-known through the Kőnig's Theorem that one can find a minimum vertex cover of a bipartite graph in polynomial-time). $\square$

Given Proposition 6, we can now prove the following theorem.

Theorem 10. *For any $\delta > 0$, Algorithm 1 is a polynomial-time $O(n)$-approximation algorithm for the δ-min-bp-super-stable-matching problem when restricted to the case of one-sided top-truncated preferences. Moreover, the $O(n)$-approximate matching that is returned is also weakly-stable.*

Proof. Consider an arbitrary instance $\mathcal{I}$ of the δ-min-bp-super-stable-matching problem when restricted to the case of one-sided top-truncated preferences. Let μ_{opt} be the optimal solution associated with $\mathcal{I}$. Next, consider the same instance $\mathcal{I}$ for the min-delete-super-stable-matching problem and let us consider the matching μ that is returned by Algorithm 1 for this instance. Also, let D_{opt} be the optimal solution of the min-delete-super-stable-matching problem for the instance $\mathcal{I}$ and D be the set that is returned by Algorithm 1. (We can assume throughout that $D_{opt} \geq 1$, for if otherwise this implies that it has a super-stable matching, and as mentioned in Section 2.2.1 we do not consider such instances.) First, it is easy to see that this matching is weakly-stable (because in every instance that results after the first proposal-rejection sequence (which is in line 23), a matching that is formed by matching each man with the first woman on his list will be weakly-stable). Second, note that we can rewrite μ as $\mu = \mu_1 \cup \mu_2$, where $\mu_1 = \mu \setminus \{\cup_{a \in D}(a, \mu(a))\}$ and $\mu_2 = \cup_{a \in D}(a, \mu(a))$. Now, if S_1 (respectively, S_2) denotes the number of super-blocking pairs associated with men in μ_1 (respectively, μ_2), then

$$\begin{aligned}
|\text{super-bp}(\mu)| &= S_1 + S_2 \\
&\leq \left(n - \frac{|D|}{2}\right) \cdot \frac{|D|}{2} + \frac{|D|}{2} \cdot n \\
&\leq n \cdot |D| \\
&\leq 2n \cdot |D_{opt}|, \qquad (2.8)
\end{aligned}$$

where the second step is using the fact that the men in μ_1 can form at most $\frac{|D|}{2}$ super-blocking pairs with women outside of μ_1 and the men in μ_2 can in the worst case form a blocking pair with all the women, and the last step is using the fact that $D_{opt} \geq 1$ and that $|D| \leq 2 \cdot |D_{opt}|$, which we know is true by Proposition 6.

Also, if μ_{opt} is the optimal solution for the δ-min-bp-super-stable-matching problem, then we know that

$$|\text{super-bp}(\mu_{opt})| \geq \frac{|D_{opt}|}{2}, \qquad (2.9)$$

as otherwise one can delete all the men who are involved in super-blocking pairs in μ_{opt} and their corresponding partners and get a super-stable matching on the remaining agents.

Finally, combining (2.8) and (2.9) gives us our theorem. □

Before we end this section, we address one final question as to whether, for the class of one-sided top-truncated preferences, one can obtain a better approximation result if one

Men	Women
$m_1 : w_2 \succ w_1 \succ [\ldots]$	$w_1 : m_1 \succ \ldots \succ m_{n-y} \succ (m_{n-y+1}, \ldots, m_n)$
$m_2 : w_3 \succ w_1 \succ w_2 \succ [\ldots]$	$w_2 : m_1 \succ [\ldots]$
$m_3 : w_4 \succ w_1 \succ w_3 \succ [\ldots]$	$w_3 : m_2 \succ [\ldots]$
$\vdots$	$w_4 : m_3 \succ [\ldots]$
$m_{n-y} : w_{n-y+1} \succ [\ldots]$	$\vdots$
$m_{n-y+1} : w_{n-y+2} \succ w_1 \succ w_{n-y+1} \succ [\ldots]$	$w_{n-y+1} : m_{n-y} \succ [\ldots]$
$m_{n-y+2} : w_{n-y+2} \succ w_1 \succ w_2 \succ [\ldots]$	$w_{n-y+2} : m_{n-y+1} \succ m_{n-y+2} \succ [\ldots]$
$m_{n-y+3} : w_{n-y+2} \succ w_1 \succ w_{n-y+3} \succ [\ldots]$	$w_{n-y+3} : m_{n-y+3} \succ [\ldots]$
$\vdots$	$w_{n-y+4} : m_{n-y+4} \succ [\ldots]$
$m_n : w_{n-y+2} \succ w_1 \succ w_n \succ [\ldots]$	$\vdots$
	$w_n : m_n \succ [\ldots]$

Figure 2.2: The instance $\mathcal{I}$ that is used in the proof of Theorem 11

continues to consider only weakly-stable matchings. In the theorem below we show that for $\delta \in \Omega(\frac{1}{n})$ Algorithm 1 is asymptotically the best one can do under this restriction.

Theorem 11. *For $\delta \leq \frac{1}{2}$, if there exists an α-approximation algorithm for δ-min-bp-super-stable-matching that always returns a matching that is weakly-stable for the case of one-sided top-truncated preferences, then $\alpha \in \Omega\left(\min\left\{n^{\frac{3}{2}}\sqrt{\delta}, n\right\}\right)$.*

Proof. Consider the instance $\mathcal{I}$ defined in Figure 2.2, where $y = \min\left\{\lfloor 2n^{\frac{3}{2}}\sqrt{\delta}\rfloor, n\right\}$, the presence of $[\ldots]$ in a preference list implies that the rest of the alternatives can be placed in some strict order, and $(m_{n-y+1}, \ldots, m_n)$ implies that these agents are tied with respect to w_1. Below we show that the statement of the theorem is true when $\delta \leq \frac{1}{2n}$, i.e., show that in this case $\alpha \in \Omega\left(n^{\frac{3}{2}}\sqrt{\delta}\right)$. For $\delta > \frac{1}{2n}$ it is then trivial to extend our instance by just having some of the other women (other than w_1) specify partial preferences.

First, it is easy to verify that this instance has at most δ amount of information missing. Second, one can see that the optimal solution, i.e., the matching with the minimum number of super-blocking pairs, for this instance is

$$\mu_{opt} = \big\{(m_1, w_1), (m_2, w_3), \ldots, (m_{n-y+1}, w_{n-y+2}), (m_{n-y+2}, w_2),$$

$$(m_{n-y+3}, w_{n-y+3}), \ldots, (m_n, w_n)\},$$

where (m_1, w_2) is the only super-blocking pair.

Given the above observation, let us now consider an arbitrary matching μ that is weakly-stable. Since m_i prefers w_{i+1} the most and vice versa for all $i \in \{1, \ldots, n-y+1\}$, we know that $\mu(m_i) = w_{i+1}$. Hence, $\mu(w_1) = m_k$, for some $k \in \{n-y+2, \ldots, n\}$. Additionally, we also know that for all $j \in \{n-y+2, \ldots, n\}$ such that $j \neq k$, $w_1 \succ_{m_j} \mu(m_j)$, as none of these can men can be matched to w_{n-y+2}. This in turn implies that since w_1 finds m_k and m_j incomparable for $j \in \{n-y+2, \ldots, n\}$ such that $j \neq k$, (m_j, w_1) is a super-blocking pair. Therefore, in any weakly-stable matching μ we have $(y-2) \in O(n^{\frac{3}{2}}\sqrt{\delta})$ super-blocking pairs. □

2.4 Beyond Weak-Stability

In the previous section we investigated weakly-stable matchings and showed several results for the case when the output matching is weakly-stable. Here we move away from this restriction and explore what happens when we do not place any restriction on the matchings. In particular, we begin this section by showing a general hardness of approximation result, and then follow it with a discussion on one possible approach that can lead to a near-tight approximation result.

2.4.1 Inapproximability of δ-min-bp-super-stable-matching

We show a hardness of approximation result for the δ-min-bp-super-stable-matching problem through a reduction from the Vertex Cover (VC) problem, which is a well-known NP-complete problem [Kar72]. In the VC problem, we are given a graph $G = (V, E)$, where $V = \{v_1, \ldots, v_k\}$, and a $k_0 \leq k$ and are asked if there exists a subset of the vertices with size less than or equal to k_0 such that it contains at least one endpoint of every edge. When given an instance $\mathcal{I}$ of VC, the key idea in the proof is to create an instance $\mathcal{I}'$ of δ-min-bp-super-stable-matching such that if $\mathcal{I}$ is a "yes" instance of VC, then $\mathcal{I}'$ will have a very small number of super-blocking pairs, and if otherwise, then $\mathcal{I}'$ will have a large number of super-blocking pairs.

Theorem 12. *For any constant $\epsilon \in (0, 1]$ and $\delta \in (0, 1)$, one cannot obtain a polynomial-time $(n\sqrt{\delta})^{1-\epsilon}$ approximation algorithm for the δ-min-bp-super-stable-matching problem unless P = NP.*

Proof. The proof here is similar to the one by Hamada, Iwama, and Miyazaki [HIM16, Theorem 1]. The main difference is in the construction of the instance, which in our case is more involved.

Given an instance $\mathcal{I} = (G = (V, E), k_0)$ of the VC problem, where $|V| = k$, we construct the following instance $\mathcal{I}'$ of the δ-min-bp-super-stable-matching problem, where

- $d = \lceil \frac{8}{\epsilon} \rceil$, $y = k^d + 1$, $z = \lceil \frac{1}{\sqrt{\delta}} \rceil$
- $M_{A_1} = \{m_1, \ldots, m_{k_0}\}$, $M_{A_2} = \{m_{k_0+1}, \ldots, m_k\}$, $W_A = \{w_1, \ldots, w_k\}$
- for every $i < j$ such that $(v_i, v_j) \in E$ and $c \in \{1, \ldots, z\}$, $S_c^{i,j} = \{s_{c,1}^{i,j}, \ldots s_{c,y}^{i,j}\}$, $T_c^{i,j} = \{t_{c,1}^{i,j}, \ldots t_{c,y}^{i,j}\}$, $P_c^{i,j} = \{p_{c,1}^{i,j}, \ldots p_{c,y}^{i,j}\}$, and $V_c^{i,j} = \{v_{c,1}^{i,j}, \ldots v_{c,y}^{i,j}\}$
- $S^{i,j} = \{S_1^{i,j}, \ldots S_z^{i,j}\}, T^{i,j} = \{T_1^{i,j}, \ldots T_z^{i,j}\}, P^{i,j} = \{P_1^{i,j}, \ldots P_z^{i,j}\}$, and $V^{i,j} = \{V_1^{i,j}, \ldots V_z^{i,j}\}$
- $M_A = M_{A_1} \cup M_{A_2}$, $S = \bigcup S^{i,j}$, $T = \bigcup T^{i,j}$, $P = \bigcup P^{i,j}$, and $V = \bigcup V^{i,j}$
- $U = M_A \cup S \cup P$ and $W = W_A \cup T \cup V$
- for each i, j the preference orders of the agents are as given in Figure 2.3. For an agent a, if R_S appears in its preference list for some R and S, then this implies that a finds all the agents in R_S as incomparable. Also, $[\ldots]$ denotes that the rest of the agents can be placed in any order.

Note that $n = |U| = |W| = k + 2yz|E|$. Also, the amount of missing information per agent is at most $\frac{\binom{k}{2}+\binom{y}{2}}{\binom{n}{2}} \leq \frac{\binom{k+y}{2}}{\binom{n}{2}}$. Therefore, the average amount of missing information is $\leq \frac{\binom{k+y}{2}}{\binom{n}{2}} \leq \frac{\binom{2y}{2}}{\binom{n}{2}} \leq \frac{4y^2}{n^2} \leq \frac{4y^2}{4y^2z^2} \leq \delta$. Next, in order to show the correctness, we prove the following claims. Throughout, for every m_i, if m_i is not matched to a woman in W_A, then we call such a pair as a *bad pair*. Additionally, for every $s_{a,b}^{i,j}$, if $s_{a,b}^{i,j}$ is matched to a woman who is outside of the top three women in his list (i.e., for instance, if $s_{1,1}^{i,j}$ is matched with anyone other than $t_{1,1}^{i,j}$, w_i, or $t_{\frac{z}{2}+1,1}^{i,j}$), then we again refer to such pairs as *bad*.

Claim 2. If a matching μ contains a bad pair, then it has at least $y - 1$ super-blocking pairs.

Proof. Consider the case when m_i is matched to a woman w' who is not in W_A. This implies that it at least forms a super-blocking pair with all $w \in V_1^{i,j}$ such that $w \neq w'$. And

Men

$m_i : W_A \succ V_1^{i,j} \succ [\ldots]$

$s_{1,1}^{i,j} : t_{1,1}^{i,j} \succ w_i \succ t_{\frac{z}{2}+1,1}^{i,j} \succ V_1^{i,j} \succ [\ldots]$

$s_{1,2}^{i,j} : t_{1,2}^{i,j} \succ w_i \succ t_{1,3}^{i,j} \succ V_1^{i,j} \succ [\ldots]$

$\vdots$

$s_{1,y}^{i,j} : t_{1,y}^{i,j} \succ w_i \succ t_{2,1}^{i,j} \succ V_1^{i,j} \succ [\ldots]$

$s_{2,1}^{i,j} : t_{2,1}^{i,j} \succ w_i \succ t_{2,2}^{i,j} \succ V_2^{i,j} \succ [\ldots]$

$\vdots$

$s_{2,y}^{i,j} : t_{2,y}^{i,j} \succ w_i \succ t_{\substack{i,j\\3,1}} \succ V_2^{i,j} \succ [\ldots]$

$\vdots$

$\vdots$

$s_{\frac{z}{2},y}^{i,j} : t_{\frac{z}{2},y}^{i,j} \succ w_i \succ t_{1,1}^{i,j} \succ V_{\frac{z}{2}}^{i,j} \succ [\ldots]$

$s_{\frac{z}{2}+1,1}^{i,j} : t_{1,2}^{i,j} \succ w_j \succ t_{\frac{z}{2}+1,2}^{i,j} \succ V_{\frac{z}{2}+1}^{i,j} \succ [\ldots]$

$s_{\frac{z}{2}+1,2}^{i,j} : t_{\frac{z}{2}+1,2}^{i,j} \succ w_j \succ t_{\frac{z}{2}+1,3}^{i,j} \succ V_{\frac{z}{2}+1}^{i,j} \succ [\ldots]$

$\vdots$

$s_{\frac{z}{2}+1,y}^{i,j} : t_{\frac{z}{2}+1,y}^{i,j} \succ w_j \succ t_{\frac{z}{2}+2,1}^{i,j} \succ V_{\frac{z}{2}+1}^{i,j} \succ [\ldots]$

$\vdots$

$\vdots$

$s_{z,y-1}^{i,j} : t_{z,y-1}^{i,j} \succ w_j \succ t_{z,y}^{i,j} \succ V_z^{i,j} \succ [\ldots]$

$s_{z,y}^{i,j} : t_{z,y}^{i,j} \succ w_j \succ t_{\frac{z}{2}+1,1}^{i,j} \succ V_z^{i,j} \succ [\ldots]$

$p_{1,1}^{i,j} : v_{1,1}^{i,j} \succ [\ldots]$

$\vdots$

$\vdots$

$p_{z,y}^{i,j} : v_{z,y}^{i,j} \succ [\ldots]$

Women

$w_i : M_{A_1} \succ S \succ M_{A_2}$

$t_{1,1}^{i,j} : M_A \succ s_{\frac{z}{2},y}^{i,j} \succ S_1^{i,j} \succ [\ldots]$

$t_{1,2}^{i,j} : M_A \succ s_{\frac{z}{2}+1,1}^{i,j} \succ S_1^{i,j} \succ [\ldots]$

$t_{1,3}^{i,j} : M_A \succ s_{1,2}^{i,j} \succ S_1^{i,j} \succ [\ldots]$

$\vdots$

$t_{1,y}^{i,j} : M_A \succ s_{1,y-1}^{i,j} \succ S_1^{i,j} \succ [\ldots]$

$t_{2,1}^{i,j} : M_A \succ s_{1,y}^{i,j} \succ S_2^{i,j} \succ [\ldots]$

$\vdots$

$t_{2,y}^{i,j} : M_A \succ s_{2,y-1}^{i,j} \succ S_2^{i,j} \succ [\ldots]$

$\vdots$

$t_{\frac{z}{2}+1,1}^{i,j} : M_A \succ s_{1,1}^{i,j} \succ S_{\frac{z}{2}+1}^{i,j} \succ [\ldots]$

$t_{\frac{z}{2}+1,2}^{i,j} : M_A \succ s_{\frac{z}{2}+1,1}^{i,j} \succ S_{\frac{z}{2}+1}^{i,j} \succ [\ldots]$

$\vdots$

$\vdots$

$t_{z,y}^{i,j} : M_A \succ s_{z,y-1}^{i,j} \succ S_z^{i,j} \succ [\ldots]$

$v_{1,1}^{i,j} : M_A \succ S_1^{i,j} \succ p_{1,1}^{i,j} \succ [\ldots]$

$\vdots$

$v_{1,y}^{i,j} : M_A \succ S_1^{i,j} \succ p_{1,y}^{i,j} \succ [\ldots]$

$v_{2,1}^{i,j} : M_A \succ S_2^{i,j} \succ p_{2,1}^{i,j} \succ [\ldots]$

$\vdots$

$v_{2,y}^{i,j} : M_A \succ S_2^{i,j} \succ p_{2,y}^{i,j} \succ [\ldots]$

$\vdots$

$\vdots$

$v_{z,y}^{i,j} : M_A \succ S_z^{i,j} \succ p_{z,y}^{i,j} \succ [\ldots]$

Figure 2.3: The instance $\mathcal{I}'$ that is used in the proof of Theorem 12

since every women in $V_1^{i,j}$ finds all the men in M_A as incomparable, therefore we at least have $|V_1^{i,j}| - 1 = y - 1$ super-blocking pairs.

Next, consider the case when there is a man $s_{a,b}^{i,j}$ who is matched to a woman w' who is outside of the top three women in his list. This implies that it at least forms a super-blocking pair with all $w \in V_a^{i,j}$ such that $w \neq w'$. And since we can assume that no women in $V_a^{i,j}$ is matched to a man in M_A (as this would anyway result in $y-1$ super-blocking pairs as proved above) and since all of them find the men in $S_a^{i,j}$ as incomparable, this implies that we have at least $|V_a^{i,j}| - 1 = y - 1$ super-blocking pairs. $\square$

Before we go on to the next claim, for every $i < j$ such that $(v_i, v_j) \in E$, consider the sets $S^{i,j}$ and $T^{i,j}$ and let us the define the following two perfect matchings, $\mu_1^{i,j}$ and $\mu_2^{i,j}$, between $S^{i,j}$ and $T^{i,j}$. The matching $\mu_1^{i,j}$ ($\mu_2^{i,j}$) can be inferred from Figure 2.3 by matching every man in $S^{i,j}$ with the woman coloured red (blue) in his list.

$$\mu_1^{i,j} = \{(s_{1,1}^{i,j}, t_{1,1}^{i,j}), (s_{1,2}^{i,j}, t_{1,2}^{i,j}), \ldots, (s_{\frac{z}{2},y}^{i,j}, t_{\frac{z}{2},y}^{i,j}),$$
$$(s_{\frac{z}{2}+1,1}^{i,j}, t_{\frac{z}{2}+1,2}^{i,j}), (s_{\frac{z}{2}+1,2}^{i,j}, t_{\frac{z}{2}+1,3}^{i,j}), \ldots, (s_{z,y}^{i,j}, t_{\frac{z}{2}+1,1}^{i,j})\}$$

$$\mu_2^{i,j} = \{(s_{1,1}^{i,j}, t_{\frac{z}{2}+1,1}^{i,j}), (s_{1,2}^{i,j}, t_{1,3}^{i,j}), \ldots, (s_{\frac{z}{2},y}^{i,j}, t_{1,1}^{i,j}),$$
$$(s_{\frac{z}{2}+1,1}^{i,j}, t_{1,2}^{i,j}), (s_{\frac{z}{2}+1,2}^{i,j}, t_{\frac{z}{2}+1,2}^{i,j}), \ldots, (s_{z,y}^{i,j}, t_{z,y}^{i,j})\}$$

Claim 3. For every $i < j$ such that $(v_i, v_j) \in E$, $\mu_1^{i,j}$ and $\mu_2^{i,j}$ are the only perfect matchings between $S^{i,j}$ and $T^{i,j}$ that do not include a bad pair. Moreover, both $\mu_1^{i,j}$ and $\mu_2^{i,j}$ have only one super-blocking pair (m, w) such that $m \in S^{i,j}$ and $w \in T^{i,j}$.

Proof. It is easy to observe the first part. As for the second part, note that none of $s_{1,1}^{i,j}, \ldots, s_{\frac{z}{2},y}^{i,j}$ form any super-blocking pair in μ_1 as they are matched to their topmost choices. Also, none of $s_{\frac{z}{2}+1,2}^{i,j}, \ldots, s_{z,y}^{i,j}$ form any super-blocking pairs since the only woman they can form super-blocking pairs with, which are $t_{\frac{z}{2}+1,2}^{i,j}, \ldots t_{z,y}^{i,j}$ respectively, strictly prefers their currently matched partner, which are $s_{\frac{z}{2}+1,2}^{i,j}, \ldots, s_{z,y}^{i,j}$ respectively. Hence, the only super-blocking pair is $(s_{\frac{z}{2}+1,1}^{i,j}, t_{1,2}^{i,j})$. We can make similar arguments with respect to μ_2 to show that $(s_{1,1}^{i,j}, t_{1,1}^{i,j})$ is the only super-blocking pair. $\square$

Given the two claims above, we can now prove the correctness of the reduction through the following lemmas.

Lemma 13. *If $\mathcal{I} = (G, k_0)$ is a "yes" instance of VC, then $\mathcal{I}'$ has a solution with at most $2k^2$ super-blocking pairs.*

Proof. Let the vertex cover of G be C and since it is a "yes" instance, we know that $|C| \le k_0$. If the size of C is strictly less than k_0, then add arbitrary vertices to it in order to make its size k_0. So from now on we assume that $|C| = k_0$. Next, construct the following matching μ for the instance $\mathcal{I}'$.

- For every woman $w_i \in W_A$, if $v_i \in C$, then match w_i with some man in M_{A_1}. Otherwise, match w_i with some man in M_{A_2}.
- For every $i < j$ such that $(v_i, v_j) \in E$, if $v_i \in C$, then match every man in $S^{i,j}$ with a woman in $T^{i,j}$ using $\mu_2^{i,j}$ as defined above. Otherwise, match every man in $S^{i,j}$ with a woman in $T^{i,j}$ using $\mu_1^{i,j}$ as defined above.
- For every $i < j$ such that $(v_i, v_j) \in E$, match $p_{a,b}^{i,j}$ with $v_{a,b}^{i,j}$.

Now, we know that the each man in M_A can form at most k super-blocking pairs (one with each woman in W_A). Additionally, we know from Claim 3 that both $\mu_1^{i,j}$ and $\mu_2^{i,j}$ have at most one super-blocking pair, and that none of the men in P form any super-blocking pair as they all get their topmost choice. Hence, the total number of blocking pairs is at most $k^2 + |E| \le 2k^2$. □

Lemma 14. *If $\mathcal{I} = (G, k_0)$ is a "no" instance of VC, then every matching for $\mathcal{I}'$ has at least $y - 1$ super-blocking pairs.*

Proof. Here we will show that if there exists a matching μ with less than $y-1$ super-blocking pairs for $\mathcal{I}'$, then $\mathcal{I}$ has a vertex cover of size at most k_0. To see this, consider μ. Since it has less than $y - 1$ blocking pairs, we know from Claim 2 that it does not have any bad pair. This in turn implies that all the men in M_A are matched with a woman in W_A (since all men in M_A have to be attached to a woman in W_A and size of both the sets are equal).

Next, for every $i < j$ such that $(v_i, v_j) \in E$, let us consider the men and women in $S^{i,j}$ and $T^{i,j}$. Since, again, we cannot have any bad pairs, we know that there has to be a perfect matching between these two sets. Additionally, from Claim 3 we know that $\mu_1^{i,j}$ and $\mu_2^{i,j}$ are the only two perfect matchings that have no bad pairs. Now, for an (i, j), if we were using $\mu_1^{i,j}$, then it is easy to see that w_j should be matched with a man in M_{A_1} as otherwise she would form a super-blocking pair with all the men in $\{s_{\frac{z}{2},1}^{i,j}, \ldots, s_{z,y}^{i,j}\}$, thus resulting in at least $zy > y - 1$ super-blocking pairs for μ. Similarly, if we were using $\mu_2^{i,j}$, then w_i should be matched with a man in M_{A_1} as otherwise we would have at least $zy > y - 1$ blocking pairs. Therefore, we have that for each edge $(v_i, v_j) \in E$ at least one of the women w_i or w_j should be matched to a man in M_{A_1}. So, now, if we define $C = \{v_i \mid \mu(w_i) \in M_{A_1}\}$, then we have a vertex cover of size at most k_0 (as size of M_{A_1} is k_0). □

Finally, from Lemmas 13 and 14, we have an inapproximability gap of α, where

$$\alpha \geq \frac{y-1}{2k^2} = \frac{k^d}{2k^2} > \frac{n\sqrt{\delta}}{16k^4} \qquad \text{(since } n = 2yzk^2 + k \leq 8k^{d+2}\tfrac{1}{\sqrt{\delta}}\text{)}$$
$$> \left(n\sqrt{\delta}\right)^{1-\epsilon}. \qquad \text{(since } n = 2yzk^2 + k > 2yz > 2k^d\tfrac{1}{\sqrt{\delta}}\text{)}$$

□

2.4.2 A possible general approach for obtaining a near-tight approximation factor for δ-min-bp-super-stable-matching

While obtaining a general near-tight approximation result for the δ-min-bp-super-stable-matching problem is still open, here we propose a potentially promising direction for this problem. In particular, we demonstrate how solving even a very relaxed version of the min-delete-stable-matching problem will be enough to get an $O(n)$-approximation for δ-min-bp-super-stable-matching in general. Below, we first define the relaxation in question, which we refer to as an (α, β)-approximation to the min-delete-super-stable-matching problem.

Definition 5 ((α, β)-min-delete-super-stable-matching)**.** Given an instance $\mathcal{I} = (\delta, p_U, p_W)$, compute a set D' such that $|D'| \leq \alpha \cdot |D_{opt}|$, where $|D_{opt}|$ is the size of the optimal solution to the min-delete-super-stable-matching for the same instance, and the instance $\mathcal{I}_{-D'} = (\delta_{-D'}, p_{U\setminus D'}, p_{W\setminus D'})$, where $\delta_{-D'} = \frac{1}{|(U\cup W)\setminus D'|}\sum_{i\in(U\cup W)\setminus D'} \delta_i$, has a matching with at most β super-blocking pairs.

Next, we show that an (α, β)-approximation to the min-delete-super-stable-matching problem gives us an $(\alpha n+\beta)$-approximation for δ-min-bp-super-stable-matching. So, in particular, if we have an (α, β)-approximation where α is a constant and $\beta \in O(n)$, then this in turn gives us an $O(n)$-approximation for δ-min-bp-super-stable-matching in general.

Proposition 15. *If there exists an (α, β)-approximation algorithm for the min-delete-super-stable-matching problem, then there exists an $(\alpha n + \beta)$-approximation algorithm for the δ-min-bp-super-stable-matching problem.*

Proof. We can proceed to prove this almost exactly as in the proof of Theorem 10. Here, if D denotes the (α, β)-approximate solution returned by the algorithm, then the only difference is that we define μ_1 to be the matching with the set of agents in $(U \cup W) \setminus D$ such that it has at most β super-blocking pairs (from the definition of the problem we know that such a matching exists) and μ_2 to be an arbitrary matching on the set of agents in D. Once we have this, then we can arrive at the bound by proceeding exactly as in the proof of Theorem 10, with the only difference being that here we would use S_1, which is the number of super-blocking pairs associated with μ_1, to be equal to $\left(n - \frac{|D|}{2}\right) \cdot \frac{|D|}{2} + \beta$. $\square$

2.5 Discussion

The focus of this chapter was on working with partial information in the context of two-sided matching and in particular to investigate *i)* what makes a matching "good" in this context and *ii)* to better understand the trade-off between the amount of missing information and the quality of different matchings. Towards this end, we introduced a measure for accounting for missing preference information in an instance, and argued that a natural definition of a "good" matching in this context is one that minimizes the maximum number of blocking pairs with respect to all the possible completions. Subsequently, using an equivalent problem (δ-min-bp-super-stable-matching) we first explored the space of matchings that contained no obvious blocking pairs (i.e., weakly-stable matchings) in order to better understand how missing preference information affected the quality, in terms of approximation with respect to the objective of minimizing the number of super-blocking pairs. Later on, by expanding the space of matchings we considered (i.e., removing the restriction that matches must be weakly-stable), we asked whether it was possible to improve on the approximation factors that were achieved under the restriction to weakly-stable matchings.

There are a number of interesting directions for future work. First, while in Section 2.4.2 we proposed one possible approach that can lead to near-tight approximations, there may be other approaches that can prove fruitful. Second, given that in many instances a super-stable matching often does not exist, we believe that the min-delete-super-stable-matching problem introduced here is potentially of independent interest, since it determines the maximum number of disjoint pairs of agents that can be matched in order to form a super-stable matching among themselves. Therefore, an open question is to see if one can obtain general results on this problem (or its relaxation defined in Definition 5). In Proposition 6 we saw that a 2-approximation was achievable for the case of one-sided top-truncated preferences and hence it would also be interesting to determine if there are other interesting classes of preferences for which constant-factor approximations are possible. Finally, there are possible extensions, like, for instance, allowing incompleteness—meaning the agents can specify that they are willing to be matched to only a subset of the agents on the other set—that one could consider and ask similar questions like the ones we considered.

Chapter 3

Mechanism Design for Locating a Facility Under Partial Information

3.1 Introduction

In this chapter we continue on the theme of working with incomplete information and we look at a classic problem in *mechanism design*. The field of mechanism design, which originated in economics and is now extensively studied by both economists and computer scientists, considers scenarios where there are multiple self-interested agents who are interacting with a decision-making system, and it aims to design *mechanisms*—which one can essentially think of as algorithms with incentives built-in—where rational agents would act in a socially-desirable way (see Section 3.2 for formal definitions).

A canonical problem in this broad field (see Part II of the book by Nisan et al. [Nis+07] and the references therein for an introduction to (algorithmic) mechanism design) is that of locating a public facility on a real line or an interval, which is often referred to as the *facility location problem*. In the simplest version of this problem, there are n agents, denoted by the set $[n] = \{1, \cdots, n\}$, and each agent $i \in [n]$ has a preferred location x_i for the public facility. The cost of an agent for a facility located at p is given by $C(x_i, p) = |p - x_i|$, the distance from the facility to the agent's ideal location, and the task in general is to locate a facility that minimizes some objective function. The most commonly considered objective functions are *a)* sum of costs for the agents and *b)* the maximum cost for an agent. In the mechanism design version of the problem, the main question is to see if the objective under consideration can be *implemented*, either optimally or approximately, in *weakly-dominant strategies*. Informally, a strategy, which one can think of as a set of actions that an agent takes, is weakly-dominant if the outcome associated with it is at as least good for the agent as the outcome associated with any other strategy, no matter the strategies of the other players. A mechanism (approximately) implements an objective function in weakly-dominant strategies if there is a set of weakly-dominant strategies, one for each agent, where the outcome associated with the mechanism (approximately) optimizes for the objective function (again, formal definitions appear in Section 3.2).

Although the facility location problem has received much attention, with several different variants like extensions to multiple facilities (e.g., [PT13; Lu+10]), looking at alternative objective functions (e.g., [FW13; CFT16]), etc., being extensively studied, the common assumption in this literature is that the agents are always precisely aware of their preferred locations on the real line (or the concerned metric space, depending on which variant is being considered). However, this might not always be the case and it is possible that the agents currently do not have accurate information on their ideal locations, or their preferences in general. To illustrate this, imagine a simple scenario where a city wants to build a school on a particular street (which we assume for simplicity is just a line) and aims to build one at a location that minimizes the maximum distance any of its residents have to travel to reach the school. While each of the residents is able to specify which block they would like the school to be located, some of them are unable to precisely pinpoint where on the block they would like it because, for example, they do not currently have access to information (like infrastructure data) to better inform themselves, or they are simply unwilling to put in the cognitive effort to refine their preferences further. Therefore, instead of giving a specific location x, they end up giving an interval $[a, b]$, intending to say "*I know that I prefer the school to be built between the points a and b, but I am not exactly sure yet as to where I want it.*"

The above described scenario is precisely the one we are concerned about in this chapter. That is, in contrast to the standard setting of the facility location problem, we consider the setting in which agents are uncertain (or partially informed) about their own true locations x_i and the only information they have is that their preferred location $x_i \in [a_i, b_i]$, where $b_i - a_i \leq \delta$ for some parameter δ which models the amount of inaccuracy in the agents' reports. Now, given such partially informed agents, our task is to look at the problem from the perspective of a designer whose goal is to design 'robust' mechanisms under this setting. Here by 'robust' we mean that, for a given performance measure and when considering implementation under an appropriate solution concept, the mechanism should provide good guarantees with respect to this measure for all the possible underlying unknown true locations of the agents. The performance measure we use here is based on the minimax regret solution criterion, which, informally, for a given objective function, S, is an outcome

	Average cost		Maximum cost	
	Upper bound	Lower bound	Upper bound	Lower bound
very weak dominance	$\frac{B}{2}$	$\frac{B}{2}$ [Thm. 18]	$\frac{B}{2}$	$\frac{B}{2}$ [Thm. 24]
minimax dominance	$\frac{3\delta}{4}$ [Thm. 21]	$\frac{\delta}{2}$ (only for mechanisms with finite range) [Thm. 23]	$\frac{B}{4} + \frac{3\delta}{8}$ [Thm. 25]	$\frac{B}{4}$ [GT17, Thm. 5]

Table 3.1: Summary of our results which show the bounds obtained under each of the solution concepts for the objective functions considered. All the bounds are with respect to deterministic mechanisms.

that has the "best worst case", or one that induces the least amount of regret after one realizes the true input (see Section 3.2.2 for a discussion on the choice of regret as the performance measure).

More formally, if $\mathcal{P} = [0, B]$ denotes the set of all points where a facility can be located and $\mathcal{I} = [a_1, b_1] \times \cdots \times [a_n, b_n]$ denotes the set of all the possible vectors that correspond to the true preferred locations of the agents, then the minimax optimal solution, p_{opt}, for some objective function S (like the sum of costs or the maximum cost) is given by

$$p_{opt} = \arg\min_{p \in \mathcal{P}} \underbrace{\max_{I \in \mathcal{I}} \left(S(I, p) - \min_{p' \in \mathcal{P}} S(I, p') \right)}_{\text{maxRegret}(p, \mathcal{I})},$$

where $S(I, p)$ denotes the value of S when evaluated with respect to $I \in \mathcal{I}$ and a point p.

Thus, our aim is to design mechanisms that approximately implement the optimal minimax value (i.e., $\text{maxRegret}(p_{opt}, \mathcal{I})$) with respect to two of the commonly-studied objective functions—average cost and maximum cost—and under two solution concepts—very weak dominance and minimax dominance—that naturally extend to our setting (see Section 3.2 for definitions). In particular, we focus on deterministic and *anonymous* mechanisms, where anonymity entails that the mechanism returns the same output on any permutation of the input (in other words, the agents' identity is irrelevant), and aim to find ones that additively approximate the optimal minimax value. Our results are summarized in Table 3.1.

3.2 Preliminaries

In the standard (mechanism design) version of the facility location problem there are n agents, denoted by the set $[n] = \{1, \ldots, n\}$, and each agent $i \in [n]$ has a true preferred[1] location $\ell_i^* \in [0, B]$, for some fixed constant $B \in \mathbb{R}$.[2] A vector $I = (\ell_1, \ldots, \ell_n)$, where $\ell_i \in [0, B]$, is referred to as a location profile and the cost of agent i for a facility located at p is given by $C(\ell_i^*, p) = |p - \ell_i^*|$ (or equivalently, their utility is $-|p - \ell_i^*|$), the distance from the facility to the agent's location.[3] In general, the task in the facility location problem is to design mechanisms—which are, informally, functions that map location profiles to a point (or a distribution over points) in $[0, B]$—that (approximately) implement the outcome associated with a particular objective function.

In the version of the problem that we are considering, each agent i, although they have a true location $\ell_i^* \in [0, B]$, is currently unaware of their true location and instead only knows an interval $[a_i, b_i] \subseteq [0, B]$ such that $\ell_i^* \in [a_i, b_i]$. The interval $[a_i, b_i]$, which we denote by K_i, is referred to as the *candidate locations* of agent i, and we use $\mathbb{K}_i$ to denote the set of all possible candidate locations of agent i (succinctly referred to as the set of candidate locations). Now, given a profile of the set of candidate locations $(\mathbb{K}_1, \ldots, \mathbb{K}_n)$, we have the following definition.

Definition 6 (δ-uncertain-facility-location-game)**.** For all $n \geq 1$, $B > 0$, and $\delta \in [0, B]$, a profile of the set of candidate locations $(\mathbb{K}_1, \ldots, \mathbb{K}_n)$ is said to induce a δ-uncertain-facility-location-game if, for each i, $\mathbb{K}_i = \{[a_i, b_i] \mid b_i - a_i \leq \delta \text{ and } [a_i, b_i] \subseteq [0, B]\}$ (or in words, for each agent i, their set of candidate locations can only have intervals of length at most δ).

Remark: We refer to δ as the inaccuracy parameter. In general, when proving lower bounds we assume that the designer knows this δ as this only makes our results stronger, whereas for positive results we explicitly state what the designer knows about δ. Additionally, note that in the definition above if $\delta = 0$, then we have the standard facility location setting where the set of candidate locations associated with every agent is just a set of points in $[0, B]$. For a given profile of candidate locations $(K_1, \ldots, K_n)$, we say that "the reports are exact" when, for each agent i, K_i is a single point and not an interval.

[1]We often omit the term 'preferred' and instead just say that ℓ_i^* is agent i's location.

[2]Note that here we make the assumption that the domain under consideration is bounded instead of assuming that the agents can be anywhere on the real line. This is necessary only because we are focusing on additive approximations instead of the usual multiplicative approximations. (For a slightly more elaborate explanation, see the introduction section of the paper by Golomb and Tzamos [GT17].)

[3]This particular utility function that is considered here is equivalent to the notion of symmetric single-peaked preferences that is often used in the economics literature (see, e.g., [MD11]).

3.2.1 Mechanisms, solution concepts, and implementation

A (deterministic) mechanism $\mathcal{M} = (X, F)$ in our setting consists of an action space $X = (X_1, \ldots, X_n)$, where X_i is the action space associated with agent i, and an outcome function F which maps a profile of actions to an outcome in $[0, B]$ (i.e., $F\colon X_1 \times \ldots \times X_n \to [0, B]$). A mechanism is said to be *direct* if, for all i, $X_i = \mathbb{K}_i$, where $\mathbb{K}_i$ is the set of all possible candidate locations of agent i. For every agent i, a strategy is a function $s_i\colon \mathbb{K}_i \to X_i$, and we use Σ_i and $\Delta(\Sigma_i)$ to respectively denote the set of all pure and mixed strategies of agent i.

Since the outcome of a mechanism needs to be achieved in equilibrium, it remains to be defined what equilibrium solution concepts we consider. Below we define, in the order of their relative strengths, the two solution concepts that we use here. We note that the first (very weak dominance) was also used by Chiesa, Micali, and Allen-Zhu [CMA12] in the context of auctions.

Definition 7 (very weak dominance)**.** In a mechanism $\mathcal{M} = (X, F)$, an agent i with candidate locations K_i has a very weakly dominant strategy $s_i \in \Sigma_i$ if $\forall s'_i \in \Sigma_i, \forall \ell_i \in K_i$, and $\forall s_{-i} \in \Sigma_{-i}$,

$$C\left(\ell_i, F(s_i(K_i), s_{-i}(K_{-i}))\right) \leq C\left(\ell_i, F(s'_i(K_i), s_{-i}(K_{-i}))\right).$$

In words, the definition above implies that for agent i with candidate locations K_i, it is always best for i to play the strategy s_i, irrespective of the actions of the other players and irrespective of which of the points in K_i is her true location.

Definition 8 (minimax dominance)**.** In a mechanism $\mathcal{M} = (X, F)$, an agent i with candidate locations K_i has a minimax dominant strategy $s_i \in \Sigma_i$ if $\forall s'_i \in \Sigma_i$ and $\forall s_{-i} \in \Sigma_{-i}$,

$$\begin{aligned}\max_{\ell_i \in K_i} \max_{\sigma_i \in \Delta(\Sigma_i)} & C(\ell_i, F(s_i(K_i), s_{-i}(K_{-i})) - C(\ell_i, F(\sigma_i(K_i), s_{-i}(K_{-i}))) \\ & \leq \max_{\ell_i \in K_i} \max_{\sigma_i \in \Delta(\Sigma_i)} C(\ell_i, F(s'_i(K_i), s_{-i}(K_{-i})) - C(\ell_i, F(\sigma_i(K_i), s_{-i}(K_{-i})).\end{aligned}$$

Before we explain what the definition above implies, let $p = F(s_i(K_i), s_{-i}(K_{-i}))$ be the outcome of the mechanism when agent i plays strategy s_i and all the others play some s_{-i}. Now, let us consider the term

$$\text{maxRegret}_i(p) = \max_{\ell_i \in K_i} \max_{\sigma_i \in \Delta(\Sigma_i)} C(\ell_i, p) - C(\ell_i, F(\sigma_i(K_i), s_{-i}(K_{-i}))), \tag{3.1}$$

which calculates agent i's maximum regret (i.e., the absolute worst case loss agent i will experience if and when she realizes her true location from her candidate locations) for playing s_i and hence getting the output p. Then, what the above definition implies is that for a regret minimizing agent i with candidate locations K_i, it is always best for i to play s_i, irrespective of the actions of the other players, as any other strategy s'_i results in an outcome p' with respect to which agent i experiences at least as much maximum regret as she experiences with p.

Remark: Note that both the solution concepts defined above can be seen as natural extensions of the classical (i.e., the usual mechanism design setting where the agents know their types exactly) weak dominance notion to our setting. That is, for all $i \in [n]$, if K_i is a single point, then both of them collapse to the classical weak dominance notion.

As stated previously, for a profile of candidate locations $(K_1, \ldots, K_n)$ and some objective function S, we want mechanisms that "perform well" against all the possible underlying true locations of the agents, i.e., with respect to all the location profiles $I = (\ell_1, \ldots, \ell_n)$ where $\ell_i \in K_i$. Before we define what "perform well" means formally, we first define the maximum regret associated with a point $p \in [0, B]$.

Given an instance $\mathcal{I} = K_1 \times \ldots \times K_n$ and an objective function S, if $S(I, p)$ denotes the value of the function S when evaluated with respect to the vector $I \in \mathcal{I}$ and the point p, then

$$\text{maxRegret}(p, \mathcal{I}) = \max_{I \in \mathcal{I}} \left(S(I, p) - \min_{p' \in [0,B]} S(I, p') \right). \tag{3.2}$$

Throughout, we refer to the point p_{opt} as the optimal minimax solution for the instance $\mathcal{I}$, where

$$p_{opt} = \arg\min_{p \in [0,B]} \text{maxRegret}(p, \mathcal{I}). \tag{3.3}$$

Equipped with this, we can now formally state our objective as trying to find mechanisms that achieve a good approximation to the optimal minimax value, which for an instance $\mathcal{I}$ is denoted by $\text{OMV}_S(\mathcal{I})$ and is defined as

$$\text{OMV}_S(\mathcal{I}) = \text{maxRegret}(p_{opt}, \mathcal{I}). \tag{3.4}$$

Finally, now that we have our performance measure, we define implementation in very weakly dominant and minimax dominant strategies.

Definition 9 (Implementation in very weakly dominant (minimax dominant) strategies)**.** For a δ-uncertain-facility-location-game, we say that a mechanism $\mathcal{M} = (X, F)$ implements α-OMV_S, for some $\alpha \geq 0$ and some objective function S, in very weakly dominant (minimax dominant) strategies, if for some $s = (s_1, \ldots, s_n)$, where s_i is a very weakly dominant (minimax dominant) strategy for agent i with candidate locations K_i,

$$\text{maxRegret}(F(s_1(K_1), \ldots s_n(K_n)), \mathcal{I}) - \text{OMV}_S(\mathcal{I}) \leq \alpha.$$

3.2.2 Some Q & A on the definitions

Why regret? As stated above, our performance measure is based on minimizing the maximum regret. We argue why this is a good measure by considering some alternatives.

1. Perhaps one of the first approaches that comes to mind is to see if we can, for every possible input $I \in \mathcal{I}$, bound the ratio of the objective values of *a)* the outcome that is returned by the mechanism and *b)* the optimal outcome for that input. For instance, this is the approach taken by Chiesa, Micali, and Allen-Zhu [CMA12] in the case of single good auctions. However, here this is not a good measure because we can quickly see that this ratio is always unbounded if there exists a point that is in the candidate locations of all the agents (i.e., if there is a $p \in [0, B]$ such that for all $i \in [n], p \in [a_i, b_i]$).

2. Another natural approach is to show that for all possible inputs we can bound the difference between the objective values of *a)* the outcome that is returned by the mechanism and *b)* the optimal outcome for that input. For instance, this is the approach taken by Chiesa, Micali, and Allen-Zhu [CMA15]. Technically, this is essentially what we are doing when using regret and finding an answer that has a max. regret that is additively close to the max. regret associated with the minimax optimal solution (one could argue in a similar way even when approximating multiplicatively—i.e., when finding an answer that has a max. regret that is multiplicatively close to the max. regret associated with the minimax optimal solution). The reason why using regret is more informative is because if we were to just mention that, for all $I \in \mathcal{I}$, the point p that is returned by the mechanism satisfies,

$$S(I, p) - S(I, p_I) \leq X,$$

   where X is the bound we obtain, then the only information this conveys is that for every I we are additively at most X-far from the optimal objective value, p_I, for I. However, instead, if we were to write it as

$$\text{maxRegret}(p, \mathcal{I}) - \text{maxRegret}(p_{opt}, \mathcal{I}) \leq Y,$$

where p_{opt} is the minimax optimal solution, then this conveys two things: a) for any point p' there is at least one $I \in \mathcal{I}$ such that $S(I, p') - S(I, p_I) \geq Z$, where $Z = \text{maxRegret}(p_{opt}) =$ optimal minimax value (i.e., it gives us a sense on what is achievable at all—which in turn can be thought of as a natural lower bound) and b) the point p that is chosen by the mechanism is at most $(Y + Z)$-far from the optimal objective value for any $I \in \mathcal{I}$. Hence, to convey these, we employ the notion of regret.

Why additive approximations? Even when working with regret, when it comes to implementing a particular objective using some solution concept, one could potentially aim to find a solution p such that $F = \frac{\text{maxRegret}(p,\mathcal{I})}{\text{maxRegret}(p_{opt},\mathcal{I})}$ is bounded (i.e., use a multiplicative approximation rather than additive). Although this is reasonable, there at at least two issues that become apparent:

1. When considering implementation in very weakly dominant strategies, it turns out that there are no bounded mechanisms when using either of the objective functions (this can be proved by proceeding like in the proof of Theorem 18).

2. When considering the objective of minimizing the maximum cost and minimax dominant strategies, it is clear that minimax dominant strategies are useless to look at as there are no bounded mechanisms. Why? Because suppose there was one. Then this implies that when the reports are exact—meaning every agent reports a single point—the mechanism should always return the optimal solution associated with this location profile, for if otherwise F will not be bounded as minimum maximum regret when valuations are exact is zero. However, given that a minimax dominant mechanism is weakly dominant under exact reports, this in turn implies that we now have a mechanism that implements the optimal solution associated with the max. cost objective in weakly dominant strategies when the reports are exact. But then, we already know that there is no such mechanism due to a result by Procaccia and Tennenholtz [PT13, Theorem 3.2].

Hence, we focus on additive approximations.[4]

[4]In mechanism design, there are other contexts in which additive approximations have been considered. For instance, the work by Golomb and Tzamos [GT17] uses additive approximations in the context of facility location, Caragiannis, Christodoulou, and Protopapas [CCP19] use in the context of impartial selection, and Chiesa, Micali, and Allen-Zhu [CMA15] use it in the context of auctions.

3.3 Related Work

Equipped with the definitions, we can now better discuss related work. There are two broad lines of research that are related to the topic of this chapter. The first is, naturally, the extensive literature that focuses on designing mechanisms in the context of the facility location problem and the second is the work done in mechanism design which considers settings where the agents do not completely specify their preferences.

Designing mechanisms with incomplete preferences. A disproportionate amount of the work in mechanism design considers settings where the agents have complete information about their preferences. Nevertheless, as one might expect, the issue of agents not specifying their complete preferences has been considered and the papers that are most relevant to the topic here are the series of papers by Chiesa, Micali, and Allen-Zhu [CMA12; CMA14; CMA15], and the works of Hyafil and Boutilier [HB07a; HB07b]. Below we briefly discuss each of these papers.

The series of papers by Chiesa, Micali, and Allen-Zhu [CMA12; CMA14; CMA15] considers settings where the agents are uncertain about their own types and they look at this model in the context of auctions. In particular, in their setting the only information agents have about their valuations is that it is contained in a set K, where K is any subset of the set of all possible valuations.[5] Under this setting, Chiesa, Micali, and Allen-Zhu [CMA12] look at single-item auctions and they provide several results on the fraction of maximum social welfare that can be achieved under implementation in very weakly dominant and undominated strategies; subsequently, Chiesa, Micali, and Allen-Zhu [CMA14] study the performance of VCG mechanisms in the context of combinatorial auctions when the agents are uncertain about their own types and under undominated and regret-minimizing strategies; and finally, Chiesa, Micali, and Allen-Zhu [CMA15] analyze the Vickrey mechanism in the context of multi-unit auctions and, again, when the agents are uncertain about their types, and in this case they essentially show that it achieves near-optimal performance (in terms of social welfare) under implementation in undominated strategies. The partial information model that we use here is inspired by this series of papers. In particular, our prior-free and absolute worst-case approach under partial information is similar to the one taken by Chiesa, Micali, and Allen-Zhu [CMA12; CMA14; CMA15] (although such absolute

[5] Chiesa, Micali, and Allen-Zhu [CMA14] argue that their model is equivalent to the Knightian uncertainty model that has received much attention in decision theory (see related works section in [CMA14] and the references therein). However, here we do not use the term Knightian uncertainty, but instead just say that the agents are partially informed. This is because, the notion we use here, which we believe is the natural one to consider in the context of our problem, is less general than the notion of Knightian uncertainty.

worst-case approaches are not uncommon and have been previously considered in many different settings). However, our work is also different from theirs in that, unlike auctions, the problem we consider falls within the domain of *mechanism design without money*—i.e., mechanism design in the context of problems where money cannot be used to incentivize agents—and so their results do not carry over to our setting.

The other set of papers that are most relevant to the broad theme here is the work of Hyafil and Boutilier [HB07a; HB07b] who considered the problem of designing mechanisms that have to make decisions using partial type information. Their focus is again on contexts where payments are allowed and in [HB07a] they mainly show that a class of mechanisms based on the minimax regret solution criterion achieves approximate efficiency under approximate dominant strategy implementation. In [HB07b] their focus is on automated mechanism design within the same framework. While the overall theme in both their works is similar to ours, i.e., to look at issues that arise when mechanisms only have access to partial information, the questions they are concerned with and the model used are different. For instance, in the context of the models used, whereas in ours and Chiesa et al.'s models the agents do not know their true types and are therefore providing partial inputs, the assumption in the works of Hyafil and Boutilier [HB07a; HB07b] is that the mechanism has access to partial types, but agents are aware of their true type. This subtle change in turn leads to the focus being on solution concepts that are different from ours.

In addition to the papers mentioned above, note that another way to model uncertain agents is to assume that each of them has a probability distribution which tells them the probability of a point being their ideal location. For instance, this is the model that is used by Feige and Tennenholtz [FT11] in the context of task scheduling. However, in our model the agents do not have any more information than that they are within some interval, which we emphasize is not equivalent to assuming that, for a given agent, every point in the its interval is equally likely to be its true preferred location.

Related work on the facility location problem. Starting with the work of Moulin [Mou80] there has been a flurry of research looking at designing strategyproof mechanisms (i.e., mechanisms where it is a (weakly) dominant strategy for an agent to reveal her true preferences) for the facility location problem. These can be broadly divided into two branches. The first one consists of work, e.g., [Mou80; BJ94; SV02; MD11; Dok+12], that focuses on characterizing the class of strategyproof mechanisms in different settings (see [Bar01] and [Nis+07, Chapter 10] for a survey on some of these results). The second branch consists of more recent papers which fall under the broad umbrella of *approximate mechanism design without money*, initially advocated by Procaccia and Tennenholtz [PT13], that focus on looking at how well a strategyproof mechanism can perform under different objective functions [PT13; Lu+10; FW13; FT16; FSY16]. Our work here, which looks at the performance of mechanisms under different solution concepts and objective functions when the agents are partially informed about their own locations, falls under this branch of the literature.

3.4 Implementing the Average Cost Objective

In this section we consider the objective of locating a facility so as to minimize the average cost (sometimes succinctly referred to as avgCost and written as AC). While the standard objective in the facility location setting is to minimize the sum of costs, here, like in work of Golomb and Tzamos [GT17], we use average cost because since we are approximating additively, it is easy to see that in many cases a deviation from the optimal solution results in a factor of order n coming up in the approximation bound. Hence, to avoid this, and to make comparisons with our second objective function, maximum cost, easier we use average cost.

In the standard setting where the agents know their true location, the average cost of locating a facility at a point p is defined as $\frac{1}{n}\sum_{i\in[n]} C(x_i, p)$, where x_i is the location of agent i. Designing even optimal strategyproof/truthful mechanisms in this case is easy since one can quickly see that the optimal location for the facility is the median of $x_1, \ldots, x_n$ and returning the same is strategyproof. Note that, for some $k \geq 0$, when $n = 2k + 1$, the median is unique and is the $(k+1)$-th largest element. However, when $n = 2k$, the "median" can be defined as any point between (and including) the $(n/2)$-th and $((n/2)+1)$-th largest numbers. As a matter of convention, here we consider the $(n/2+1)$-th element to be the median. Hence, throughout, we always write that the median element is the $(k+1)$-th element, where $k = \lfloor \frac{n}{2} \rfloor$.

In contrast to the standard setting, for some $\delta \in (0, B]$ and a corresponding δ-uncertain-facility-location-game, even computing what the minimax optimal solution for the average cost objective (see (3.3)) is is non-trivial, let alone seeing if it can be implemented with any of the solution concepts discussed in Section 3.2.1. Therefore, we start by stating some properties about the minimax optimal solution that will be useful when designing mechanisms. A complete discussion on how to find the minimax optimal solution when using the average cost objective, as well as the proofs for the lemmas stated in the next section, are in Appendix B.2.

3.4.1 Properties of the minimax optimal solution for avgCost

Given the candidate locations $K_i = [a_i, b_i]$ for all i, where, for some $\delta \in [0, B]$, $b_i - a_i \leq \delta$, consider the left endpoints associated with all the agents, i.e., the set $\{a_i\}_{i\in[n]}$. We denote the sorted order of these points as $L_1, \dots, L_n$ (throughout, by sorted order we mean sorted in non-decreasing order). Similarly, we denote the sorted order of the right endpoints, i.e., the points in the set $\{b_i\}_{i\in[n]}$, as $R_1, \dots, R_n$. Next, we state the following lemma which gives a succinct formula for the maximum regret associated with a point p (i.e., $\text{maxRegret}(p, \mathcal{I})$, where $\mathcal{I} = [a_1, b_1] \times \ldots \times [a_n, b_n]$; see (3.2)). As stated above, all the proofs for lemmas in the section appear in Appendix B.2.

Lemma 16. *Given a point p, the maximum regret associated with p for the average cost objective can be written as $\max(obj_1^{AC}(p), obj_2^{AC}(p))$, where*

- $obj_1^{AC}(p) = \frac{1}{n}\left(2\sum_{i=j}^{k}(R_i - p) + (n-2k)(R_{k+1} - p)\right)$, *where j is the smallest index such that $R_j > p$ and $j \leq k$*
- $obj_2^{AC}(p) = \frac{1}{n}\left(2\sum_{i=k+2}^{h}(p - L_i) + (n-2k)(p - L_{k+1})\right)$, *where h is the largest index such that $L_h < p$ and $h \geq k+2$.*

Our next lemma states that the minimax optimal solution, p_{opt}, associated with the avgCost objective function is always in the interval $[L_{k+1}, R_{k+1}]$.

Lemma 17. *If p_{opt} is the minimax optimal solution associated with the avgCost objective function, then $p_{opt} \in [L_{k+1}, R_{k+1}]$.*

Equipped with these properties, we are now ready to talk about implementation under the solution concepts defined in Section 3.2.1.

3.4.2 Implementation in very weakly dominant strategies

As discussed in Section 3.2.1, the strongest solution concept that we consider is very weak dominance, where for an agent i, with candidate locations K_i, strategy s_i is very weakly dominant if it is always best for i to play s_i, irrespective of the actions of the other players and irrespective of which of the points in K_i is her true location. While it is indeed a natural solution concept which extends the classical notion of weak dominance, we will see below in, Theorem 18, that it is too strong as no deterministic mechanism can achieve a better approximation bound than $\frac{B}{2}$. This in turn implies that, among deterministic mechanisms, the naive mechanism which always, irrespective of the reports of the agents, outputs the point $\frac{B}{2}$ is the best one can do.

Theorem 18. *Given a $\delta \in (0, B]$, let $\mathcal{M} = (X, F)$ be a deterministic mechanism that implements α-OMV_{AC} in very weakly dominant strategies for a δ-uncertain-facility-location-game. Then, $\alpha \geq \frac{B}{2}$.*

Proof. Let us assume for the sake of contradiction that, for some $\gamma > 0$, $\alpha = \frac{B}{2} - \gamma$. First, note that here we can restrict ourselves to direct mechanisms since Chiesa, Micali, and Allen-Zhu showed that the revelation principle holds with respect to this solution concept [CMA14, Lemma A.2].[6] So, now, let us consider a scenario where the profile of true candidate locations of the agents are $([a_1, b_1], \ldots, [a_n, b_n])$ and let $p = F([a_1, b_1], \ldots, [a_n, b_n])$. Since reporting the true candidate locations is a very weakly dominant strategy in $\mathcal{M}$, this implies that for an agent i and for all $\ell \in [a_i, b_i]$,

$$|\ell - p| \leq |\ell - p'|, \tag{3.5}$$

where for some $[a'_i, b'_i] \neq [a_i, b_i]$, $p' = F([a_1, b_1], \ldots, [a'_i, b'_i], \ldots, [a_n, b_n])$.

Next, consider the profile of true candidate locations $([a_1, b_1], \ldots, [a'_i, b'_i], \ldots, [a_n, b_n])$. Then, again, using the fact that reporting the truth is a very weakly dominant strategy, we have that for agent i and for all $\ell' \in [a'_i, b'_i]$,

$$|\ell' - p'| \leq |\ell' - p|. \tag{3.6}$$

(3.5) and (3.6) together imply that for a $k \in [a_i, b_i] \cap [a'_i, b'_i]$,

$$|k - p| = |k - p'|. \tag{3.7}$$

So, if $Q = [a_i, b_i] \cap [a'_i, b'_i]$ and $|Q| > 1$, then (3.7) implies that $p = p'$.

Now, let us consider $F([a_1, b_1], \ldots, [a_1, b_1])$, where $a_1 = 0$ and $b_1 = \epsilon$, and let $p = F([a_1, b_1], \ldots, [a_1, b_1])$. By repeatedly using the observation made above, we have that for $\delta \in (0, B]$, $\epsilon \in (0, \min\{\delta, \gamma\})$, $\epsilon_1 \in (0, \epsilon)$, and $b_i = \epsilon + i(\delta - \epsilon_1)$,

$$p = F\left([a_1, b_1], \ldots, [a_1, b_1]\right)$$

[6] Informally, revelation principle w.r.t. very weakly dominant strategies states that if a mechanism M implements a social choice function f in very weakly dominant strategies, then there is a direct mechanism M' that implements f and where for every agent reporting their types directly is very weakly dominant strategy.

$$
\begin{aligned}
&= F\left([b_1 - \epsilon_1, b_2], [a_1, b_1], \ldots, [a_1, b_1]\right) \\
&= F([b_2 - \epsilon_1, b_3], [a_1, b_1], \ldots, [a_1, b_1]) \\
&\vdots \\
&= F([B - \epsilon_1, B], [a_1, b_1], \ldots, [a_1, b_1]) \\
&\vdots \\
&= F([B - \epsilon_1, B], \ldots, [B - \epsilon_1, B]). \qquad (3.8)
\end{aligned}
$$

Next, it is easy to see that the minimax optimal solution associated with the profile $([a_1, b_1], \ldots, [a_1, b_1])$ is $\frac{a_1+b_1}{2} = \frac{\epsilon}{2}$, whereas for the profile $([B - \epsilon_1, B], \ldots, [B - \epsilon_1, B])$ it is $B - \frac{\epsilon_1}{2}$. Also, from (3.8) we know that $\mathcal{M}$ outputs the same point p for both these profiles. So, if we assume without loss of generality that $p \leq \frac{B+\epsilon/2-\epsilon_1/2}{2}$, this implies that for $\mathcal{I} = [B - \epsilon_1, B] \times \ldots \times [B - \epsilon_1, B]$,

$$
\begin{aligned}
\alpha &\geq \text{maxRegret}(p, \mathcal{I}) - \text{OMV}_{AC}(\mathcal{I}) \\
&\geq \text{regret}(p, (B, \ldots, B)) - \text{OMV}_{AC}(\mathcal{I}) \\
&\geq (B/2 - \epsilon/4 + \epsilon_1/4) - \epsilon_1/2 \\
&\geq B/2 - \epsilon \\
&> B/2 - \gamma.
\end{aligned}
$$

This in turn contradicts our assumption that $\alpha = \frac{B}{2} - \gamma$. □

Although one could argue that this result is somewhat expected given how Chiesa, Micali, and Allen-Zhu also observed similar poor performance for implementation with very weakly dominant strategies in the context of the single-item auctions [CMA12, Theorem 1], we believe that it is still interesting because not only do we observe a similar result in a setting that is considerably different from theirs, but this observation also reinforces their view that one would likely have to look beyond very weakly dominant strategies in settings like ours. This brings us to our next section, where we consider an alternative, albeit weaker, but natural, extension to the classical notion of weakly dominant strategies.

3.4.3 Implementation in minimax dominant strategies

Here we move our focus to implementation in minimax dominant strategies and explore whether by using this weaker solution concept one can obtain a better approximation bound than the one obtained in the previous section. To this end, we first present a general result that applies to all mechanisms in our setting that are anonymous and minimax dominant, in particular showing that any such mechanism cannot be onto. Following this, we look at non-onto mechanisms and here we provide a mechanism that achieves a much better approximation bound than the one we observed when considering implementation in very weak dominant strategies.

Remark: Note that in this section we focus only on direct mechanisms. This is without loss of generality because, like in the case with very weakly dominant strategies, it turns out that the revelation principle holds in our setting for minimax dominant strategies. A proof of the same can be found in Appendix B.1.

Theorem 19. *Given a $\delta \in (0, B]$, let $\mathcal{M} = (X, F)$ be a deterministic mechanism that is anonymous and minimax dominant for a δ-uncertain-facility-location-game. Then, $\mathcal{M}$ cannot be onto.*

Proof. Suppose this were not the case and there exists a deterministic mechanism $\mathcal{M}$ that is anonymous, minimax dominant, and onto. First, note that if we restrict ourselves to profiles where every agent's report is a single point (instead of intervals as in our setting), then $\mathcal{M}$ must have $n-1$ fixed points $y_1 \leq \ldots \leq y_{n-1}$ such that for any profile of single reports $(x_1, \ldots, x_n)$,

$$\mathcal{M}(x_1, \ldots, x_n) = \text{median}(y_1, \ldots, y_{n-1}, x_1, \ldots, x_n).$$

This is so because, given the fact that $\mathcal{M}$ is anonymous, onto, and minimax dominant in our setting, when restricted to the setting where reports are single points, $\mathcal{M}$ is strategyproof, anonymous, and onto, and hence we know from the characterization result by Massó and De Barreda [MD11, Corollary 2] that every such mechanism must have $n-1$ fixed points $y_1 \leq \ldots \leq y_{n-1}$ such that for any profile $(x_1, \ldots, x_n)$,

$$\mathcal{M}(x_1, \ldots, x_n) = \text{median}(y_1, \ldots, y_{n-1}, p_1, \ldots, p_n),$$

where p_i is the most preferred alternative of agent i (i.e., agent i's peak; since the utility of agent i for an alternative $a \in [0, B]$ is defined as -$|x_i - a|$, we know that the preferences of agent i is symmetric single-peaked with the peak $p_i = x_i$).[7,8]

[7] The original statement by Massó and De Barreda [MD11, Corollary 2] talks about mechanisms that are anonymous, strategyproof, and efficient. However, it is known that for strategyproof mechanisms in (symmetric) single-peaked domains, efficiency is equivalent to being onto (see, e.g., [Nis+07, Lemma 10.1] for a proof).

[8] It is worth noting that the characterization result by Massó and De Barreda [MD11, Corollary 2] for mechanisms that are anonymous, stratgeyproof, and onto under symmetric single-peaked preferences is the same as Moulin's characterization of the set of such mechanisms on the general single-peaked domain [Mou80, Theorem 1]).

Now, given the observation above, for $1 \leq j \leq n-1$, let us consider the smallest j such that $y_j \neq y_{j+1}$ (if there is no such j define $j = n-1$ if $y_{n-1} \neq B$ and $j = 0$ otherwise) and consider the following input profile $\mathcal{L}_0$

$$\left(\underbrace{y_j, \ldots, y_j}_{n-j-1 \text{ agents}}, [\ell, r], z, \underbrace{B, \ldots, B}_{j-1 \text{ agents}}\right),$$

where $y_j < \ell < r < y_{j+1}$, $r - \ell < \delta$, $z = \frac{\ell+r}{2} - \epsilon$ and $0 < \epsilon < \frac{r-\ell}{2}$.

In the profile $\mathcal{L}_0$, let a and b denote the agents who report $[\ell, r]$ and z, respectively, and let $p_0 = \mathcal{M}(\mathcal{L}_0)$. First, note that if $\mathcal{L}_1$ denotes the profile where agent a reports ℓ instead of $[\ell, r]$ and every other agent reports as in $\mathcal{L}_0$, then

$$p_1 = \mathcal{M}(\mathcal{L}_1) = \text{median}(y_1, \ldots, y_{n-1}, y_j, \ldots, y_j, \ell, z, B, \ldots B) = \ell.$$

Also, if $\mathcal{L}_2$ denotes the profile where agent a reports r instead of $[\ell, r]$ and every other agent reports as in $\mathcal{L}_0$, then

$$p_2 = \mathcal{M}(\mathcal{L}_2) = \text{median}(y_1, \ldots, y_{n-1}, y_j, \ldots, y_j, r, z, B, \ldots B) = z.$$

Now, since $p_1 = \ell$ and $p_2 = z$, this implies that

$$p_0 = \mathcal{M}(\mathcal{L}_0) = \frac{\ell + z}{2},$$

for, if otherwise, agent a can deviate from $\mathcal{L}_0$ by reporting $\frac{\ell+z}{2}$ instead (it is easy to see that this reduces agent a's maximum regret), thus violating the fact that $\mathcal{M}$ is minimax dominant.

Given this, consider the profile $\mathcal{L}_3$ which is the same as $\mathcal{L}_0$ except for the fact that agent b reports $(2z - \ell)$ instead of z. By again using a similar line of reasoning as in the case for $\mathcal{L}_0$, one can see that

$$p_3 = \mathcal{M}(\mathcal{L}_3) = z.$$

However, this in turn implies that agent b can deviate from $\mathcal{L}_0$ to $\mathcal{L}_3$, thus again violating the fact that $\mathcal{M}$ is minimax dominant. $\square$

Given the fact that we cannot have an anonymous, minimax dominant, and onto mechanism, the natural question to consider is if we can find non-onto mechanisms that perform well. We answer this question in the next section.

Input: a $\delta \geq 0$ and for each agent i, their input interval $[a_i, b_i]$
Output: location of the facility p

1: $A \leftarrow \{g_1, g_2, \ldots, g_k\}$, where $g_1 = 0, g_k \leq B$, and $g_{i+1} - g_i = \frac{\delta}{2}$
2: **for** each $i \in \{1, \ldots, n\}$ **do**
3:     $x_i \leftarrow$ point closest to a_i in A (in case of a tie, break in favour of the point in $[a_i, b_i]$ if there exists one, break in favour of point to the left otherwise)
4:     $y_i \leftarrow$ point closest to b_i in A (break ties as in line 3)
5:     **if** $|[x_i, y_i] \cap A| == 1$ **then** $\triangleright$ the case when $x_i = y_i$
6:         $\ell_i \leftarrow x$
7:     **else if** $|[x_i, y_i] \cap A| == 2$ **then**
8:         **if** $|[x_i, y_i] \cap [a_i, b_i]| < 2$ **then**
9:             **if** $a_i + b_i \leq x_i + y_i$ **then**
10:                 $\ell_i \leftarrow x_i$
11:             **else**
12:                 $\ell_i \leftarrow y_i$
13:             **end if**
14:         **else**
15:             $\ell_i \leftarrow x_i$
16:         **end if**
17:     **else if** $|[x_i, y_i] \cap A| == 3$ **then**
18:         $\ell_i \leftarrow z_i$, where z_i is the point in $[x_i, y_i] \cap A$ that is neither x_i nor y_i
19:     **end if**
20: **end for**
21: return median$(\ell_1, \ldots, \ell_n)$

Algorithm 2: $\frac{\delta}{2}$-equispaced-median mechanism

3.4.3.1 Non-onto mechanisms

We first show an upper bound, presenting an anonymous mechanism that implements $\frac{3\delta}{4}$-OMV$_{AC}$ in minimax dominant strategies. Following this, we present a conditional lower bound that shows that one cannot achieve an approximation bound better than $\frac{\delta}{2}$ when considering mechanisms that have a finite range.

An anonymous and minimax dominant mechanism. Consider the $\frac{\delta}{2}$-equispaced-median mechanism defined in Algorithm 2, which can be thought of as an extension to the standard median mechanism. The key assumption in this mechanism is that the designer knows a δ such that any agent's candidate locations has a length at most δ. Given this δ, the main idea is to divide the interval $[0, B]$ into a set of "grid points" and then map every profile of reports to one of these points, while at the same time ensuring that the mapping is minimax dominant. In particular, in the case of the $\frac{\delta}{2}$-equispaced-median mechanism, when $\delta > 0$, its range is restricted to the finite set of points $A = \{g_1, g_2, \ldots g_m\}$ such that, for $i \geq 1$, $g_{i+1} - g_i = \frac{\delta}{2}$, $g_1 = 0$, and $g_m \leq B$.

Below we first prove a lemma where we show that the $\frac{\delta}{2}$-equispaced-median mechanism is minimax dominant. Subsequently, we then use this to prove our main theorem which shows that the $\frac{\delta}{2}$-equispaced-median mechanism implements $\frac{3\delta}{4}$-OMV$_{AC}$ in minimax dominant strategies.

Lemma 20. *Given a $\delta \in [0, B]$ and for every agent i in a δ-uncertain-facility-location-game, reporting the candidate locations $[a_i, b_i]$ is a minimax dominant strategy for agent i in the $\frac{\delta}{2}$-equispaced-median mechanism.*

Proof. Let us first fix an agent i and let $[a_i, b_i]$ be their candidate locations. Also, let $\mathcal{L}_0$ be some arbitrary profile of candidate locations, where $\mathcal{L}_0 = ([a_1, b_1], \ldots, [a_n, b_n])$. We need to show that it is minimax dominant for agent i to report $[a_i, b_i]$ in the $\frac{\delta}{2}$-equispaced-median mechanism (denoted by $\mathcal{M}$ from now on), and for this we broadly consider the following two cases. Intuitively, in both cases what we try to argue is that for an agent i with candidate locations $[a_i, b_i]$ the ℓ_i that is associated with i in the mechanism is in fact the agent's "best alternative" among the alternatives in A (see line 1 in Algorithm 2).

Case 1: $a_i = b_i$**.** In this case, we show that it is a very weakly dominant strategy for agent i to report a_i. To see this, let p be the output of $\mathcal{M}$ when agent i reports a_i and x_i be the point that is closest to a_i in A (with ties broken in favour of the point which is to the left of a_i). From line 6 in the mechanism, we know that $\ell_i = x_i$. Now, if either $\ell_i < p$ or $\ell_i > p$, then any report that changes the median will only result in the output being further away from agent i. And if $\ell_i = p$, then since we choose ℓ_i to be the point that is closest to a_i in A, it is clear that it is very weakly dominant for the agent to report a_i. Hence, from both the cases above, we have our claim.

Case 2: $a_i \neq b_i$**.** Let p be the output of $\mathcal{M}$ when agent i reports $[a_i, b_i]$, and let x_i and y_i be the points that are closest (with ties being broken in favour of points in $[a_i, b_i]$ in both cases) to a_i and b_i, respectively. From the mechanism we can see that $\ell_i \in [x_i, y_i]$. Next, let us first consider the case when $p < x_i$ or $p > y_i$. In both these cases, given the fact that x_i and y_i are the points closest to a_i and b_i, respectively, p has to be outside $[a_i, b_i]$. And so, if this is the case, then if agent i misreports and the output changes to some $p' < p$ or $p' > p$, in both cases, it is easy to see that the maximum regret associated with p' is greater than the one associated with p. Hence, the only case where an agent i can successfully misreport is if $p \in [x_i, y_i]$. So, we focus on this scenario below.

Considering the scenario when $p \in [x_i, y_i]$, first, note that the interval $[x_i, y_i]$ can have at most three points that are also in A (this is proved in Claim 18, which is in Appendix B.3). So, given this, let us now consider the following cases.

i) $|[x_i, y_i] \cap A| = 1$. Since $p \in [x_i, y_i]$, this implies that $p = x_i = \ell_i$. Therefore, in this case, if agent i misreports, then she only experiences a greater maximum regret as the resulting output p' would be outside $[x_i, y_i]$ and we know from our discussion above that these points have a greater maximum regret than a point in $[x_i, y_i]$.

ii) $|[x_i, y_i] \cap A| = 2$. First, note that since $p \in [x_i, y_i]$ and the only other point in $[x_i, y_i]$ that is also in A is y_i, if $p < \ell_i$ or $p > \ell_i$, then agent i can only increase her maximum regret by misreporting and changing the outcome (because the new outcome will be outside $[x_i, y_i]$). Therefore, we only need to consider the case when $p = \ell_i$, and here we consider the following sub-cases.

   (a) $p = \ell_i = x_i$. From the mechanism we know that this happens only when either both x_i and y_i are in $[a_i, b_i]$ (lines 14–15) or when $a_i + b_i \leq x_i + y_i$ (lines 9–10). Now, since we know that every point outside $[x_i, y_i]$ is worse in terms of maximum regret than x_i or y_i, we only need to consider the case when agent i misreports in such a way that it results in the new outcome p' being equal to y_i. And below we show that under both the conditions stated above (lines 9–10 and 14–15, both of which result in ℓ_i being defined as being equal to x_i in the mechanism) the maximum regret associated with y_i is at least as much as the one associated with x_i.

   To see this, consider the profile where agent i reports a_i instead of $[a_i, b_i]$ and all the other agents' reports are the same as in $\mathcal{L}_0$. Let p_a be the output of $\mathcal{M}$ for this profile. Note that $p_a = x_i$ since the ℓ_i associated with agent i in this profile is x_i and so the outcome in this profile is the same as $p = x_i$. Similarly, let p_b be the outcome when agent i reports b_i instead of $[a_i, b_i]$. Since the ℓ_i associated with agent i in this profile is y_i, one can see that $x_i \leq p_b \leq y_i$. Now, if $p_b = x_i$, then $\text{maxRegret}_i(x_i) = 0$, where $\text{maxRegret}_i(x_i)$ is the maximum regret associated with the point x_i for agent i (see (3.1) for the definition), and so x_i is definitely better than y_i. Therefore, we can ignore this case and instead assume that $p_b = y_i$. So, considering this, since for the maximum regret calculations only the endpoints a_i and b_i matter (this is proved in Claim 19, which is in Appendix B.3), we have that

$$\text{maxRegret}_i(y_i) = \max\{|a_i - y_i| - |a_i - p_a|, |b_i - y_i| - |b_i - p_b|\}$$

$$= |a_i - y_i| - |a_i - p_a| \qquad \big(\text{since } p_b = y_i \text{ and } p_a = x_i\big)$$
$$\geq |b_i - x_i| - |b_i - p_b|$$
$$\big(\text{depending on the case, use either the fact that } x_i, y_i \in [a_i, b_i] \text{ or that } a_i + b_i \leq x_i + y_i\big)$$
$$= \text{maxRegret}_i(x_i). \qquad \big(\text{since } p_b = y_i \text{ and } p_a = x_i\big)$$

Hence, we see that in this case agent i cannot gain by misreporting.

(b) $p = \ell_i = y_i$. We can show that in this case maxRegret$(x_i) \geq$ maxRegret(y_i) by proceeding similarly as in the case above.

iii) $|[x_i, y_i] \cap A| = 3$. Let $x_i < z_i < y_i$ be the three points in $[x_i, y_i] \cap A$. Since the length of $[a_i, b_i]$ is at most δ, note that both x_i and y_i cannot be in the interval (a_i, b_i). So below we assume without loss of generality that $x_i \leq a_i$. From the mechanism we know that in this case $\ell_i = z_i$ (line 18). Also, as in the cases above, note that if $p < \ell_i$ or $p > \ell_i$, then agent i can only increase her maximum regret by misreporting and changing the outcome (because, again, the new outcome will be outside $[x_i, y_i]$). Therefore, the only case we need to consider is if $p = \ell_i$, and for this case we show that both x_i and y_i have a maximum regret that is at least as much as that associated with z_i (we do not need to consider points outside $[x_i, y_i]$ since we know that these points have a worse maximum regret than any of the points in $[x_i, y_i]$). Note that if this is true, then we are done as this shows that agent i cannot benefit from misreporting.

To see why the claim is true, consider the profile where agent i reports a_i instead of $[a_i, b_i]$ and all the other agents' reports are the same as in $\mathcal{L}_0$. Let p_a be the outcome of $\mathcal{M}$ for this profile. Similarly, let p_b be the outcome when agent i reports b_i instead of $[a_i, b_i]$. Note that $p_a \leq p = z_i$ and $p_b \geq p = z_i$. Now, since, again, for the maximum regret calculations only the endpoints a_i and b_i matter (this is proved in Claim 19, which is in Appendix B.3), we have that

$$\text{maxRegret}_i(z_i) = \max\{|a_i - z_i| - |a_i - p_a|, |b_i - z_i| - |b_i - p_b|\}$$
$$\leq |b_i - x_i| - |b_i - p_b|$$
$$\big(\text{since } x_i \text{ is the closest point to } a_i \text{ in } A \text{ and using very weak dominance for single reports}\big)$$
$$= \max\{|a_i - x_i| - |a_i - p_a|, |b_i - x_i| - |b_i - p_b|\}$$
$$\big(\text{since } x_i \leq a_i \text{ and } |b_i - p_b| \leq |b_i - p_a| \text{ by very weak dominance for single reports}\big)$$
$$= \text{maxRegret}_i(x_i).$$

Similarly, we can show that maxRegret$(z_i) \leq$ maxRegret(y_i). Hence, agent i will not derive any benefit from misreporting her candidate locations.

Finally, combining all the cases above, we have that the $\frac{\delta}{2}$-equispaced-median mechanism is minimax dominant. This concludes the proof of our lemma. ☐

Equipped with the lemma above, we can now prove the following theorem.

Theorem 21. *Given a $\delta \in [0, B]$, the $\frac{\delta}{2}$-equispaced-median mechanism is anonymous and implements $\frac{3\delta}{4}$-OMV$_{AC}$ in minimax dominant strategies for a δ-uncertain-facility-location-game.*

Proof. From Lemma 20 we know that the mechanism is minimax dominant. Therefore, the only thing left to show is that it achieves an approximation bound of $\frac{3\delta}{4}$. To do this, consider an arbitrary profile of candidate locations $\mathcal{L}_0$, where $\mathcal{L}_0 = ([a_1, b_1], \ldots, [a_n, b_n])$, and let $L_1, \ldots, L_n$ and $R_1, \ldots, R_n$ denote the sorted order of the left endpoints (i.e., $\{a_i\}_{i\in[n]}$) and right endpoints (i.e., $\{b_i\}_{i\in[n]}$), respectively. Next, consider the interval $[L_{k+1}, R_{k+1}]$ that is associated with the given profile $\mathcal{L}_0$, where $k = \frac{\lfloor n \rfloor}{2}$ and L_{k+1} and R_{k+1} are the medians of the sets $\{a_i\}_{i\in[n]}$ and $\{b_i\}_{i\in[n]}$, respectively. (Note that $R_{k+1} - L_{k+1} \leq \delta$ from Claim 15, which is in Appendix B.3.) Also, let i be an agent who reported an a_i such that $a_i = L_{k+1}$ and j denote one who reported a b_j such that $b_j = R_{k+1}$. Next, from the mechanism, consider x_i (see line 3) and y_j (see line 4).

First, for every agent c with $b_c \leq R_{k+1}$ (and there are $k+1$ of them), $\ell_c \leq y_j$. This is so because, for every agent c, the point ℓ_c that is associated with the agent is in $[x_c, y_c]$ and also for two agents c, c', if $b_{c'} \leq b_c$, then $y_{c'} \leq y_c$. Similarly, for every agent d with $a_d \geq L_{k+1}$ (and, again, there are $k+1$ of them), $\ell_d \geq x_i$. Hence, it follows from the two observations above that the output p of the mechanism, which is the median of $\{\ell_1, \ldots, \ell_n\}$, will be in the interval $[x_i, y_j]$.

Second, we will show that if $|[x_i, y_j] \cap A| = 3$, then the output p of the mechanism will be the point z in interval (x_i, y_j) such that $z \in A$. To see this, consider the points $R_1, \ldots, R_{k+1}$ and consider the largest q, where $q \leq k$, such that $R_q \leq z$ (define $q = 0$, if $R_1 > z$). We will assume without of loss of generality that, for each i, the agent associated with R_i (i.e., one who reports a right endpoint such that it is equal to R_i) is agent i. Now, for each agent r from 1 to q, the ℓ_r associated with them in the mechanism is at most z. This is so because, for each such agent r the y_r associated with them at most z. Also, for each agent s from $q+1$ to $k+1$, the only way the ℓ_s associated with agent s is greater than z is if the left endpoints associated with them are greater than L_{k+1} (because if not, then one can see that the $[x_s, y_s]$ associated with agent s has x_i and z in it and so ℓ_s cannot be greater than z). Now, let n_1 be the number of agents among the agents from $q+1$ to $k+1$ such that their left endpoints are greater than L_{k+1}. This implies that there are n_1 agents among the ones that report a left endpoint in $\{L_1, \ldots, L_{k+1}\}$ that have a corresponding right endpoint greater than or equal to R_{k+1}. And this in turn implies that the ℓs associated with these n_1 agents can be at most z (because, again, one can see that the $[x, y]$ associated with such an agent will have x_i and z in it). Hence, combining all the observations above, we see that $q + ((k + 1 - (q + 1) + 1) - n_1) + n_1 = k + 1$ agents report an ℓ that is at most z. Now, we can employ a similar line of reasoning to show that $k + 1$ agents report an ℓ that is at least z. Hence, it follows that the median of $\{\ell_1, \dots, \ell_n\}$ must be z.

Given the observations above, let us now look at the following cases. In all the cases we show that any point in $[L_{k+1}, R_{k+1}]$ is at a distance of at most $\frac{3\delta}{4}$ from p, the output of the mechanism.

Case 1: both x_i and y_j are not in $[L_{k+1}, R_{k+1}]$. In this case, let us consider the following sub cases.

a) $|[x_i, y_j] \cap A| = 2$. Since both x_i and y_j are not in $[L_{k+1}, R_{k+1}]$ and there is no other point in A that is in $[L_{k+1}, R_{k+1}]$, we have that the distance from any point in $[L_{k+1}, R_{k+1}]$ to either x_i or y_j is at most $\frac{\delta}{2}$ (as the distance between x_i and y_j is $\frac{\delta}{2}$). Also, from the discussion above we know that the median that is returned by the mechanism is either x_i or y_j, so we have our bound.

b) $|[x_i, y_j] \cap A| = 3$. In this case we know from above that the output of the mechanism $p = z$, where $z \in [x_i, y_j] \cap A$ and is neither x_i nor y_j. Therefore, we have that $(z - L_{k+1}) \leq (z - x_i) = \frac{\delta}{2}$, and $(R_{k+1} - z) < (y_j - z) = \frac{\delta}{2}$. Hence, any point in $[L_{k+1}, R_{k+1}]$ is a distance of at most $\frac{\delta}{2}$ from p.

c) $|[x_i, y_j] \cap A| > 3$. Since for any interval $[a, b]$ of length at most δ, the $[x, y]$ associated with it can have at most three points in A (this is proved in Claim 18, which is in Appendix B.3), where x and y are as defined in the mechanism (see lines 3–4), this case is impossible.

Case 2: at least one of x_i or y_j is in $[L_{k+1}, R_{k+1}]$. Let us assume without loss of generality that x_i is the point in $[L_{k+1}, R_{k+1}]$, and let us consider the following sub cases.

a) $|[x_i, y_j] \cap A| = 2$. In this case, if the output of the mechanism $p = x_i$, then we have that $(x_i - L_{k+1}) \leq \frac{\delta}{4}$ (since, by definition, x_i is the point in A that is closest to L_{k+1}), and $(R_{k+1} - x_i) \leq |R_{k+1} - y_j| + (y_j - x_i) \leq \frac{\delta}{4} + \frac{\delta}{2}$ (since, by definition, y_j is the point in A that is closest to R_{k+1}). Hence, any point in $[L_{k+1}, R_{k+1}]$ is a distance of at most $\frac{3\delta}{4}$ from p. On other hand, if the output of the mechanism $p = y_j$, then we have that $(y_j - L_{k+1}) = (y_j - x_i + x_i - L_{k+1}) \leq \frac{\delta}{2} + \frac{\delta}{4}$ (since x_i is the point in A that is closest to L_{k+1}), and $|y_j - R_{k+1}| \leq \frac{\delta}{4}$ (since y_j is the point in A that is closest to R_{k+1}). Hence, when $p = y_j$, any point in $[L_{k+1}, R_{k+1}]$ is a distance of at most $\frac{3\delta}{4}$ from p. Combining the two, we see that our claim is true in this case.

b) $|[x_i, y_j] \cap A| = 3$. Here again since $p = z$, where $z \in [x_i, y_j] \cap A$ and is neither x_i nor y_j, we have that $(z - L_{k+1}) \leq (z - x_i + x_i - L_{k+1}) \leq \frac{\delta}{2} + \frac{\delta}{4}$ (since x_i is the point in A that is closest to L_{k+1}), and $(R_{k+1} - z) \leq (y_j - z) \leq \frac{\delta}{2}$. Hence, even in this case, any point in $[L_{k+1}, R_{k+1}]$ is at a distance of at most $\frac{3\delta}{4}$ from p.

Finally, since the output of the mechanism is a distance of at most $\frac{3\delta}{4}$ from any point in $[L_{k+1}, R_{k+1}]$, and given the fact that the minimax optimal solution is in $[L_{k+1}, R_{k+1}]$ (from Lemma 17), one can see using Lemma 16 that for $\mathcal{I} = [a_1, b_1] \times \ldots \times [a_n, b_n]$, $\text{maxRegret}(p, \mathcal{I}) - \text{OMV}_{AC}(\mathcal{I})$ is also bounded by $\frac{3\delta}{4}$ (this is proved in Claim 20, which is in Appendix B.3). $\square$

A conditional lower bound. In the context of our motivating example from the introduction, it is possible, and in fact quite likely, that the city can only build the school at a finite set of locations on the street. Therefore, an interesting class of non-onto mechanisms to consider is ones which have a finite range. Below, we consider such mechanisms and show that the approximation bound associated with any mechanism that is anonymous, minimax dominant, and has a finite range, is at least $\frac{\delta}{2}$. The key idea that is required in order to show this bound is the following lemma, which informally says that if the mechanism has a finite range, is minimax dominant, and achieves a bound less than $\frac{3\delta}{4}$, then there is "sufficient-gap" between four consecutive points in the range, A, of the mechanism. Once we have this observation it is then in turn used to construct profiles that will result in the stated bound.

Lemma 22. *Given a $\delta \in (0, \frac{B}{6}]$, let $\mathcal{M}$ be a deterministic mechanism that has a finite range A (of size at least six), is anonymous, and one that implements α-OMV$_{AC}$ in minimax dominant strategies for a δ-uncertain-facility-location-game. Then, either $\alpha \geq \frac{3\delta}{4}$, or there exists four consecutive points $g_1, g_2, g_3, g_4 \in A$ such that $g_1 < g_2 < g_3 < g_4$ and $\frac{d_1}{2} + d_2 + \frac{d_3}{2} \geq \delta$, where, for $i \in [3], d_i = g_{i+1} - g_i$.*

Proof. The proof here is broadly similar to the proof of Theorem 19 in that here again we use a characterization result by Massó and De Barreda [MD11], albeit a different one, and then construct profiles in order to prove that the claim is true. The result we will rely on is the one that characterizes the set of anonymous and strategyproof mechanisms in the symmetric single-peaked domain [MD11, Corollary 1]. In particular, Massó and De Barreda showed that any mechanism that is anonymous and strategyproof in the symmetric single-peaked domain can be described as a *disturbed median.* We will not be defining disturbed medians precisely since the definition is quite involved and we do not need it for the results here, but broadly a mechanism $\mathcal{M}'$ is a disturbed median if

i) it has $n+1$ fixed points $0 \leq y_1 \leq \ldots \leq y_{n+1} \leq 1$,

ii) its range has a countable number of non-intersecting discontinuity intervals $\{[a_m, b_m]\}_{m \in M}$, where M is some indexation set and where for all i, $y_i \notin [a_m, b_m]$ for all $m \in M$,

iii) has a family of anonymous tie-breaking rules, and

iv) behaves the following way with respect to a profile $(x_1, \ldots, x_n)$ of exact reports: if $\text{median}(y_1, \ldots, y_{n+1}, x_1, \ldots, x_n) \neq d_m$ for all $m \in M$, where $d_m = \frac{a_m + b_m}{2}$, then $\mathcal{M}'(x_1, \ldots, x_n) = \text{median}(y_1, \ldots, y_{n+1}, t_1, \ldots, t_n)$, where t_i is the most preferred alternative of agent i in the range of $\mathcal{M}'$.[9]

Given the partial description of disturbed medians mentioned above, let us consider the mechanism $\mathcal{M}$ that is anonymous and minimax dominant. First, note that if we restrict ourselves to profiles where every agent's report is a single point (instead of intervals as in our setting), then we know from the result of Massó and De Barreda that $\mathcal{M}$ must be a disturbed median [MD11, Corollary 1]. Second, note that for each y_i, $\mathcal{M}(y_i, \ldots, y_i) = \text{median}(y_1, \ldots, y_{n+1}, y_i, \ldots, y_i) = y_i$. Therefore, each y_i belongs to A, the range of $\mathcal{M}$. Additionally, also note that y_1 and y_{n+1} are the minimum and maximum elements, respectively, in A, for if not then for a $g \in A$ that is either less than y_1 or greater than y_{n+1}, $\text{median}(y_1, \ldots, y_{n+1}, g, \ldots, g) \neq g$, which is impossible if $g \in A$ and $\mathcal{M}$ is minimax dominant.

Next, given the observations above, consider the following two cases.

Case 1: there exists at most 2 points in A that are greater than y_2. In this case, let us consider the points y_1 and y_2. If $y_2 - y_1 \geq \frac{3\delta}{2}$, then consider the profile $\mathcal{L}_0$ where $n-1$ agents report y_1 and the last agent reports y_2, i.e., $\mathcal{L}_0 = (y_1, \ldots, y_1, y_2)$. Let $p_0 = \mathcal{M}(\mathcal{L}_0)$. From above we know that $p_0 = \text{median}(y_1, \ldots, y_{n+1}, y_1, \ldots, y_1, y_2) = y_2$ (since y_2 is the $(n+1)$-th largest number). However, the minimax optimal solution associated with this profile is y_1 and now one can verify from the expressions in Lemma 16 that, for an appropriate choice of n, $\alpha \geq \text{maxRegret}(p_0, \mathcal{I}) - \text{OMV}_{AC}(\mathcal{I}) \geq \frac{3\delta}{4}$, where $\mathcal{I}$ is the instance associated with the profile $\mathcal{L}_0$. Hence, "the either part" of our lemma is true in this case.

On the other hand, if $y_2 - y_1 < \frac{3\delta}{2}$, then consider the case when there are two points $g, g' \in A$ such that $y_2 < g' < g$ and the three points are consecutive in A (the arguments below can be easily modified when there is only one or zero points greater than y_2). Note

[9] Note that we have only defined disturbed medians for profiles which satisfy the property mentioned above since we will only be using such profiles in this proof; the interested reader can refer to [MD11, Definition 7] for a complete definition of disturbed medians.

that if $g' - y_2 \geq \frac{3\delta}{2}$ or $g - g' \geq \frac{3\delta}{2}$, then we can construct profiles like in the case above to show that $\alpha \geq \frac{3\delta}{4}$. Therefore, let us assume that this is not the case. Now, this implies that if $d_1 = (y_1 - 0)$ and $d_2 = (B - g)$, then

$$B = B - g + g - y_2 + y_2 - y_1 + y_1 < d_2 + \frac{3\delta}{2} + \frac{3\delta}{2} + \frac{3\delta}{2} + d_1.$$

However, this in turn implies that at least one of d_1 or d_2 is at least $\frac{B}{2} - \frac{9\delta}{4} \geq \frac{3\delta}{4}$. So, if we assume without loss of generality that $d_1 \geq \frac{3\delta}{4}$, then we can consider the profile $\mathcal{L}_0$ where all the agents report 0 (note that by our assumption on d_1, $0 \notin A$, because if so, then since y_1 is the minimum element in A, y_1 should be equal to 0), and let $p_1 = \mathcal{M}(\mathcal{L}_1)$. Given the fact that $p_1 \geq y_1$, this in turn implies that one can verify from the expressions in Lemma 16 that $\alpha \geq \text{maxRegret}(p_1, \mathcal{I}) - \text{OMV}_{AC}(\mathcal{I}) \geq \frac{3\delta}{4}$, where $\mathcal{I}$ is the instance associated with the profile $\mathcal{L}_1$. Hence, again, "the either part" of our lemma is true.

Case 2: there are at least 3 points in A that are greater than y_2. In this case, consider four consecutive points g_1, g_2, g_3, g_4 in A such that $g_1 \geq y_2$ and $g_1 < g_2 < g_3 < g_4$. Also, for $i \in [3]$, let d_i denote the distance between the points g_i and g_{i+1}. Below we will show that $\frac{d_1}{2} + d_2 + \frac{d_3}{2} \geq \delta$.

Now, to prove our claim, let us assume for the sake of contradiction that $\frac{d_1}{2} + d_2 + \frac{d_3}{2} < \delta$. Also, let us consider the largest j such that $y_j \leq g_1$ (note that we have $2 \leq j \leq n$), and let us consider the following input profile $\mathcal{L}_0$

$$\left(\underbrace{y_j, \ldots, y_j}_{n-j \text{ agents}}, [\ell, r], g_3, \underbrace{B, \ldots, B}_{j-2 \text{ agents}} \right),$$

where $\ell = (g_1 + \frac{d_1}{2} - \gamma_1)$, $r = (g_3 + \frac{d_3}{2} + \gamma_2)$, $\gamma_1 + \gamma_2 < (\delta - (\frac{d_1}{2} + d_2 + \frac{d_3}{2}))$, $\gamma_1 < \gamma_2$, $\gamma_1 < \frac{d_1}{2}$, and $\gamma_2 < \frac{d_3}{2}$.

Let $p_0 = \mathcal{M}(\mathcal{L}_0)$ and let a and b be the agents who report $[\ell, r]$ and g_3, respectively, in $\mathcal{L}_0$. Next, consider the profile $\mathcal{L}_1$ where the only difference from $\mathcal{L}_0$ is that agent a reports ℓ here instead of $[\ell, r]$, and let $p_1 = \mathcal{M}(\mathcal{L}_1)$. Now, given the fact that g_1, g_2, g_3, g_4 are consecutive points in A and g_1 is the unique closest point to ℓ in A, we know that g_1 is the most preferred alternative of agent a in the range of $\mathcal{M}$. Therefore, we have that

$$p_1 = \text{median}(y_1, \ldots, y_{n+1}, y_j, \ldots, y_j, g_1, g_3, B, \ldots B) = g_1,$$

since g_1 is the $(n+1)$-th largest number.

Using a similar line of reasoning one can see that if $\mathcal{L}_2$ denotes the profile where agent a reports r instead of $[\ell, r]$ and every other agent reports as in $\mathcal{L}_0$, then

$$p_2 = \mathcal{M}(\mathcal{L}_2) = \text{median}(y_1, \ldots, y_{n+1}, y_j, \ldots, y_j, g_4, g_3, B, \ldots B) = g_3,$$

since here g_3 is the $(n+1)$-th largest number.

Now, since $p_1 = g_1$ and $p_2 = g_3$, this implies that $p_0 = g_2$, for if otherwise agent a can deviate from $\mathcal{L}_0$ by reporting g_2 instead (it is easy to see that this reduces agent a's maximum regret), thus violating the fact that $\mathcal{M}$ is minimax dominant. Given this, now consider the profile $\mathcal{L}_3$ which is the same as $\mathcal{L}_0$ except for the fact that agent b reports g_4 instead of g_3. Here $p_3 = \mathcal{M}(\mathcal{L}_3)$ has to be equal to g_3 because one can see from the max. regret calculations associated with agent a that a has a lesser maximum regret for g_3 than for g_2 (this is because of our choice of appropriate γ_1 and γ_2). So if p_3 is not equal to g_3, then agent a can move from $\mathcal{L}_3$ to $\mathcal{L}_4$ where $\mathcal{L}_4$ is the same as $\mathcal{L}_3$ except for agent a reporting g_3 instead of $[\ell, r]$ (and this would be beneficial for a since the output for $\mathcal{L}_4 = \text{median}(y_1, \ldots, y_{n+1}, y_j, \ldots, y_j, g_3, g_4, B, \ldots B) = g_3$). However, now if $p_3 = g_3$, then this implies that agent b can deviate from $\mathcal{L}_0$ (which, as discussed above, has an output $p_0 = g_2$) to $\mathcal{L}_3$, thus again violating the fact that $\mathcal{M}$ is minimax dominant. Hence, for this case, we have that $\frac{d_1}{2} + d_2 + \frac{d_3}{2} \geq \delta$.

Finally, combining all the cases above, we have our lemma. $\square$

Given the lemma above, the proof of our lower bound is straightforward. (Note that below we ignore mechanisms which have less than six points in their range as one can easily show that such mechanisms perform poorly.)

Theorem 23. *Given a $\delta \in (0, \frac{B}{6}]$, let $\mathcal{M}$ be a deterministic mechanism that has a finite range (of size at least six), is anonymous, and one that implements α-OMV_{AC} in minimax dominant strategies for a δ-uncertain-facility-location-game. Then, for any $\epsilon > 0$, $\alpha \geq \frac{\delta}{2} - \epsilon$.*

Proof. Consider the mechanism $\mathcal{M}$ and let A denote its range. From Lemma 22 we know that either $\alpha \geq \frac{3\delta}{4}$ or there exists four consecutive points $g_1, g_2, g_3, g_4 \in A$ such that $g_1 < g_2 < g_3 < g_4$ and $\frac{d_1}{2} + d_2 + \frac{d_3}{2} \geq \delta$, where, for $i \in [3], d_i = g_{i+1} - g_i$. Since the former case results in a bound that is bigger than the one in the statement of our theorem, below we just consider the latter case where there exists g_1, g_2, g_3, g_4 satisfying the conditions stated above. Also, since $\mathcal{M}$ is minimax dominant and anonymous, we can again make use of Massó and De Barreda's characterization result [MD11, Corollary 1] as we did in Lemma 22.

Now, since $\frac{d_1}{2} + d_2 + \frac{d_3}{2} \geq \delta$, we know that at least one of d_1, d_2 or d_3 is at least $\frac{\delta}{2}$. Also if any of d_1, d_2, or d_3 is at least δ, then it is easy to construct profiles so as to achieve our bounds. Therefore, for the rest of the proof we assume that $d_c < \delta$, for all $c \in [3]$. So, given this, let $d_i \geq \frac{\delta}{2}$ for some $i \in [3]$, and let us consider the largest j such that $y_j \leq g_i$ (again, like in the proof of Lemma 22, $j \leq n$). We have the following two cases, where $k = \lfloor \frac{n}{2} \rfloor$.

Case 1: $j \geq n - k$**.** Consider the profile $\mathcal{L}_0$ where $k+1$ agents report g_i and the rest of the agents report g_{i+1}. Let $p_0 = \mathcal{M}(\mathcal{L}_0)$. Since $p_0 = \text{median}(y_1, \ldots, y_{n+1}, g_i, \ldots, g_i, g_{i+1}, \ldots g_{i+1})$, and since $j \geq n-k$, we have that $p_0 = g_i$. Next, consider the profile $\mathcal{L}_1$, where the only change from $\mathcal{L}_0$ is that here agent 1 reports $[g_i, g_{i+1} - \gamma]$, where $0 < \gamma < \min(\frac{d_i}{2}, \frac{\delta}{2})$, instead of g_i. Let $p_1 = \mathcal{M}(\mathcal{L}_1)$. Now, if either $p_1 > g_i$ or $p_1 \leq g_{i-1}$ (if such a point exists in A), it is easy to see that agent 1 will deviate to $\mathcal{L}_0$ since her maximum regret for the point g_i is lesser in either of the cases. This in turn implies that $p_1 = g_i$. Continuing this way, one can reason along the same lines that for $c \leq k+1$ the profile $\mathcal{L}_c$, where $\mathcal{L}_c$ is the same as $\mathcal{L}_{c-1}$ except for agent c reporting $[g_i, g_{i+1} - \gamma]$ instead of g_i, the output p_c associated with $\mathcal{L}_c$ is equal to g_i.

Given the observations above, consider the profile $\mathcal{L}_{k+1}$ where the first $k+1$ agents report $[g_i, g_{i+1} - \gamma]$ and the rest of the agents report g_{i+1}. Now, one can calculate using Algorithm 9 (in Appendix B.2) that the minimax optimal solution $p_{opt} \geq \frac{g_i + (2k+1)(g_{i+1} - \gamma)}{2(k+1)}$. Also, we know from above that $p_{k+1} = \mathcal{M}(\mathcal{L}_{k+1}) = g_i$. And so, using the fact that $d_i = g_{i+1} - g_i \geq \frac{\delta}{2}$ and that $p_{opt} - p_{k+1} \geq \frac{(2k+1)(g_{i+1} - g_i - \gamma)}{2(k+1)}$ we can now use the expressions from Lemma 16 to see that $\text{maxRegret}(p_{k+1}, \mathcal{I}) - \text{OMV}_{AC}(\mathcal{I}) \geq \frac{(2k+1)(g_{i+1} - g_i - \gamma)}{2(k+1)}$, which for an appropriately chosen value of n and γ is greater than or equal to $\frac{\delta}{2} - \epsilon$ for any $\epsilon > 0$.

Case 2: $j < n - k$**.** We can handle this similarly as in the previous case. In particular, consider the profile $\mathcal{L}_0$ where $k+1$ agents report g_i and the rest of the agents report g_{i+1}. Let $p_0 = \mathcal{M}(\mathcal{L}_0)$. Since $p_0 = \text{median}(y_1, \ldots, y_{n+1}, g_i, \ldots, g_i, g_{i+1}, \ldots g_{i+1})$, and since $j < n - k$, we know that $p_0 = \min(g_{i+1}, y_{j+1})$ (as the $(n+1)$-th largest number will be either y_{j+1} or g_{i+1}). However, we know that y_{j+1} is in the range of A since $\mathcal{M}(y_{j+1}, \ldots, y_{j+1}) = \text{median}(y_1, \ldots, y_{n+1}, y_{j+1}, \ldots, y_{j+1}) = y_{j+1}$, and also that g_i and g_{i+1} are consecutive in A. Therefore, $y_{j+1} \geq g_{i+1}$, and we have that $p_0 = g_{i+1}$. Next, consider the profile $\mathcal{L}_1$, where the only change from $\mathcal{L}_0$ is that here agent $(k+2)$ reports $[g_i + \gamma, g_{i+1}]$, where $0 < \gamma < \min(\frac{d_i}{2}, \frac{\delta}{2})$, instead of g_{i+1}. Let $p_1 = \mathcal{M}(\mathcal{L}_1)$. Now, if either $p_1 > g_{i+1}$ (if such a point exists in A) or $p_1 \leq g_i$, it is easy to see that agent $(k+2)$ will deviate to $\mathcal{L}_0$ since her maximum regret for the point g_{i+1} is lesser in either of the cases. This in turn implies that $p_1 = g_{i+2}$. Continuing this way one can reason along the same lines that for $c \geq k+2$ the profile $\mathcal{L}_{c-k-1}$, where $\mathcal{L}_{c-k-1}$ is the same as $\mathcal{L}_{c-k-2}$ except for agent c reporting $[g_i + \gamma, g_{i+1}]$ instead of g_{i+1}, the output p_c associated with $\mathcal{L}_c$ is equal to g_{i+1}.

Given the observations above, consider the profile $\mathcal{L}_{n-k-1}$ where the last $n-k-1$ agents report $[g_i + \gamma, g_{i+1}]$ and the rest of the agents report g_i. Now, one can calculate using Algorithm 9 (in Appendix B.2) that the minimax optimal solution $p_{opt} = g_i$. Also, we know from above that $p_{n-k-1} = \mathcal{M}(\mathcal{L}_{n-k-1}) = g_{i+1}$. And so, using the fact that $d_i = g_{i+1} - g_i \geq \frac{\delta}{2}$, we can now use the expressions from Lemma 16 to see that $\text{maxRegret}(p_{n-k-1}, \mathcal{I}) - \text{OMV}_{AC}(\mathcal{I}) \geq \frac{(2k-2)(g_{i+1}-g_i-\gamma)}{2k}$, which for an appropriately chosen value of n and γ is greater than or equal to $\frac{\delta}{2} - \epsilon$ for any $\epsilon > 0$.

Finally, combining the two cases above, we have our lower bound. $\square$

3.5 Implementing the Maximum Cost Objective

In this section we turn our attention to the objective of minimizing the maximum cost (sometimes succinctly referred to as maxCost and written as MC) which is another well-studied objective function in the context of the facility location problem. In the standard setting where the reports are exact, the maximum cost associated with locating a facility at p is defined as $\max_{i\in[n]} C(x_i, p)$ and if we assume without loss of generality that the x_is are in sorted order, then one can easily see that the optimal solution to this objective is to locate the facility at $p = \frac{x_1+x_n}{2}$. However, unlike in the case of the average cost objective that was considered in Section 3.4, one cannot design an optimal strategyproof mechanism even when the reports are exact, and it is known that the best one can do in terms of additive approximation is to achieve a bound of $\frac{B}{4}$ in the case of deterministic mechanisms and $\frac{B}{6}$ in the case of randomized mechanisms [GT17, Theorems 5, 15].

Now, coming to our setting, unlike in the case of the average cost objective, calculating the minimax optimal solution is straightforward in this case. In fact, given the candidate locations $[a_i, b_i]$ for all i, if $L_1, \ldots, L_n$ and $R_1, \ldots, R_n$ denote the sorted order of the points in $\{a_i\}_{i\in[n]}$ and $\{b_i\}_{i\in[n]}$, respectively, then it is not too hard to show that the minimax optimal solution is the point $\frac{L_1+R_1+L_n+R_n}{4}$ (a complete discussion on how to find the minimax optimal solution when using the maximum cost objective is in Appendix B.4). Therefore, below we directly move on to implementation using the solution concepts defined in Section 3.2.1.

3.5.1 Implementation in very weakly dominant strategies

In the case of the maximum cost objective we again see that very weak dominance is too strong a solution concept as even here it turns out that we cannot do any better than the naive mechanism which always outputs the point $\frac{B}{2}$ as the solution. The following theorem, which can be proved by proceeding exactly like in the proof of Theorem 18, formalizes this statement.

Theorem 24. *Given a $\delta \in (0, B]$, let $\mathcal{M} = (X, F)$ be a deterministic mechanism that implements α-OMV$_{MC}$ in very weakly dominant strategies for a δ-uncertain-facility-location-game. Then, $\alpha \geq \frac{B}{2}$.*

Given the negative result, we move on to implementation in minimax dominant strategies in the hope of getting an analogous positive result as Theorem 21.

3.5.2 Implementation in minimax dominant strategies

When it comes to implementation in minimax dominant strategies, we again see that even in the case of the maxCost objective function one can do a lot better under this solution concept than under very weak dominance. But before we see the exact bounds one can obtain here, recall that Theorem 19 rules out the existence of mechanisms that are anonymous, minimax dominant, and onto. Hence, our focus will be on non-onto mechanisms. We note that the ideas in the following section can be broadly described as being similar to the ones in Section 3.4.3.1 since here, too, we focus on similar "grid-based" mechanisms.

3.5.2.1 Non-onto mechanisms

We show that there exists a mechanism, $\frac{\delta}{2}$-equispaced-phantom-half, that implements $\left(\frac{B}{4} + \frac{3\delta}{8}\right)$-OMV$_{MC}$ in minimax dominant strategies. The mechanism, which can be considered as an extension to the phantom-half mechanism introduced by Golomb and Tzamos [GT17],[10] is similar to the $\frac{\delta}{2}$-equispaced-median mechanism shown in Algorithm 2. Hence, we only highlight the changes in the description below.

$\frac{\delta}{2}$-equispaced-phantom-half. Consider the mechanism described in Algorithm 2. We need to make only two changes: *i)* instead of the definition of A used in Algorithm 2, we

[10] Given a profile of exact reports $(x_1, \ldots, x_n)$, the phantom-half mechanism returns the median of x_{min}, x_{max} and $B/2$, where $x_{min} = \min_i\{x_i\}$ and $x_{max} = \max_i\{x_i\}$.

define it to be the set $\{g_1, \ldots, g_j, \ldots, g_m\}$, where $g_j = \frac{B}{2}, g_{i+1} - g_i = \frac{\delta}{2}$, for $1 \leq i \leq k-1$, $g_0 \geq 0$, and $g_m \leq B$. *ii)* instead of returning the median of the l_is in line 21, we return the median of the points ℓ_{min}, $\frac{B}{2}$, and ℓ_{max}, where $\ell_{min} = \min_i\{\ell_i\}$ and $\ell_{max} = \max_i\{\ell_i\}$.

Below, we show that the $\frac{\delta}{2}$-equispaced-phantom-half mechanism described above implements $\left(\frac{B}{4} + \frac{3\delta}{8}\right)$-OMV$_{MC}$ in minimax dominant strategies.

Theorem 25. *Given a $\delta \in [0, \frac{2B}{3}]$, the $\frac{\delta}{2}$-equispaced-phantom-half mechanism is anonymous and one that implements $\left(\frac{B}{4} + \frac{3\delta}{8}\right)$-OMV$_{MC}$ in minimax dominant strategies for a δ-uncertain-facility-location-game.*

Proof. The proof that the $\frac{\delta}{2}$-equispaced-phantom-half mechanism is minimax dominant is very similar to the proof of Lemma 20 where we show that the $\frac{\delta}{2}$-equispaced-median mechanism is minimax dominant. Therefore, below we only show the approximation bound.

We begin by defining some notation and making a few observations. Let $\mathcal{L}_0 = ([a_1, b_1], \ldots, [a_n, b_n])$ be a profile of candidate locations of the agents and let $L_1, \ldots, L_n$ and $R_1, \ldots, R_n$ be the sorted order of the a_is and b_is, respectively. Also, let ℓ_i be the ℓ_i associated with the agent i in the mechanism (see Algorithm 2), $\ell_{min} = \min_i\{\ell_i\}$, and $\ell_{max} = \max_i\{\ell_i\}$. From the mechanism we know that $\ell_i \in [x_i, y_i]$, where x_i and y_i are as defined in the mechanism (see lines 3–4 in Algorithm 2). Additionally, for two agents i and j, if $a_i \leq a_j$, then it is clear from the definition of x_ks (see line 3 in Algorithm 2) that $x_i \leq x_j$. Similarly, it again follows from the definition of y_ks that for two agent i and j, if $b_i \leq b_j$, then $y_i \leq y_j$.

Next, consider the agent associated with L_1 (i.e., the agent who reports the smallest left endpoint). Without loss of generality we can assume that this agent is agent 1. From the discussion above we know that for every agent $i > 1$, we have that $\ell_i \geq x_i \geq x_1$ (since their left endpoints are at least L_1). Therefore, $\ell_{min} = \min_j\{\ell_j\} \geq x_1$, and since $x_1 \geq L_1 - \frac{\delta}{4}$ (this easily follows from the definition of x_i and the fact that the points in A are placed at a distance of $\frac{\delta}{2}$ apart), we have that $\ell_{min} \geq L_1 - \frac{\delta}{4}$. Now, one can employ a similar line of reasoning to see all of the following: $\ell_{max} \leq R_n + \frac{\delta}{4}$, $\ell_{min} \leq R_1 + \frac{\delta}{4}$, and $\ell_{max} \geq L_n - \frac{\delta}{4}$.

Given all the observations above, we are now ready to prove our bound. To do this, let us consider the following cases. In each of these cases, we will show that the output p of the $\frac{\delta}{2}$-equispaced-phantom-half mechanism is at a distance of at most $(\frac{B}{4} + \frac{3\delta}{8})$ from p_{opt}, where from Proposition 44, which is in Appendix B.4, we know that $p_{opt} = \frac{L_1+R_1+L_n+R_n}{4}$.

Case 1: $p = \frac{B}{2}$**.** In this case, we have,

$$|p_{opt} - p| = \max\left\{\frac{R_1 + R_n + L_1 + L_n}{4} - p, p - \frac{R_1 + R_n + L_1 + L_n}{4}\right\}$$

$$= \max\left\{\frac{R_1 - L_1}{4} + \frac{R_n + L_n}{4} + \frac{L_1}{2} - p, p - \frac{R_n}{2} + \frac{R_n - L_n}{4} - \frac{R_1 + L_1}{4}\right\}$$

$$\le \max\left\{\frac{\delta}{4} + \frac{\ell_{min} + \frac{\delta}{4}}{2}, \frac{B}{2} - \frac{(\ell_{max} - \frac{\delta}{4})}{2} + \frac{\delta}{4}\right\}$$

$$\left(\text{since } p = \tfrac{B}{2},\ R_i - L_i \le \delta \text{ (using Claim 15)},\ \ell_{min} \ge L_1 - \tfrac{\delta}{4}, \text{ and } \ell_{max} \le R_n + \tfrac{\delta}{4}\right)$$

$$\le \max\left\{\frac{3\delta}{8} + \frac{B}{4}, \frac{B}{4} + \frac{3\delta}{8}\right\} \qquad \left(\text{since } \ell_{min} \le \tfrac{B}{2} \le \ell_{max}\right)$$

$$= \frac{B}{4} + \frac{3\delta}{8}.$$

Case 2: $p = \ell_{min}$**.** In this case, we have,

$$|p_{opt} - p| = \max\left\{\frac{R_1 + R_n + L_1 + L_n}{4} - p, p - \frac{R_1 + R_n + L_1 + L_n}{4}\right\}$$

$$\le \max\left\{\frac{R_1 - L_1}{4} + \frac{R_n + L_n}{4} + \frac{L_1}{2} - p, p - \frac{R_1}{2} - \frac{L_1}{2}\right\} \qquad \left(\text{since } L_1 \le L_n,\ R_1 \le R_n\right)$$

$$\le \max\left\{\frac{\delta}{4} + \frac{B}{2} + \frac{\delta}{8} - \frac{\ell_{min}}{2}, \frac{R_1}{2} - \frac{L_1}{2} + \frac{\delta}{4}\right\}$$

$$\left(\text{since } p = \ell_{min},\ R_i - L_i \le \delta \text{ (using Claim 15)},\ \ell_{min} \ge L_1 - \tfrac{\delta}{4}, \text{ and } \ell_{min} \le R_1 + \tfrac{\delta}{4}\right)$$

$$\le \max\left\{\frac{3\delta}{8} + \frac{B}{4}, \frac{3\delta}{4}\right\} \qquad \left(\text{since } \tfrac{B}{2} \le \ell_{min} \le \ell_{max}\right)$$

$$= \frac{B}{4} + \frac{3\delta}{8}. \qquad \left(\text{since } \delta \le \tfrac{2B}{3}\right)$$

Case 3: $p = \ell_{max}$**.** This can be handled analogously as in Case 2.

Finally, for an instance $\mathcal{I}$, since from all the cases above we have that $|p_{opt} - p| \le (\frac{B}{4} + \frac{3\delta}{8})$, maxRegret$(p, \mathcal{I}) - \text{OMV}_{MC}(\mathcal{I})$ is also bounded by $\left(\frac{B}{4} + \frac{3\delta}{8}\right)$ (this can seen by using Lemma 43, which is in Appendix B.4). □

Given this result, it is natural to ask if we have a lower bound like the one in Section 3.4.3.1. Unfortunately, the only answer we have is the obvious lower bound of $\frac{B}{4}$ that follows from the result of Golomb and Tzamos [GT17, Theorem 15] who showed that under exact reports, and when using deterministic mechanisms, one cannot achieve a bound lower than $\frac{B}{4}$.

3.6 Discussion

The standard assumption in mechanism design that the agents are precisely aware of their complete preferences may not be realistic in many situations. Hence, we believe that there is a need to look at models that account for partially informed agents and, at the same time, design mechanisms that provide robust guarantees. In this chapter, we looked at such a model in the context of the classic single-facility location problem, where an agent specifies an interval instead of an exact location, and our focus was on designing robust mechanisms that perform well with respect to all the possible underlying true locations of the agents. Towards this end, we looked at two solution concepts, very weak dominance and minimax dominance, and we showed that, with respect to both the objective functions we considered, while it was not possible to achieve any good mechanism in the context of the former solution concept, extensions to the classical optimal mechanisms—i.e., mechanisms that perform optimally in the classical setting where the agents exactly know their locations—performed significantly better under the latter, weaker, solution concept. Our results are summarized in Table 3.1.

There are some immediate open questions in the context of the problem we considered like looking at randomized mechanisms, providing tighter bounds, and potentially even finding deterministic mechanisms that perform better than the ones we showed. More broadly, we believe that it will be interesting to revisit the classic problems in mechanism design, see if one can look at models which take into account partially informed agents, and design mechanisms where one can explicitly relate the performance of the mechanism with the quality of preference information.

Chapter 4

Improving Welfare in One-Sided Matching Using Simple Threshold Queries

4.1 Introduction

The previous two chapters looked at the question of designing 'robust' algorithms, where, broadly, by 'robust' we meant algorithms that when given access to incomplete preference information performed well with respect to all the possible underlying complete preferences. Crucially, the assumption in both the previous chapters was that the algorithm has to work with given incomplete information. However, this might not always be required, for it might be possible to ask the agents to refine their preferences further. In this chapter, we look at such a setting in the context of *one-sided matching problems*. One-sided matching problems model scenarios where there is a set of n agents who have preferences over a set of m objects and the goal is to assign each agent to at most one object. Such problems are ubiquitous. For instance, the case where the objects are houses represents the well-studied housing allocation problem (e.g., see [HZ79; AS98; Abr+04]) or, when agents have initial endowments, the housing market problem (e.g., see [SS74; RP77; AS99]). Other examples include assigning faculty members to school committees, workers to tasks, etc.

Most of the literature on one-sided matching problems typically assume that the agents have an acceptable set of objects and that they submit a preference order over the acceptable set. Given this, the standard objective is to come up with an assignment of agent to objects

	object 1	object 2	object 3
agent 1	0.9	0.1	0
agent 2	0.9	0.1	0
agent 3	0.51	0.49	0

Table 4.1: Example which illustrates how there is a loss in welfare due to not accounting for preference intensities.

(henceforth, matching) such that it satisfies some notion of economic efficiency like, e.g., Pareto optimality or rank-maximality. Informally, a matching is Pareto optimal if there is no other matching where all the agents are assigned to an object that they like at least as much as the one they are assigned to currently and if there is at least one agent who strictly prefers the new assignment. A matching is rank-maximal if it maximizes the number of agents who are matched to their first choice object, and, subject to that, it maximizes the number of agents matched to their second choice object, and so on.

Although matchings that satisfy such notions are better than arbitrary matchings, their main drawback is that they do not take into account agents' preference intensities. To illustrate this, consider the simple example in Table 4.1 where there are three agents and three objects. The agents all have the same ordinal preferences, but have different preference intensities. For instance, agent 3 only has a slight preference for object 1 over object 2 (0.51 vs 0.49), whereas agents 1 and 2 both prefer object 1 much more than object 2 (0.9 vs 0.1). In this example, if we were concerned about Pareto optimality or rank-maximality, then it is easy to see that any matching satisfies both these notions. However, it is also clear that any matching that matches agent 3 to object 3 is worse in terms of overall welfare, since agent 3 experiences a much larger loss in utility when matched to object 3 instead of object 2, whereas for agents 1 and 2 this difference is very small.

The observation that there might be a loss in welfare due to not capturing preference intensities (henceforth, cardinal utilities) is not new, and in particular, this has been a much debated issue surrounding various *school-choice* mechanisms (e.g., see [ACY11; ACY15]).[1] This has also lead to, for instance, proposals for new school choice mechanisms that ask agents to provide some extra information along with their ordinal preferences [ACY15]. Our work here is partially motivated by this line of work, but takes a more computational and algorithmic approach to this issue that is similar to the flurry of work that looks at *distortion*—which is essentially the cost of using only ordinal information—in various

[1] School-choice is the problem of allocating students to schools. This is essentially a one-sided matching problem, except that here the schools usually have, what are called, *priorities*—which is an ordering over students that is used only when the school is over-demanded (see, e.g., [ACY11] for formal definitions).

settings (e.g., see [PR06; Bou+15; AS16; AZ17; GKM17; AA18]).

In particular, given a one-sided matching instance (which is the set of agents, objects, and the agents' preferences), we are still interested in computing matchings that satisfy some notion of economic efficiency, say, X, but at the same time our aim is to account for the agents' cardinal utilities. We do this by aiming to design algorithms that always return a matching that satisfies X and at the same time achieves a good approximation to the optimal welfare among all matchings that satisfy X—i.e., the maximum utilitarian social welfare achievable among matchings that satisfy X. Now, of course, one way to achieve this is to just ask agents directly for their cardinal utilities, since once we have them we can compute the matching with the best welfare among ones that satisfy X. However, as one can imagine, asking agents for their cardinal utilities is cognitively non-trivial, since even attributing values to objects is not easy in many scenarios. Therefore, here our focus is to achieve a middle-ground between completely ordinal and completely cardinal elicitation, and to do this, we use threshold queries which ask an agent if their value for an object is at least some real number v. Our goal is to ask each agent a small number of such queries and then use it to pick a matching that achieves a good approximation as described above.

Although the general idea of using queries to elicit some information regarding cardinal utilities is not new and has even been considered in the context of one-sided matching in a recent paper [Ama+21], there are some differences. The main difference between our approach here and the one by Amanatidis et al. [Ama+21] is that the focus in their paper is to come up with algorithms that have low distortion, whereas our focus is on algorithms that, for a particular notion of economic efficiency X, always produce a matching that satisfies X and has a good approximation to the optimal welfare among all matchings that satisfy X. We believe that while achieving low distortion might be a good objective in certain settings, it is too reliant on cardinal preferences—the presence of which is already an assumption. Hence, comparing different algorithms solely based on this value (i.e., the distortion they achieve) does not seem ideal. Our approach, on the other hand, is less reliant on cardinal information, using it only to pick a matching from the set of matchings that satisfy some property X, which in turn is dependent only on the ordinal information that is arguably more robust. Moreover, the query model used by Amanatidis et al. [Ama+21] is much stronger than the one we employ. Ours just asks for a binary answer to whether the value of an item is greater than some real number v, whereas in their model a query asks an agent to reveal their utility for the object, which in turn is cognitively much more demanding. These differences mean that there are no direct overlaps between our results here and that of Amanatidis et al. [Ama+21].

	Ordinal algorithms	**Adaptive threshold query algorithms** (for any $\epsilon > 0$, $O(c \log n)$ queries per agent, where $c = \left\lceil \frac{\log(n^2 \cdot 1/\epsilon)}{\log(1+\epsilon/2)} \right\rceil$)	**Non-adaptive threshold query algorithms** (at most 1 query per (agent, object) pair)
unit-sum valuations	UB: $O(n^2)$ [Theorem 26] LB: $\Omega(n^2)$ [Theorem 26]	$1+\epsilon$ [Theorem 28]	UB: $O(n^{2/3})$ [Theorems 29 and 31] LB: $\Omega(\sqrt{n})$ [Theorem 32]
unit-range valuations	UB: $O(n)$ [Theorem 46] LB: $\Omega(n)$ [Theorem 46]	$1+\epsilon$ [Theorem 28]	UB: $O(\sqrt{n})$ [Theorem 47] LB: $\Omega(\sqrt{n})$ [Theorem 48]

Table 4.2: Summary of our results. For X, where X is one of the properties in the set {Pareto optimal, rank-maximal, max-cardinality rank-maximal, fair}, an upper bound (UB) of α indicates that there is a deterministic algorithm that always produces a matching that satisfies X and achieves an α-approximation to the optimal welfare among matchings that satisfy X. A lower bound (LB) of β indicates that there is no deterministic algorithm that produces a matching that satisfies X and achieves a β-approximation to the optimal welfare among matchings that satisfy X.

4.1.1 Our contributions

We consider the following four well-studied types of matchings that satisfy a specific notion of economic efficiency: *i)* Pareto optimal matchings, *ii)* rank-maximal matchings, *iii)* max-cardinality rank-maximal matchings, and *iv)* fair matchings. As mentioned above, for each of these types, our goal is to find deterministic algorithms that always output a matching of the corresponding type and one that achieves a good approximation to the optimal welfare among all matchings of that type. Towards this end, we consider two kinds of cardinal utilities, namely, unit-sum and unit-range valuations, and show the following results, which are summarized in Table 4.2.

- We first look at adaptive algorithms—i.e., algorithms that are able to change their queries depending on how agents answer its previous queries—and show how for each of the notions mentioned above and for any $\epsilon > 0$, there is a deterministic algorithm that asks $O(c \log n)$ queries per agent, where $c = \left\lceil \frac{\log(n^2 \cdot 1/\epsilon)}{\log(1+\epsilon/2)} \right\rceil$, and returns a matching that satisfies this notion and also achieves a $(1+\epsilon)$-approximation to the optimal welfare among all matchings that satisfy this notion when the agents have unit-sum or unit-range valuations.

- While the previous result achieves the best possible approximation one can hope for—and in particular it results in an $O(1)$-approximation with just $O(\log^2 n)$ queries per agent—we believe that the fact it is adaptive is not ideal because of the following reasons: *i)* Adaptive algorithms may not be practical in many settings since it involves waiting for the agents to respond, and potentially having them respond multiple times. *ii)* For every agent, the algorithm mentioned above potentially asks multiple queries with respect to the same object and as this number increases, one could argue that it defeats the real purpose of such algorithms—since responding to them entails that the agents are somewhat sure about their cardinal utilities.

  As a result, we focus on non-adaptive algorithms, which address the first issue mentioned above, and in order to address the second one, we consider a very special type of non-adaptive algorithm—one which is allowed to ask at most one query per (agent, object) pair. We believe that this is the most practical setting to consider for this problem, since this means every agent is asked to provide just one extra bit of information per object.[2]

  For this setting, we show how for each of the notions considered, there is a deterministic algorithm that returns a matching that satisfies this notion and also achieves an $O(n^{2/3})$-approximation to the optimal welfare among all matchings that satisfy this notion when the agents have unit-sum valuations. We also derive a similar result for the unit-range case, showing an algorithm that achieves an $O(\sqrt{n})$-approximation. Note that these bounds are as opposed to a $\Theta(n^2)$ and $\Theta(n)$ approximation that is achievable when using only ordinal preferences and when the agents have unit-sum and unit-range valuations, respectively.

- Finally, we also show that, for all the notions considered, any deterministic algorithm that uses at most one query per (agent, object) pair can only achieve an approximation factor of $\Omega(\sqrt{n})$, both for the case when agents have unit-sum or unit-range valuations. Note that for the unit-range case, this bound in turn is asymptotically tight.

[2] Also, deploying this seems easier, since instead of the current system which presumably just asks the agents to list their preferences, now all that needs to be done is to have a checkbox next to it, indicating whether their answer is a "Yes" or "No" w.r.t. a certain threshold query.

4.1.2 Related work

One-sided matching scenarios are ubiquitous and have been well-studied, especially as the housing allocation or housing market problem both in economics (e.g., see [SS74; HZ79; RP77; AS98; AS99; SÜ10]) and in theoretical computer science and computational economics (e.g., see [Abr+04; Irv04; Abr+06; FFZ14; Ama+21]). Unlike here where we assume that the agents have underlying cardinal utilities that are consistent with their ordinal preferences, most of this literature assumes that the agents only have ordinal preferences, and their goal is usually to find matchings that satisfy some notion of economic efficiency like, e.g., Pareto optimal matchings [SS74; Abr+04], rank-maximal matchings [Irv04; Irv+06], max-cardinality rank-maximal matchings [MM05; Abr+06], and fair matchings [MM05; Hua+13].

As mentioned in the Introduction, part of the motivation for our work is derived from work on the school choice problem that talks about the loss in welfare due to not taking the preference intensities into account (e.g., [ACY11; ACY15]). Our concern here is on similar lines, but we take a more computational approach to the problem which is reminiscent of the vast body of work on distortion (e.g., see [PR06; Bou+15; AS16; AZ17; GKM17; AA18]). However, unlike this body of work which aims to calculate the worst-case loss in welfare due to only having ordinal preferences, we assume that, in addition to their ordinal preferences, it is also possible to obtain some information about the agents' cardinal utilities. This in turn is similar to the approach taken by Abramowitz, Anshelevich, and Zhu [AAZ19] in the context of voting and more closely to the ones by Amanatidis et al. [Ama+20; Ama+21] in the context of voting and one-sided matching, respectively. In the work by Abramowitz, Anshelevich, and Zhu [AAZ19] it assumed that in addition to the ordinal preferences there is also some information regarding how many agents prefer candidate P over Q above a certain threshold, whereas in the work by Amanatidis et al. [Ama+20; Ama+21], and as is the case here, it is assumed that the one can use some specific type of query in order to get more information regarding the cardinal utilities.

Finally, our work is also related to the work that studies the communication complexity of voting protocols [Man+19; MSW20], to the work on participatory budgeting which compares different elicitation methods based on the distortion achieved (e.g., see [Goe+19; Ben+20]), and is more broadly in line with the growing body of work that explicitly aims to make mechanisms or algorithms more robust, by either making use of coarse preference information [CMA12; CMA14; ML19] or by making sure that the algorithms designed produce solutions that work "well" (in the approximation sense) even under slightly modified inputs [SYE13; Bre+17; ML18; MV18; CSS19].

4.2 Preliminaries

For $k \in \mathbb{Z}^+$, let $[k]$ denote the set $\{1, \ldots, k\}$. We use $\mathcal{N}$, where $|\mathcal{N}| = n$, to denote the set of agents $\{a_1, \ldots, a_n\}$, and use $\mathcal{H}$, where $|\mathcal{H}| = n$, to denote the set of objects $\{h_1, \ldots, h_n\}$. We refer to a_i as agent i and h_j as object j. Every agent a_i is assumed to have a weak order P_i over a subset of objects $A_i \subseteq \mathcal{H}$. For an agent a_i, A_i indicates the set of objects a_i is willing to be matched to and we refer to A_i as the *acceptable set* of a_i and assume that $|A_i| \geq 1$. We use $\mathcal{P} = (P_1, \ldots, P_n)$ to refer to the weak orders of all the agents in $\mathcal{N}$ and refer to $\mathcal{P}$ as the *preference profile* of the agents. For an agent a_i, and for two objects $h_j, h_k \in A_i$, we use $h_j \succ_i h_k$ to denote that a_i strictly prefers h_j over h_k, and use $h_j \succeq_i h_k$ to indicate that h_j is either strictly preferred or considered to be equivalent to h_k. We refer to $\mathcal{I} = (\mathcal{N}, \mathcal{H}, \mathcal{P} = (P_1, \ldots, P_n))$ as an *instance*, which encodes all the information about the agents, objects, and the agents' preferences, and use $\mathbb{I}$ to denote the set of all possible instances.

Given an instance $\mathcal{I} = (\mathcal{N}, \mathcal{H}, \mathcal{P} = (P_1, \ldots, P_n))$, we use $\mathbb{G}_{\mathcal{I}} = (\mathcal{N} \cup \mathcal{H}, \mathcal{E})$ to denote the bipartite graph where there as an edge $(a_i, h_j) \in \mathcal{E}$ if $h_j \in A_i$. We refer to $\mathbb{G}_{\mathcal{I}}$ as the graph induced by $\mathcal{I}$ and refer to $e = (a_i, h_j) \in \mathcal{E}$ as a rank-k edge if $|\mathcal{U}_{ij}| = k - 1$, where $\mathcal{U}_{ij} = \{h_\ell \in A_i \mid h_\ell \succ_i h_j\}$. We also use $rank(a_i, h_j)$ to denote the k such that (a_i, h_j) is a rank-k edge and refer to an object h_j as a_i's rank-k (or k-th choice) object if $rank(a_i, h_j) = k$.

Although the model described thus far is the standard model in one-sided matching, here we additionally assume that each agent a_i has a cardinal utility function $v_i \colon \mathcal{H} \to [0, 1]$, which is consistent with the preference order P_i (meaning, $h_1 \succeq_i h_2 \Leftrightarrow v_i(h_1) \geq v_i(h_2)$); we assume that if $h \notin A_i$, then $v_i(h) = 0$. In this work we consider two specific kinds of (normalized) valuation functions which are defined below, and use $v = (v_1, \ldots, v_n)$ to denote the valuation profile of agents and $\mathcal{V}_{\mathcal{I}}$ to denote the set of all possible valuation profiles that are consistent with the given preference profile in $\mathcal{I}$.

1. Unit-sum valuations: Agents are said to have unit-sum valuations if for each agent i, v_i is such that $\sum_{h \in A_i} v_i(h) = 1$.

2. Unit-range valuations: Agents have unit-range valuations if for each agent i, there exists $h_j, h_k \in A_i$ such that $h_j \succ_i h_k$, and $\max_{h \in A_i} v_i(h) = 1$ and $\min_{h \in A_i} v_i(h) = 0$. In words, the most preferred objects have value 1, the least preferred objects have value 0, and every other acceptable object has value between 0 and 1.

Note that information about the cardinal utilities is not part of an instance $\mathcal{I}$. Also, note that although the "internal utilities" of the agents may not be normalized, we have to assume some normalization in order to only use threshold queries. Given this, both unit-sum and unit-range arise from two natural ways to normalize agents' "internal utilities". More precisely, if $u_i \colon \mathcal{H} \to \mathbb{R}_{\geq 0}$ is the "internal utility" of an agent, then, for all $h \in_i A$, the corresponding unit-sum and unit-range valuation functions are, respectively,

$$v_i(h) = \frac{u_i(h)}{\sum_{h \in A_i} u_i(h)} \qquad \text{and} \qquad v_i(h) = \frac{u_i(h) - \min_{h \in A_i} u_i(h)}{\max_{h \in A_i} u_i(h) - \min_{h \in A_i} u_i(h)}.$$

For an instance $\mathcal{I}$, we are interested in matchings that assign agents to objects, and a *matching* of agents to objects is a bijection $\mu \colon \mathcal{N} \to \mathcal{H}$ and, for $c \in \mathcal{N} \cup \mathcal{H}$, we refer to $\mu(c)$ as c's *partner* in μ or as c's *allocation* in μ. Alternatively, a matching is also defined as a collection of edges μ in $\mathbb{G}_{\mathcal{I}}$ such that each vertex is part of at most one edge in μ. We use $\mathcal{M}_{\mathbb{G}_{\mathcal{I}}}$ to denote the set of all possible matchings in $\mathbb{G}_{\mathcal{I}}$. Although for a given instance there are several possible matchings, we are interested in matchings that satisfy some notion of economic efficiency; these are defined next.

4.2.1 Notions of economic efficiency

We consider the following well-studied notions: Pareto optimal matchings [SS74; Abr+04], rank-maximal matchings [Irv04; Irv+06], max-cardinality rank-maximal matchings [MM05; Abr+06], and fair matchings [MM05; Hua+13]. The latter three are different ways to strengthen Pareto optimality and are together referred in the rest of this chapter as *signature-based* matchings.

Definition 10 (Pareto optimal matchings)**.** Given an instance $\mathcal{I} = (\mathcal{N}, \mathcal{H}, \mathcal{P} = (P_1, \ldots, P_n))$, a matching $\mu \in \mathcal{M}_{\mathbb{G}_{\mathcal{I}}}$ is Pareto optimal (PO) w.r.t. $\mathcal{I}$ if

$$\forall \mu' \in \mathcal{M}_{\mathbb{G}_{\mathcal{I}}} : \; (\exists a_i \in \mathcal{N}, \mu'(a_i) \succ_i \mu(a_i)) \Rightarrow (\exists a_j \in \mathcal{N}, \mu'(a_j) \prec_j \mu(a_j)) \, .$$

In words, a matching μ is Pareto optimal if there is no other matching μ' such that every agent weakly-prefers their allocation in μ' over their allocation in μ and at least one agent (strictly) prefers their allocation in μ' over their allocation in μ.

Definition 11 (Signature-based matchings)**.** Given an instance $\mathcal{I} = (\mathcal{N}, \mathcal{H}, \mathcal{P} = (P_1, \ldots, P_n))$, and a matching $\mu \in \mathcal{M}_{\mathbb{G}_{\mathcal{I}}}$, let s_i denote the number of agents that are matched to a rank-i edge in μ. Then, μ is

- **rank-maximal** if μ maximizes the number of agents who are matched to a rank-1 edge and, subject to that, it maximizes the number of agents who are matched to rank-2 edges, and so on. Formally, if we associate an n-tuple $(s_1, \ldots, s_n)$ with every matching in $\mathcal{M}_{\mathbb{G}_\mathcal{I}}$, then μ is the matching that has lexicographically the best n-tuple $(s_1, \ldots, s_n)$ associated with it.

- **max-cardinality rank-maximal** if μ is a maximum cardinality matching and, subject to that, is also rank-maximal. Formally, if we associate an $(n+1)$-tuple $(\sum_{i=1}^{n} s_i, s_1, \ldots, s_n)$ with every matching in $\mathcal{M}_{\mathbb{G}_\mathcal{I}}$, then μ is the matching that has lexicographically the best $(n+1)$-tuple $(\sum_{i=1}^{n} s_i, s_1, \ldots, s_n)$ associated with it.

- **fair** if μ is a maximum cardinality matching and, subject to that, minimizes the number of agents who are matched to a rank-n edge and, subject to that, minimizes the number of agents who are matched to a rank-$(n-1)$ edge, and so on. Formally, if we associate an $(n+1)$-tuple $(\sum_{i=1}^{n} s_i, -s_n, -s_{n-1}, \ldots, -s_1)$ with every matching in $\mathcal{M}_{\mathbb{G}_\mathcal{I}}$, then μ is the matching that has lexicographically the best $(n+1)$-tuple $(\sum_{i=1}^{n} s_i, -s_n, -s_{n-1}, \ldots, -s_1)$ associated with it.

For each type of signature-based matching defined above and a matching of that type, we refer to the corresponding tuple, as defined above, to be the matching's signature. That is, for instance, for a max-cardinality rank-maximal matching μ, signature of μ refers to the $(n+1)$-tuple $(\sum_{i=1}^{n} s_i, s_1, \ldots, s_n)$. The example below illustrates the difference between Pareto optimal, rank-maximal, and fair matchings for an instance.

Example 1. Consider an instance with 7 agents and 7 objects, where the preferences of the agents are as defined in the table below. Here each column corresponds to the strict preferences of an agent and if the column corresponding to agent, say, a_i is h_1, h_2, h_3, then this implies that a_i prefers h_1 the most, h_2 second most, and so on. For this instance, the

a_1	a_2	a_3	a_4	a_5	a_6	a_7
h_1	h_2	h_1	h_3	h_1	h_1	h_1
h_4	h_5	h_3	h_6	h_4	h_2	h_2
h_3	h_6			h_5	h_4	h_5
h_7						

matching (of size 6) that corresponds to each agent being matched to the object (if any) coloured blue in its column is an example of a Pareto optimal matching. Similarly, the matching (of size 5, with signature $(3, 1, 1, 0, 0, 0)$) where each agent is matched to the object (if any) coloured red is an example of a rank-maximal matching, and the matching (of size 7, with signature $(7, 0, 0, -1, 0, -5, -1)$) where each is matched to the object (if any) coloured yellow is an example of a fair matching.

It is well-known (see [Irv04; Irv+06; MM05; Hua+13; Mic07]) that signature-based matchings can be reduced to an instance of the following problem, which we refer to as priority-$\boldsymbol{p}$ matchings, for a given $\boldsymbol{p} = (p_1, \cdots, p_n)$.

Definition 12. Given an instance $\mathcal{I} = (\mathcal{N}, \mathcal{H}, \mathcal{P} = (P_1, \ldots, P_n))$ and a priority vector $\boldsymbol{p} = (p_1, \ldots, p_n)$, where $\forall i \in [n], p_i \in \mathbb{Z}_{\geq 0}$ and $\exists j, k \in [n]$ such that $p_j \neq p_k$, a matching $\mu \in \mathcal{M}_{\mathbb{G}_\mathcal{I}}$ is said to be a priority-$\boldsymbol{p}$ matching if μ is a matching of maximum weight in $\mathcal{M}_{\mathbb{G}_\mathcal{I}}$, where a rank-$r$ edge in $\mathbb{G}_\mathcal{I}$ is assigned the weight p_r.

In particular, given an instance $\mathcal{I}$, we can show that,[3]

- when $p_j = n^{2(n-j+1)}$ for all $j \in [n]$, a matching is a priority-$\boldsymbol{p}$ matching if and only if it is rank-maximal matching w.r.t. $\mathcal{I}$.
- when $p_j = n^{2n} + n^{2(n-j)}$ for all $j \in [n]$, a matching is a priority-$\boldsymbol{p}$ matching if and only if it is a max-cardinality rank-maximal matching w.r.t. $\mathcal{I}$.
- when $p_j = 4n^{2n} - 2n^{j-1}$ for all $j \in [n]$, a matching is a priority-$\boldsymbol{p}$ matching if and only if it is a fair matching w.r.t. $\mathcal{I}$.

Although priorty-$\boldsymbol{p}$ matchings can potentially be defined for several values of $\boldsymbol{p}$, here we are interested in the three cases described above. Additionally, for ease of exposition, we also sometimes use priority-$\boldsymbol{p}$, where $p_i = 0$ for all $i \in [n]$ to refer to Pareto optimal matchings. Note that this is purely for notational convenience (priority-$\boldsymbol{p}$ matching as defined in Definition 12 is not defined when $p_i = p_j$ for all $i, j \in [n]$), since the algorithms we discuss in the context of Pareto optimal matchings are extensions to the ones for priority-$\boldsymbol{p}$ matchings. Throughout, we use $\mathbb{P}$ to denote the priority vectors of interest. That is, $\mathbb{P} = \{(n^{2n}, \ldots, n^{2(n-j+1)}, \ldots, n^2), (n^{2n} + n^{2(n-1)}, \ldots, n^{2n} + n^{2(n-j)}, \ldots, n^{2n} + n), (4n^{2n} - 2, \ldots, 4n^{2n} - 2n^{j-1}, \ldots, 4n^{2n} - 2n^{n-1})\}$.

[3] The proof of this can be found in Claim 22 in Appendix C.1. Note that such observations have also been made in previous works (e.g., see [Irv04; Irv+06; MM05; Hua+13; Mic07]), although the value of p_js used may be different.

4.2.2 Going beyond completely ordinal or completely cardinal algorithms

Given an instance $\mathcal{I}$, we are interested in deterministic algorithms $\mathcal{A}$ that always output a matching that satisfies one of the economic notions defined in the previous section. However, even when restricted to such matchings, as the example in the Introduction illustrates, there are potentially many choices, and there might be a loss in welfare due to not accounting for the cardinal utilities of the agents. Therefore, ideally we want our algorithm to have small worst-case loss in welfare. Formally, for an instance $\mathcal{I}$, consider the set of matchings $S \subseteq \mathcal{M}_{\mathbb{G}_\mathcal{I}}$ such that S is the set of all Pareto optimal/rank-maximal/max-cardinality rank-maximal/fair matchings in $\mathbb{G}_\mathcal{I}$. Next, for a matching $\mu \in S$, $v \in \mathcal{V}_\mathcal{I}$, and for an edge $e = (a_i, h_j) \in \mu$, let $\text{value}(e) = v_i(h_j)$ and $\text{SW}(\mu \mid v) = \sum_{e \in \mu} \text{value}(e)$, the social welfare of μ given the valuations v. (For notational convenience, when v is clear from the context, we just write $\text{SW}(\mu)$ instead of $\text{SW}(\mu \mid v)$.) Given this, consider a deterministic algorithm $\mathcal{A}$ where, for all $\mathcal{I} \in \mathbb{I}$, $\mathcal{A}(\mathcal{I}) \in S$ and let $\mathcal{L}(\mathcal{A})$, which we refer to as the *worst-case welfare loss* of $\mathcal{A}$, be defined as below.

$$\mathcal{L}(\mathcal{A}) = \max_{\mathcal{I} \in \mathbb{I}} \mathcal{L}(\mathcal{A}, \mathcal{I}), \text{ where } \mathcal{L}(\mathcal{A}, \mathcal{I}) \;=\; \sup_{v \in \mathcal{V}_\mathcal{I}} \frac{\max_{\mu^* \in S} \text{SW}(\mu^* \mid v)}{\text{SW}(\mathcal{A}(\mathcal{I}) \mid v)}. \tag{4.1}$$

As mentioned above, we want algorithms $\mathcal{A}$ that have as small a value of $\mathcal{L}(\mathcal{A})$ as possible. To achieve this, on the one extreme we have completely ordinal algorithms—which are algorithms that only consider the ordinal preferences. We argue below that that any deterministic ordinal algorithm has a very poor worst-case loss in welfare, in particular $\Omega(n^2)$ when agents have unit-sum valuations. The proof of this result appears in Appendix C.3.1. (The corresponding result showing a bound of $\Omega(n)$ for unit-range valuations can be found as Theorem 46 in Appendix C.3.1.)

Theorem 26. *Let X denote one of the properties in the set {Pareto-optimal, rank-maximal, max-cardinality rank-maximal, and fair}. Let $\mathcal{A}$ be a deterministic ordinal algorithm that always produces a matching that satisfies property X. If there are n agents with unit-sum valuation functions, then $\mathcal{L}(\mathcal{A}) \in \Omega(n^2)$. Moreover, this bound is asymptotically tight.*

At the other extreme, when we have access to all the cardinal utilities, we show (see Theorem 27) in the next section how, for all the notions considered here, computing the welfare-optimal matching reduces to the max-weight matching problem. Although this is ideal, as mentioned in the Introduction, asking agents for cardinal utilities might not be reasonable in many situations, as this is a cognitively involved task. Therefore, in this chapter, we aim for a middle-ground between completely ordinal and completely cardinal algorithms. We do this by trying to get at least some information regarding the cardinal preferences by asking the agents certain queries. In particular, we are interested in the following type of query, which we refer to as *binary threshold query*.

Definition 13 (binary threshold query)**.** For an agent a_i, object h_j, and a real number $t_k \in [0, 1]$, a binary threshold query, denoted $\mathcal{Q}(a_i, h_j, t_k)$, asks agent a_i to return 1 (alternatively, asks them to say "Yes") if $v_i(h_j) \geq t_k$, and 0 (alternatively, asks them to say "No") otherwise.

Given an instance $\mathcal{I}$ and answers to a certain number of binary threshold queries, our aim is to design deterministic algorithms $\mathcal{A}$ that minimize the worst-case welfare loss $\mathcal{L}(\mathcal{A})$ and, for all $\mathcal{I} \in \mathbb{I}$, produces a matching in S (i.e., $\mathcal{A}(\mathcal{I}) \in S$), where S is the set of all Pareto optimal/rank-maximal/max-cardinality rank-maximal/fair matchings in $\mathbb{G}_{\mathcal{I}}$.

Remark: Throughout this chapter, we say that, for an $\alpha \geq 1$, an algorithm $\mathcal{A}$ achieves an α-approximation to the optimal social welfare among Pareto-optimal/rank-maximal/max-cardinality rank-maximal/fair matchings if $\mathcal{L}(\mathcal{A}) \leq \alpha$. Also, note that although the ratio defined in (4.1) might seem very similar to the notion of *distortion* that is widely used in computational social choice literature (e.g., see [PR06; AS16]), it is important to note that it is different. Here we are interested in algorithms that produce a matching with a certain property (like Pareto optimality, rank-maximality, etc.) and that has social welfare as close to the optimal welfare achievable with a matching that satisfies the same property of interest, whereas in the context of distortion there is no such restriction. That said, it is also worth noting that in the context of one-sided matching, an algorithm that achieves an α-approximation to the optimal social welfare among Pareto-optimal matchings also has a distortion of α since a welfare-optimal matching is also Pareto optimal.

4.2.3 Finding welfare-optimal priority-$\boldsymbol{p}$ matchings when utilities are known

Before we consider the main question of this chapter, a natural question that arises is on how to compute the welfare-optimal Pareto optimal or welfare-optimal priority-$\boldsymbol{p}$ matchings for the priority vectors of interest, when the agents' utilities are known. Given an instance $\mathcal{I} = (\mathcal{N}, \mathcal{H}, \mathcal{P})$ and valuation functions of the agents $v = (v_1, \ldots, v_n)$, where $v_i \colon \mathcal{H} \to [0, 1]$, the welfare-optimal priority-$\boldsymbol{p}$ problem is to find a matching of maximum welfare among

Input: an instance $\mathcal{I} = (\mathcal{N}, \mathcal{H}, \mathcal{P} = (P_1, \ldots, P_n))$, priorities $\boldsymbol{p} = (p_1, \ldots, p_n)$, where $\boldsymbol{p} \in \mathbb{P}$, and $v = (v_1, \ldots, v_n)$, where $v_i \colon \mathcal{H} \to [0, 1]$
Output: returns a welfare-optimal priority-$\boldsymbol{p}$ matching w.r.t. $\mathcal{I}$
1: $\mathbb{G}_{\mathcal{I}} = (\mathcal{N} \cup \mathcal{H}, \mathcal{E}) \leftarrow$ graph induced by $\mathcal{I}$
2: **for** $e = (a_i, h_j) \in \mathcal{E}$ **do**
3:     $r \leftarrow \text{rank}(a_i, h_j)$
4:     $w_e \leftarrow p_r + v_i(h_j)$
5: **end for**
6: $\mu \leftarrow$ max-weight matching in $\mathbb{G}_{\mathcal{I}}$ with weights $\{w_e\}_{e \in \mathcal{E}}$
7: **return** μ

Algorithm 3: returns a welfare-optimal priority-$\boldsymbol{p}$ matching.

the set of priority-$\boldsymbol{p}$ matchings. Below, we show how for all the priority vectors of interest, computing this reduces to an instance of the max-weight matching problem on $\mathbb{G}_{\mathcal{I}}$.

To see this, first note that finding the welfare-optimal Pareto optimal matching directly reduces to the max-weight matching problem on $\mathbb{G}_{\mathcal{I}}$, where the weight of an edge (a_i, h_j) is $v_i(h_j)$. Therefore, below we show how even the question of computing the welfare-optimal priority-$\boldsymbol{p}$ matchings for $\boldsymbol{p} \in \mathbb{P}$ reduces to the max-weight matching problem on $\mathbb{G}_{\mathcal{I}}$.

Theorem 27. *Given an instance $\mathcal{I} = (\mathcal{N}, \mathcal{H}, \mathcal{P} = (P_1, \ldots, P_n))$, a vector of priorities $\boldsymbol{p} = (p_1, \ldots, p_n)$, where $\boldsymbol{p} \in \mathbb{P}$, and $v = (v_1, \ldots, v_n)$, where $v_i \colon \mathcal{H} \to [0, 1]$, Algorithm 3 returns a welfare-optimal priority-$\boldsymbol{p}$ matching w.r.t. $\mathcal{I}$.*

Proof. Let μ be the matching that is returned by Algorithm 3. First, we argue that μ is a priority-$\boldsymbol{p}$ matching w.r.t. $\mathcal{I}$. To do this, fix a $\boldsymbol{p} \in \mathbb{P}$, and let us suppose that μ is not a priority-$\boldsymbol{p}$ matching, but is a max-weight matching in $\mathbb{G}_{\mathcal{I}}$ with weights $\{w_e\}_{e \in \mathcal{E}}$, where for $e = (a_i, h_j)$ and $r(e) = rank(a_i, h_j)$, $w_e = p_{r(e)} + v_i(h_j)$ (see line 4 in Algorithm 3). Let μ' be a priority-$\boldsymbol{p}$ matching w.r.t. $\mathcal{I}$, which by definition means that it is the max-weight matching in $\mathbb{G}_{\mathcal{I}}$ with weights $\{w'_e\}_{e \in \mathcal{E}}$, where $w'_e = p_{r(e)}$. Also, for a matching $\mu_1 \in \mathcal{M}_{\mathbb{G}_{\mathcal{I}}}$, let $W[\mu_1]$ and $W'[\mu_1]$ denote the sum of the edge weights in μ_1 when using weights $\{w_e\}_{e \in \mathcal{E}}$ and $\{w'_e\}_{e \in \mathcal{E}}$, respectively.

Next, note that for any matching $\mu_1 \in \mathcal{M}_{\mathbb{G}_{\mathcal{I}}}$, $W'[\mu_1] \leq W[\mu_1]$. Additionally, note that, since $p_j \geq n^2$ for all $j \in [n]$, $W'[\mu'] - W'[\mu] \geq n^2$. Finally, observe that, $W[\mu] = \sum_{e \in \mu} w(e) = \sum_{e \in \mu} p_{r(e)} + v_i(h) \leq W'[\mu] + n < W'[\mu'] \leq W[\mu']$, where the first inequality follows from the fact that $v_i(\cdot) \in [0, 1]$ and the second inequality follows since $W'[\mu'] - W'[\mu] \geq n^2$. However, note that $W[\mu] < W[\mu']$ is a contradiction since μ is the max-weight matching in $\mathbb{G}_{\mathcal{I}}$ with weights $\{w_e\}_{e\in\mathcal{E}}$.

Given that μ is a priority-$\boldsymbol{p}$ matching, the fact that it is welfare-optimal follows since we are computing the max-weight matching in $\mathbb{G}_{\mathcal{I}}$ and any priority-$\boldsymbol{p}$ matching $\mu' \in \mathcal{M}_{\mathbb{G}_{\mathcal{I}}}$. □

4.3 Improving Welfare using Threshold Queries

In this section we look at the main question considered in this chapter, which is broadly: *how can one improve social welfare in one-sided matching problems by asking only a small number of queries regarding cardinal utilities.* As mentioned previously, we are interested in binary threshold queries, $\mathcal{Q}(a_i, h_j, t_k)$, which asks an agent a_i if a particular object h_j is of value at least t_k. Towards this end, we begin by considering adaptive algorithms—i.e., algorithms that are allowed to change its queries based on the agents' responses to its previous queries—and show how, when considering each of the four notions (i.e., Pareto optimal, rank-maximal, max-cardinality rank-maximal, and fair matchings) of interest, one can obtain a $(1+\epsilon)$-approximation to the optimal welfare among all matchings that satisfy that notion. Following this, we look at non-adaptive algorithms, which we believe are the more interesting and practical ones for this setting. In particular, we restrict ourselves to algorithms that can ask at most one query per (agent, object) pair and show upper and lower bounds on the approximation achievable. Unless explicitly specified, the results in this section work with respect to both unit-sum and unit-range valuations.

4.3.1 Adaptive algorithm to achieve $(1+\epsilon)$-approximation

The idea in Algorithm 4 is simple. For a specific choice of c, it associates a partition of objects with respect to every agent, where, for $k \in [c]$, an object is in $\mathcal{E}_{ik}$ if agent i's value for the object is within the interval $B_k = [t_k, t_{k-1})$, where $t_k = (\frac{2}{2+\epsilon})^k$, $t_0 = 1$, and the right endpoint interval is closed when $k = 1$. Following this, for every edge $e = (a_i, h_j)$ in $\mathbb{G}_{\mathcal{I}}$, it assigns a weight $w_e = p_r + t_k$, where $r = rank(a_i, h_j)$, $h_j \in \mathcal{E}_{ik}$, and t_k is the left endpoint of the interval B_k, and computes the max-weight matching on the resulting weighted graph. Below we show for $c = \left\lceil \frac{\log(n^2 \cdot 1/\epsilon)}{\log(1+\epsilon/2)} \right\rceil$, this results in an $(1+\epsilon)$-approximation algorithm that uses $O(c \log n)$ queries per agent. In particular, this means that one can achieve a 2-approximation using $O(\log^2 n)$ queries per agent.

Theorem 28. *Given an $\epsilon > 0$, an instance $\mathcal{I} = (\mathcal{N}, \mathcal{H}, \mathcal{P})$, and a priority vector $\boldsymbol{p} = (p_1, \ldots, p_n)$, Algorithm 4 is an adaptive algorithm that asks $O(c \log n)$ queries per agent,*

Input: an $\epsilon > 0$, an instance $\mathcal{I} = (\mathcal{N}, \mathcal{H}, \mathcal{P})$, and a priority vector $\boldsymbol{p} = (p_1, \ldots, p_n)$
Output: returns a PO matching when $p_i = 0$ for all $i \in [n]$ and a priority-$\boldsymbol{p}$ matching when $\boldsymbol{p} \in \mathbb{P}$

1: $\mathbb{G}_{\mathcal{I}} = (\mathcal{N} \cup \mathcal{H}, \mathcal{E}) \leftarrow$ graph induced by $\mathcal{I}$
2: $c \leftarrow \left\lceil \frac{\log(n^2 \cdot 1/\epsilon)}{\log(1+\epsilon/2)} \right\rceil$
3: $t_i \leftarrow (\frac{2}{2+\epsilon})^i$, for $i \in [c]$
4: **for** $a_i \in \mathcal{N}$ **do**
5:     **for** $k \in [c]$ **do**
6:         $\mathcal{E}_{ik} \leftarrow \{(a_i, h_j) \in \mathcal{E} \mid \mathcal{Q}(a_i, h_j, t_k) = 1$ and, if $k \geq 2, \mathcal{Q}(a_i, h_j, t_{k-1}) = 0\}$
7:         **for** $e = (a_i, h_j) \in \mathcal{E}_{ik}$ **do**
8:             $r \leftarrow \text{rank}(a_i, h_j)$
9:             $\text{value}'(e) \leftarrow t_k$
10:             $w_e \leftarrow p_r + \text{value}'(e)$
11:         **end for**
12:     **end for**
13: **end for**
14: $\mu \leftarrow$ max-weight matching in $\mathbb{G}_{\mathcal{I}}$ with weights $\{w_e\}_{e \in \mathcal{E}}$
15: **if** $p_i = 0$ for all $i \in [n]$ **then**
16:     $\mu \leftarrow$ with μ (from line 14) as the initial endowment, run modified top-trading cycles (TTC) algorithm by Saban and Sethuraman [SS13, Algorithm 1, Rule 2] and return the resulting matching
17: **end if**
18: **return** μ

Algorithm 4: returns a PO matching that achieves a $(1+\epsilon)$-approximation to the optimal social welfare among PO matchings or a priority-$\boldsymbol{p}$ matching that achieves a $(1+\epsilon)$-approximation to the optimal social welfare among priority-p matchings.

where $c = \left\lceil \frac{\log(n^2 \cdot 1/\epsilon)}{\log(1+\epsilon/2)} \right\rceil$, *and returns a*

i) Pareto optimal matching μ that achieves a $(1+\epsilon)$-approximation to the optimal welfare among all Pareto optimal matchings when $p_i = 0$ for all $i \in [n]$.

ii) priority-$\boldsymbol{p}$ matching μ that achieves a $(1+\epsilon)$-approximation to the optimal welfare among all priority-$\boldsymbol{p}$ matchings when $\boldsymbol{p} \in \mathbb{P}$.

Proof. We first argue that Algorithm 4 adaptively asks $O(c \log n)$ queries per agent. To see this, note that for each agent a_i and any $k \in [c]$, creating the set $\mathcal{E}_{ik}$ takes $O(\log |A_i|)$ queries. This is so because we know a_i's weak order over A_i, the set of acceptable objects of a_i, and because we are using an adaptive algorithm, we can see the results of one query before we ask the next. Therefore, we can perform binary search to determine the least preferred object h_{j_k} such that $v_i(h_{j_k}) \in [t_k, t_{k-1})$, where $t_i = (\frac{2}{2+\epsilon})^i$, $t_0 = 1$, and the right endpoint interval is closed when $k = 1$ (note that there could multiple such objects since we allow ties, but we can still do this since we have the weak order). Additionally, since, for all $k \in [c]$, $t_k < t_{k-1}$, it is easy to see that we can start from $k = 1$ and proceed to $k = c$ and form the sets $\mathcal{E}_{ik} = \{h \in \mathcal{H} \mid h \succeq h_{j_k} \text{ and, if } k \geq 2, h \prec h_{j_{k-1}}\}$. Therefore, since we create c sets per agent, Algorithm 4 asks $O(c \log n)$ queries per agent.

Next, we need to argue that the matching returned is Pareto optimal when $p_i = 0$ for all $i \in [n]$ and is a priority-$\boldsymbol{p}$ matching when $\boldsymbol{p} \in \mathbb{P}$. To see this, note that in the former case, the matching returned (see line 16 in Algorithm 4) is one that is returned by the modified top-trading cycles (TTC) algorithm by Saban and Sethuraman [SS13, Algorithm 1, Rule 2]. Therefore, we know that it is PO [SS13, Theorem 1]. As for the case when $\boldsymbol{p} \in \mathbb{P}$, note that the matching returned is the max-weight matching in $\mathbb{G}_{\mathcal{I}}$ with weights $\{w_e\}_{e \in \mathcal{E}}$, where $w_e = p_r + \text{value}'(e)$ (see line 10). Now, since for any $e \in \mathcal{E}$, $\text{value}'(e) \in [0, 1]$, we know from Theorem 27 that such a matching is a priority-$\boldsymbol{p}$ matching.

Finally, in order to show that the returned matching achieves a $(1 + \epsilon)$-approximation to optimal welfare among all Pareto optimal/priority-$\boldsymbol{p}$ matchings, let μ^* be the matching that maximizes the total welfare when true edge-weights (which in turn are the true utilities of the agents) are known. That is, μ^* is the max-weight matching in $\mathbb{G}_{\mathcal{I}}$ with weights $(w^*_e)_{e \in \mathcal{E}}$, where, for $e = (a_i, h_j) \in \mathcal{E}$, $w^*_e = p_r + v_i(h_j)$. By Theorem 27 we know that μ^* is the welfare-optimal priority-$\boldsymbol{p}$ matching when $\boldsymbol{p} \in \mathbb{P}$ and it is the welfare optimal Pareto optimal matching when $p_i = 0$ for all $i \in [n]$. So, now, if μ is the matching computed by Algorithm 4, then we need to show that $\text{SW}(\mu^*) \leq (1 + \epsilon) \cdot \text{SW}(\mu)$.

To see this, let us partition the edges in μ^* into sets H and L such that $H = \{(a_i, h_j) \in \mu^* \mid v_i(h_j) \geq \frac{\epsilon}{2n^2}\}$ and $L = \mu^* \setminus H$. Next, note that we have chosen the value of c such that $t_c \geq \frac{\epsilon}{2n^2}$. This in turn implies that for any $k \in [c]$ and $e \in \mathcal{E}_{ik}$, the value assigned to $e = (a_i, h_j) \in H$ in the algorithm (i.e, $\text{value}'(e)$ assigned in line 9 in Algorithm 4) is at least $\frac{t_k}{t_{k-1}} \cdot \text{value}(e)$ (and is at most $\text{value}(e)$). So, using this, we have that $\text{SW}(H) = \sum_{e \in H} \text{value}(e) \leq \frac{2+\epsilon}{2} \sum_{e \in H} \text{value}'(e) \leq \frac{2+\epsilon}{2} \cdot \text{SW}(\mu')$, where μ' is the matching computed in line 14 and the last inequality follows from the fact that $H \subseteq \mathcal{E}$, and μ' is the max-weight matching in $\mathbb{G}_{\mathcal{I}}$ (see line 14).

Finally, in order to get the approximation bound, let us first bound the ratio $\frac{\text{SW}(\mu^*)}{\text{SW}(\mu')}$. Note that since $|L| \leq n$ and for each $e = (a_i, h_j) \in L$, $v_i(h_j) \leq \frac{\epsilon}{2n^2}$, we have,

$$\frac{\mathrm{SW}(\mu^*)}{\mathrm{SW}(\mu')} = \frac{\mathrm{SW}(H) + \mathrm{SW}(L)}{\mathrm{SW}(\mu')} \leq \frac{\frac{2+\epsilon}{2} \cdot \mathrm{SW}(\mu') + n \cdot \frac{\epsilon}{2n^2}}{\mathrm{SW}(\mu')}. \tag{4.2}$$

Next, note that for the case when $\boldsymbol{p} \in \mathbb{P}$, $\mu' = \mu$, and since μ' is a priority-$\boldsymbol{p}$ matching, we have that $\mathrm{SW}(\mu') = \mathrm{SW}(\mu) \geq \frac{1}{n}$, where the inequality follows since in a priority-$\boldsymbol{p}$ matching at least one of the agents is matched to a rank-1 object (see Claim 23 in Appendix C.1 for a proof) and the valuations are either unit-sum or unit-range. Therefore, using this along with (4.2), we have that,

$$\frac{\mathrm{SW}(\mu^*)}{\mathrm{SW}(\mu')} \leq \frac{\frac{2+\epsilon}{2} \cdot \mathrm{SW}(\mu') + n \cdot \frac{\epsilon}{2n^2}}{\mathrm{SW}(\mu')} \leq 1 + \epsilon.$$

For the case when $p_i = 0$ for all $i \in [n]$, note that μ is the matching that is returned by the modified TTC algorithm by Saban and Sethuraman [SS13, Algorithm 1, Rule 2] on initial endowments μ'. Therefore, since this matching is individually rational and PO [SS13, Theorems 1, 2], we have that $\mathrm{SW}(\mu) \geq \mathrm{SW}(\mu')$. Hence, using this along with (4.2), we have that,

$$\frac{\mathrm{SW}(\mu^*)}{\mathrm{SW}(\mu)} \leq \frac{\frac{2+\epsilon}{2} \cdot \mathrm{SW}(\mu') + n \cdot \frac{\epsilon}{2n^2}}{\mathrm{SW}(\mu)} \leq 1 + \epsilon.$$

where the last inequality follows from the fact that $\mathrm{SW}(\mu) \geq \frac{1}{n}$, since in a Pareto optimal matching at least one of the agents is matched to a rank-1 object (see Claim 23 in Appendix C.1 for a proof) and the valuations are either unit-sum or unit-range. □

Remark: As stated previously, an algorithm that produces a PO matching and also achieves an α-approximation to the optimal welfare among all PO matchings has a distortion of α, since a welfare-optimal matching is also PO. This in turn implies that, although the objectives here and in the paper by Amanatidis et al. [Ama+21] are different, the distortion guarantees implied by the algorithm above is similar to the one provided by the λ-ThresholdStepFunction algorithm in their paper (which, though, even works for unrestricted valuations, unlike the case here). Moreover, while the algorithms share some similarities, we use a weaker query model and, most importantly, for an appropriate choice of $\boldsymbol{p}$, our algorithm produces a Pareto optimal or a priority-$\boldsymbol{p}$ matching.

4.3.2 Non-adaptive algorithms: asking one query per (agent, object) pair

In this section we turn our attention to non-adaptive algorithms, in particular looking at algorithms that can only ask one query per (agent, object) pair and cannot change these queries depending on the responses to previous ones. As mentioned in the Introduction, we believe that, at least in some contexts, this is the more interesting and practical setting to consider, since such an algorithm does not have to wait for the agents to respond and also does not require an agent to answer multiple queries with respect to the same object—doing which would in turn entail that the agent is somewhat sure about their cardinal utilities.

Below, we present two algorithms for when agents have unit-sum valuations, first in the context of priority-$\boldsymbol{p}$ matchings and second for Pareto optimal matchings. The latter is an extension of the former and since their proofs are similar, we present only the proof for Pareto optimal matchings in the main body, relegating the one for priority-$\boldsymbol{p}$ matchings to Appendix C.2. Informally, the main idea in the algorithm for Pareto optimal matchings is to first carefully choose a set of values $\{t_k\}_k \in [n]$ and then ask every agent a_i if an object h_j of value at least t_r, where $r = rank(a_i, h_j)$. In particular, we set $t_1 = \frac{1}{n^{1/3}}$ and $t_i = \frac{1}{\min\{i, n^{1/3}\} \cdot n^{2/3}}$, for all $i \in \{2, \ldots, n\}$. Next, we draw a bipartite graph where there is an edge between an agent a_i and object h_j, if the agent answered "Yes" w.r.t. this object; this edge is assigned a weight t_r. Following this, we find the max-weight matching in this graph. Note that not all the agents/objects may have been matched. Therefore, we compute what we refer to as an auxiliary matching, where the auxiliary matching is a matching that maximizes the number of agents who are matched with an edge of rank at most $\lfloor \sqrt[3]{n}/2 \rfloor$ in the case when the max-weight matching in the previous step has no edge of rank-1 (which happens when every agent responds "NO" to the query w.r.t. their rank-1 object), and is the matching that maximizes the number of agents who are matched with a rank-1 edge otherwise. Finally, we combine the max-weight matching computed in the first step with the auxiliary matching (by adding unmatched pairs from the auxiliary one to the max-weight one), arbitrarily match any leftover pairs, and run modified TTC algorithm by Saban and Sethuraman [SS13, Algorithm 1, Rule 2] with the resulting matching as initial endowment. In the theorem below, we show that this achieves an $O(n^{2/3})$-approximation to the optimal social welfare among PO matchings.

Theorem 29. *Given an instance $\mathcal{I} = (\mathcal{N}, \mathcal{H}, \mathcal{P} = (P_1, \ldots, P_n))$, Algorithm 6 asks one non-adaptive query per (agent, object) pair and returns a Pareto optimal matching that achieves an $O(n^{2/3})$-approximation to the optimal welfare among all Pareto optimal matchings for the case when agents have unit-sum valuations.*

Input: an instance $\mathcal{I} = (\mathcal{N}, \mathcal{H}, \mathcal{P} = (P_1, \dots, P_n))$ and priorities $\boldsymbol{p} = (p_1, \dots, p_n)$
Output: a priority-$\boldsymbol{p}$ matching when $\boldsymbol{p} \in \mathbb{P}$

$\mathbb{G}_{\mathcal{I}} = (\mathcal{N} \cup \mathcal{H}, \mathcal{E}) \leftarrow$ graph induced by $\mathcal{I}$
$t_1 \leftarrow \frac{1}{n^{1/3}}$
$t_i \leftarrow \frac{1}{\min\{i, n^{1/3}\} \cdot n^{2/3}}$, for all $i \in \{2, \dots, n\}$
for $e = (a_i, h_j) \in \mathcal{N} \times \mathcal{H}$ **do**
    $r \leftarrow \text{rank}(a_i, h_j)$
    if $\mathcal{Q}(a_i, h_j, t_r)$ **then**
        $w_e \leftarrow p_r + t_r$
    else
        $w_e \leftarrow p_r$
    end if
end for
$\mu \leftarrow$ max-weight matching in $\mathbb{G}_{\mathcal{I}}$, where weights are $(w_e)_{e \in \mathcal{E}}$
return μ

Algorithm 5: returns a priority-$\boldsymbol{p}$ matching that achieves an $O(n^{2/3})$-approximation to the optimal social welfare among priority-p matchings for the case when the agents have unit-sum valuations.

Before we prove this, we introduce the following notations and terminologies which will be useful. Let μ^* denote the matching that achieves optimal welfare among all Pareto optimal matchings when the agents have unit-sum valuations. Let H_i denote the set of agents who are matched to their i-th choice in μ^* and have value at least t_i for their partner in μ^*. Similarly, let L_i denote the set of agents who are matched to their i-th choice in μ^* and have value less than t_i for their partner in μ^*. Additionally, let $H = \cup_{i=1}^{n} H_i$, $L = \cup_{i=1}^{n} L_i$, and, for some $S \subseteq \mathcal{N}$, $\mu_S^* \subseteq \mu^*$ be the set of edges (a_i, h_j) such that $a_i \in S$ and $(a_i, h_j) \in \mu^*$. Now, if $\text{SW}(\mu_S)$ denotes the sum of values of the edges in μ_S (calculated based on the true utilities of the agents), then note that $\text{SW}(\mu^*) = \text{SW}(\mu_H^*) + \text{SW}(\mu_L^*) = \sum_{(a_i,h_j) \in \mu_H^*} v_i(h_j) + \sum_{(a_i,h_j) \in \mu_L^*} v_i(h_j)$.

Next, we prove the following lemma.

Lemma 30. *Let μ_{MM} be the matching that is computed in line 2 in Algorithm 6. Then, $SW(\mu_H^*) \leq n^{2/3} \cdot SW(\mu_{MM})$.*

Proof. Note that $\mu_H^* \subseteq \mathcal{E}$ and for every $e = (a_i, h_j^\ell) \in \mu_H^*$, where h_j^ℓ is a ℓ-th choice of agent a_i, the weight t_ℓ that is assigned to this edge in $\mathbb{G}_{\mathcal{I}}$ is at least $\frac{1}{n^{2/3}} \cdot v_i(h_j^\ell)$ (and is at most $v_i(h_j^\ell)$). This is so because $t_1 = \frac{1}{n^{1/3}}$, $t_i = \frac{1}{\min\{i, n^{1/3}\} \cdot n^{2/3}}$ for $i \geq 2$, and $v_i(h_j^\ell) \leq \frac{1}{\ell}$,

Input: an instance $\mathcal{I} = (\mathcal{N}, \mathcal{H}, \mathcal{P} = (P_1, \ldots, P_n))$
Output: a Pareto optimal matching
1: $\mu'_{MM} \leftarrow$ matching returned by Algorithm 5 on $\mathcal{I}$ and $\boldsymbol{p} = (0, \ldots, 0)$
2: $\mu_{MM} \leftarrow \mu'_{MM} \setminus \{e \in \mu_{MM} \mid w_e = 0\}$ ▷ remove edges with weight 0 from μ'_{MM}
3: $\mu^1_{MM} = \{(a_i, h_j) \mid (a_i, h_j) \in \mu_{MM}$ and $rank(a_i, h_j) = 1\}$
4: **if** $|\mu^1_{MM}| == 0$ **then**
5: $\quad \mu'_{aux} \leftarrow$ matching in $\mathbb{G}_{\mathcal{I}}$ that maximizes the number of agents who are matched with an edge of rank at most $\lfloor \sqrt[3]{n}/2 \rfloor$
6: **else**
7: $\quad \mu'_{aux} \leftarrow$ matching in $\mathbb{G}_{\mathcal{I}}$ where as many agents as possible to a rank-1 edge
8: **end if**
9: $\mu_{aux} \leftarrow \mu'_{aux} \setminus \{(a, o) \mid (a, o) \in \mu'_{aux}$ and either a or o is matched in $\mu_{MM}\}$
10: $\mu_{\text{rest}} \leftarrow$ arbitrarily match the acceptable (agent, object) pairs that are not matched in $\mu_{MM} \cup \mu_{aux}$
11: $\mu \leftarrow$ with $\mu_{MM} \cup \mu_{aux} \cup \mu_{\text{rest}}$ as the initial endowment, run modified top-trading cycles (TTC) algorithm by Saban and Sethuraman [SS13, Algorithm 1, Rule 2] and return the resulting matching.
12: **return** μ

Algorithm 6: returns a PO matching achieves an $O(n^{2/3})$-approximation to the optimal social welfare among PO matchings for the case when the agents have unit-sum valuations.

since the valuations are unit-sum and h^ℓ_j is in the ℓ-th choice of agent i. Combining these two observations, we have that $\mathrm{SW}(\mu^*_H) = \sum_{(a_i,h_j)\in\mu^*_H} v_i(h_j) \leq n^{2/3} \cdot \mathrm{SW}(\mu_{MM})$, where the last inequality follows by using the fact that μ_{MM} (computed in line 2) is the max-weight matching on the graph $\mathbb{G}_{\mathcal{I}}$, and as discussed above $\mu^*_H \subseteq \mathcal{E}$ and the edge weights in $\mathbb{G}_{\mathcal{I}}$ are off by a factor of at most $n^{2/3}$. □

Equipped with this lemma, we can now prove our theorem.

Proof of Theorem 29. First, note that Algorithm 6 returns a matching μ that is Pareto optimal since it is the matching returned by running modified TTC algorithm by Saban and Sethuraman [SS13, Algorithm 1, Rule 2] on some initial endowments [SS13, Theorem 1]. Next, let μ_{MM} be the matching computed in line 2 in Algorithm 6, and let $\mu^1_{MM} \subseteq \mu_{MM}$ be set of edges of rank-1 in μ_{MM}. We will proceed by considering the following two cases and show that in each case $\frac{\mathrm{SW}(\mu^*)}{\mathrm{SW}(\mu)} \in O(n^{2/3})$.

Case 1: $|\mu^1_{MM}| \geq 1$. For this case, let μ'_{aux} be the matching in $\mathbb{G}_{\mathcal{I}}$ where as many agents as possible to a rank-1 edge (see line 7 in Algorithm 6) and let $\mu_{aux} = \mu'_{aux} \setminus \{(a,o) \mid (a,o) \in \mu'_{aux}$ and either a or o is matched in $\mu_{MM}\}$. Below, we will argue that $\text{SW}(\mu_{MM}) + \text{SW}(\mu_{aux}) \geq \max\left\{\frac{|X'_1|}{2n}, t_1\right\}$, where X'_i is the set of edges of rank-i in μ'_{aux}.

Claim 4. If $|\mu^1_{MM}| \geq 1$, then $\text{SW}(\mu_{MM}) + \text{SW}(\mu_{aux}) \geq \max\left\{\frac{|X'_1|}{2n}, t_1\right\}$.

Proof. Consider the set of edges in X'_1, and let X_{aux} be the edges in X'_1 that belong to μ_{aux} (i.e., $X_{aux} = X'_1 \cap \mu_{aux}$). By our definition of μ_{aux}, for each $(a_i, h_j) \in X'_1 \setminus X_{aux}$, at least one of a_i or h_j is matched in μ_{MM}. Therefore,

$$\begin{aligned}
\text{SW}(\mu_{MM}) + \text{SW}(\mu_{aux}) &\geq \frac{|X'_1 \setminus X_{aux}|}{2} \cdot t_n + \text{SW}(X_{aux}) \\
&\geq \frac{|X'_1 \setminus X_{aux}|}{2} \cdot t_n + |X_{aux}| \cdot \frac{1}{n} \\
&\geq \frac{|X'_1|}{2} \cdot \frac{1}{n}.
\end{aligned}$$

In the set of inequalities above, the first inequality follows from the fact that there are at least $\frac{|X'_1 \setminus X_{aux}|}{2}$ unique edges in μ_{MM} and each of them have weight at least t_n; the second inequality follows because each edge in X_{aux} is of value at least $\frac{1}{n}$ (since $X_{aux} = X'_1 \cap \mu_{aux}$, the valuations are unit-sum, and the agents are matched to their first choice in μ_{aux} and so have value at least $\frac{1}{n}$).

Additionally, note that since $|\mu^1_{MM}| \geq 1$, $\text{SW}(\mu_{MM}) \geq t_1$. Hence, combining the two observations above, we have that $\text{SW}(\mu_{MM}) + \text{SW}(\mu_{aux}) \geq \max\{\frac{|X'_1|}{2n}, t_1\}$. ☐

Equipped with the claim, next, consider μ^*_L and note that,

$$\begin{aligned}
\text{SW}(\mu^*_L) &= \text{SW}(\mu^*_{L_1}) + \sum_{i=2}^{n} \text{SW}(\mu^*_{L_i}) \\
&\leq |L_1| \cdot t_1 + (n - |L_1|) \cdot t_2 \\
&\leq |X'_1| \cdot t_1 + n \cdot t_2, \qquad (4.3)
\end{aligned}$$

where the first inequality uses the facts that, by the definition of L_i, every agent in $\mu^*_{L_i}$ has value less than t_i for their partner and that $t_i \geq t_j$ for $i \leq j$ and $i, j \in [n]$, and the final inequality follows since μ'_{aux} is a matching in $\mathbb{G}_{\mathcal{I}}$ where as many agents as possible is matched to a rank-1 edge.

Given the above, if $\mu' = \mu_{MM} \cup \mu_{aux} \cup \mu_{rest}$, then we have that,

$$\begin{aligned}
\frac{\mathrm{SW}(\mu^*)}{\mathrm{SW}(\mu')} &= \frac{\mathrm{SW}(\mu^*_H) + \mathrm{SW}(\mu^*_L)}{\mathrm{SW}(\mu_{MM}) + \mathrm{SW}(\mu_{aux}) + \mathrm{SW}(\mu_{rest})} \\
&\leq \frac{n^{2/3} \cdot \mathrm{SW}(\mu_{MM}) + |X'_1| \cdot t_1 + n \cdot t_2}{\mathrm{SW}(\mu_{MM}) + \mathrm{SW}(\mu_{aux})} \\
&\leq \frac{n^{2/3} \cdot \mathrm{SW}(\mu_{MM})}{\mathrm{SW}(\mu_{MM})} + \frac{|X'_1| \cdot t_1}{|X'_1|/(2n)} + \frac{n \cdot t_2}{t_1} \\
&\leq n^{2/3} + 2n^{2/3} + n^{2/3}. \qquad (4.4)
\end{aligned}$$

In the set of inequalities above, the first one follows from using Lemma 30 and (4.3); the second inequality follows from Claim 4; the last inequality follows from our choice of t_1 and t_2.

Case 2: $|\mu^1_{MM}| = 0$**.** For this case, let $k = \lfloor \sqrt[3]{n}/2 \rfloor$ and μ'_{aux} be the matching in $\mathbb{G}_{\mathcal{I}}$ where as many agents as possible to an edge of rank at most k (see line 5 in Algorithm 6) and let $\mu_{aux} = \mu'_{aux} \setminus \{(a, o) \mid (a, o) \in \mu'_{aux}$ and either a or o is matched in $\mu_{MM}\}$. Below, we will argue that $\mathrm{SW}(\mu_{MM}) + \mathrm{SW}(\mu_{aux}) \geq \frac{|X'|}{2n}$, where $X' = \cup_{i=1}^k X'_i$ and X'_i is the set of edges of rank-i in μ'_{aux}.

Claim 5. If $|\mu^1_{MM}| = 0$, then $\mathrm{SW}(\mu_{MM}) + \mathrm{SW}(\mu_{aux}) \geq \frac{|X'|}{2n}$.

Proof. Since $|\mu^1_{MM}| = 0$, every agent values their first choice object at a value less than t_1 (since otherwise $|\mu^1_{MM}| \geq 1$). This implies that, since their valuations are unit-sum, for $j \in [k]$, their value for a rank-j object is at least $\frac{1}{2n}$ (see Claim 21 in Appendix C.1 for a proof). Next, like in the proof of Claim 4, consider $X_{aux} = \mu_{aux} \cap X'$. Since for every edge $(a_i, h_j) \in X' \setminus X_{aux}$, at least one of a_i or h_j is matched in μ_{MM}, and every edge in μ_{MM} has weight at least t_n, we have,

$$\begin{aligned}
\mathrm{SW}(\mu_{MM}) + \mathrm{SW}(\mu_{aux}) &\geq \frac{|X' \setminus X_{aux}|}{2} \cdot t_n + \mathrm{SW}(X_{aux}) \\
&\geq \frac{|X' \setminus X_{aux}|}{2} \cdot t_n + |X_{aux}| \cdot \frac{1}{2n} \\
&\geq \frac{|X'|}{2n}.
\end{aligned}$$

□

Given the claim, next, note that,

$$\mathrm{SW}(\mu^*_L) = \sum_{i=1}^{k} \mathrm{SW}(\mu^*_{L_i}) + \sum_{i=k+1}^{n} \mathrm{SW}(\mu^*_{L_i})$$

$$\leq \left(\sum_{i=1}^{k} |L_i|\right) \cdot t_1 + \left(\sum_{i=k+1}^{n} |L_i|\right) \cdot t_{k+1}$$
$$\leq \left(\sum_{i=1}^{k} |X'_i|\right) \cdot t_1 + \left(\sum_{i=k+1}^{n} |L_i|\right) \cdot t_{k+1}$$
$$= |X'| \cdot t_1 + \left(\sum_{i=k+1}^{n} |L_i|\right) \cdot t_{k+1}. \tag{4.5}$$

Equipped with the above, if $\mu' = \mu_{MM} \cup \mu_{aux} \cup \mu_{rest}$, then for this case we have that,

$$\begin{aligned}
\frac{\mathrm{SW}(\mu^*)}{\mathrm{SW}(\mu')} &= \frac{\mathrm{SW}(\mu^*_H) + \mathrm{SW}(\mu^*_L)}{\mathrm{SW}(\mu_{MM}) + \mathrm{SW}(\mu_{aux}) + \mathrm{SW}(\mu_{rest})} \\
&\leq \frac{n^{2/3} \cdot \mathrm{SW}(\mu_{MM}) + |X'| \cdot t_1 + \left(\sum_{i=k+1}^{n} |L_i|\right) \cdot t_{k+1}}{\mathrm{SW}(\mu_{MM}) + \mathrm{SW}(\mu_{aux})} \\
&\leq \frac{n^{2/3} \cdot \mathrm{SW}(\mu_{MM})}{\mathrm{SW}(\mu_{MM})} + \frac{|X'| \cdot t_1}{|X'|/(2n)} + \frac{\left(\sum_{i=k+1}^{n} |L_i|\right) \cdot t_{k+1}}{|X'|/(2n)} \\
&\leq n^{2/3} + 2n^{2/3} + 8n^{2/3}.
\end{aligned} \tag{4.6}$$

In the set of inequalities above, the first one follows from using Lemma 30 and (4.5); the second inequality follows Claim 5; the last inequality follows since $|X'| \geq \min\left\{k, \sum_{i=k+1}^{n} |L_i|\right\}$ (see Claim 24 in Appendix C.1 for a proof).

Finally, combining (4.4) and (4.6), and using the fact that the modified TTC algorithm by Saban and Sethuraman [SS13, Algorithm 1, Rule 2] is individually rational [SS13, Theorem 2] (which in turn implies $SW(\mu) \geq SW(\mu')$), gives us our theorem. □

Next, we state the following result for priority-$\boldsymbol{p}$ matchings, whose proof, as mentioned previously, is similar the proof above and hence appears in Appendix C.2.1.

Theorem 31. *Given an instance $\mathcal{I} = (\mathcal{N}, \mathcal{H}, \mathcal{P} = (P_1, \ldots, P_n))$ and a vector of priorities $\boldsymbol{p} = (p_1, \ldots, p_n)$, where $\boldsymbol{p} \in \mathbb{P}$, Algorithm 5 asks one non-adaptive query per (agent, object) and returns a priority-$\boldsymbol{p}$ matching that achieves an $O(n^{2/3})$-approximation to the optimal welfare among all priority-$\boldsymbol{p}$ matchings for the case when agents have unit-sum valuations.*

Finally, we also consider the case when agents have unit-range valuations and show how one can obtain an $O(\sqrt{n})$-approximation to the optimal social welfare among Pareto optimal and priority-$\boldsymbol{p}$ matchings. Since the algorithms and analyses for these are somewhat similar to the ones above, we present these results in Appendix C.3.2.1.

4.4 Lower Bounds

Here we turn our attention to lower bounds for the case when an algorithm can ask at most one query per (agent, object) pair—i.e., for the setting considered in Section 4.3.2. We show that, for the unit-sum and unit-range cases, any deterministic algorithm $\mathcal{A}$ that asks at most one query per (agent, object) pair and produces a Pareto-optimal/rank-maximal/max-cardinality rank-maximal/fair matching has a worst-case welfare loss of $\Omega(\sqrt{n})$, i.e., $\mathcal{L}(\mathcal{A}) \in \Omega(\sqrt{n})$.

Theorem 32. *Let X denote one of the properties in the set {Pareto-optimal, rank-maximal, max-cardinality rank-maximal, and fair}. Let $\mathcal{A}$ be a non-adaptive deterministic algorithm that always produces a matching that satisfies property X and asks at most one query per (agent, object) pair. If there are n agents with unit-sum valuation functions, then $\mathcal{L}(\mathcal{A}) \in \Omega(\sqrt{n})$.*

Proof. Let $n \geq 18$ and $n = 5k + r$, where $k = \lfloor \frac{n}{5} \rfloor$ and $0 \leq r \leq 4$. Next, let us construct an instance $\mathcal{I}$ with the set of agents $\mathcal{N}$, where $|\mathcal{N}| = n$ and the set of object $\mathcal{H}$, where $|\mathcal{H}| = n$. We partition the set of agents into sets $B_1, \ldots, B_{k+1}$ such that $|B_{k+1}| = r$ and $|B_i| = 5$ for all $i \in [k]$. We will refer to each B_i, where $i \in [k]$, as a *block*. Given this, the preferences of the agents are as defined below, where $h_j \succ \mathcal{H} \setminus \{h_j\}$ implies that all the objects in the set $\mathcal{H} \setminus \{h_j\}$ are less preferred than h_j and are preferred in some arbitrary linear order (which is the same for all the agents).

$$\begin{aligned} \forall i \in [k],\ \text{agents in } B_i \text{ have the following preferences} &: h_1 \succ h_{i+1} \succ \mathcal{H} \setminus \{h_1, h_{i+1}\} \\ \text{agents in } B_{k+1} \text{ have the following preferences} &: h_1 \succ h_{n-1} \succ \mathcal{H} \setminus \{h_1, h_{n-1}\} \end{aligned}$$

Next, since $\mathcal{A}$ is a non-adaptive deterministic algorithm that asks at most one query per (agent, object) pair, we can think of $\mathcal{A}$ to consist of two components—an outcome function f which for an instance $\mathcal{I} \in \mathbb{I}$ outputs a matching in $\mathcal{M}_{\mathbb{G}_{\mathcal{I}}}$, and a $|\mathcal{N}| \times |\mathcal{H}|$ matrix T, where $T_{ij} \in [0, 1]$ represents the threshold asked to agent a_i w.r.t. the object h such that $rank(a_i, h) = j$ (i.e., if $\mathcal{A}$ asks the query $\mathcal{Q}(a_i, h_i^j, t)$, where $rank(a_i, h_i^j) = j$, then $T_{ij} = t$). Note that, for all $i \in [n]$, depending on the values of T_{i1} and T_{i2}, we can classify a_i as belonging to one of the following four types.

$$\text{Type-1} : T_{i1} \in [0, \frac{1}{2}),\ T_{i2} \in [0, \frac{1}{\sqrt{n}}] \qquad \text{Type-2} : T_{i1} \in [\frac{1}{2}, 1],\ T_{i2} \in [0, \frac{1}{\sqrt{n}}]$$

$$\text{Type-3} : T_{i1} \in [0, \frac{1}{\sqrt{n}}),\ T_{i2} \in (\frac{1}{\sqrt{n}}, 1] \qquad \text{Type-4} : T_{i1} \in [\frac{1}{\sqrt{n}}, 1],\ T_{i2} \in (\frac{1}{\sqrt{n}}, 1]$$

Also, note that, for $i \in [k]$, since each block B_i has 5 agents, at least two of them will be classified as the same type (where types are as defined above). For each block B_i and $j \in [4]$, we say that B_i is a *Type-j block* if there are at least two agents such that both are of Type-j and for every $1 \leq k < j$, there is at most one agent of Type-k. We use n_j to denote the number of Type-j blocks, and hence have that $\sum_{j \in [4]} n_j = k$. Additionally, for a Type-j block B_i, we let a_{2i-1} and a_{2i} to denote the two agents who are of Type-j in B_i (if there are more than two such agents, consider any two of them and denote them as a_{2i-1} and a_{2i}) and say that a_{2i-1} and a_{2i} are the *special agents* in B_i.

As a final step before proving the theorem, we need to define a set of valuation profiles that are consistent with the preferences mentioned above. Our objective will be to show that there is at least one valuation profile in this set which achieves the desired bound on $\mathcal{L}(\mathcal{A})$. So, to do this, let us first define the following unit-sum utility functions,[4]

$$u_0 = (c_1^2 + \epsilon/2, c_1^2, c_1^2 - c_2\epsilon, \dots, c_1^2 - c_2\epsilon)$$

$$u_1 = (1 - c_1, c_1, 0, \dots, 0) \qquad u_2 = (1/2 + \epsilon, 1/2 - \epsilon, 0, \dots, 0)$$

$$u_3 = (1/2 - c_1 - \epsilon, c_1 + \epsilon, c_2, \dots, c_2) \qquad u_4 = (1/4 + \epsilon, 1/4 - \epsilon, c_2, \dots, c_2)$$

$$u_5 = (1 - c_1^2, c_1^2, 0 \dots, 0) \qquad u_6 = (1 - c_1 + \epsilon, c_1 - \epsilon, 0, \dots, 0)$$

$$u_7 = (c_1 - c_1^2, c_1^2, c_3, \dots, c_3) \qquad u_8 = (3c_1/4, c_1/4, c_3, \dots, c_3),$$

where $u_j = (x_1, x_2, x_3 \dots, x_3)$ implies that they value their most preferred object at x_1, second most preferred object at x_2, and every other object at x_3, $\epsilon > 0$ is a very small real number ($\epsilon < 1/n^4$ will suffice), $c_1 = \frac{1}{\sqrt{n}}$, $c_2 = \frac{1}{2(n-2)}$, and $c_3 = \frac{1-c_1}{n-2}$. Given this, consider the following set of valuation profiles $\mathcal{V}_{\mathcal{I}}$, all of which are consistent with the preferences in $\mathcal{I}$ and where each of them is defined the following way.

i) for $i \in [k]$, $j \in [4]$, and special agents a_{2i-1} and a_{2i} in B_i, where B_i is a Type-j block, let one of the agents have utility function u_{2j-1} and the other have u_{2j}.

ii) let the agents in $\mathcal{N} \setminus \{a_1, \dots, a_{2k}\}$ have utility function u_0.

Note that the only difference between valuation profiles in $\mathcal{V}_{\mathcal{I}}$ is w.r.t. the utility functions of the special agents. Additionally, note that for each $v \in \mathcal{V}_{\mathcal{I}}$ and for each Type-j block B_i, the utility functions, u_{2j-1} and u_{2j}, of the special agents have been defined in such a way that they will respond identically to the queries in T. This can be seen by considering Table 4.3 which shows how for all $j \in [4]$, the special agents of a Type-j block will respond

[4] Note that in each of these valuations, objects 3 to n have been given the same value. This is purely for ease of exposition. To ensure that no two objects have the same value, we can just perturb the values slightly.

	Response to $\mathcal{Q}(\cdot, \cdot, T_{i1})$	Response to $\mathcal{Q}(\cdot, \cdot, T_{i2})$
special agents in a Type-1 block (assigned u_1 and u_2)	Yes	Yes
special agents in a Type-2 block (assigned u_3 and u_4)	No	Yes
special agents in a Type-3 block (assigned u_5 and u_6)	Yes	No
special agents in a Type-4 block (assigned u_7 and u_8)	No	No

Table 4.3: Responses of the special agents in a Type-j block to queries $\mathcal{Q}(\cdot, \cdot, T_{i1})$ or $\mathcal{Q}(\cdot, \cdot, T_{i2})$

to queries $\mathcal{Q}(\cdot, \cdot, T_{i1})$ or $\mathcal{Q}(\cdot, \cdot, T_{i2})$, and by observing that utility functions u_{2j-1} and u_{2j} have the same values for all objects except the two most preferred ones.

Now, equipped with all of the above, let us first argue about the welfare-optimal matching that satisfies property X. To do this, consider any $v \in \mathcal{V}_{\mathcal{I}}$ and consider the matching μ^* of size n where in each Type-j block B_i, the special agent with utility function u_{2j} gets h_{i+1} (so they all get their second most preferred object), if $|B_{k+1}| > 0$, then one of the agents in B_{k+1} gets h_{n-1}, and the rest of the objects are allocated arbitrarily. From the way the preferences of the agents are defined, it is easy to verify that such a matching satisfies property X. Also, if $Z = n - 2\sum_{j\in[4]} n_j$, then from the definition of utility functions u_{2j} for all $j \in [4]$, and since the utility of matching h_1 and all other objects to some agent is at least $(c_1^2 + \epsilon/2)$ and $(c_1^2 - c_2\epsilon)$, respectively, we have that,

$$\begin{aligned}
\mathrm{SW}(\mu^*) &\geq (c_1^2 + \epsilon/2) + (1/2 - \epsilon)n_1 + (1/4 - \epsilon)n_2 + (c_1 - \epsilon)n_3 + \frac{c_1}{4}n_4 + (c_1^2 - c_2\epsilon)Z \qquad (4.7)\\
&\geq (c_1^2 + \epsilon/2) + (n_1 + n_2 + n_3 + n_4)\frac{c_1}{4} + (c_1^2 - c_2\epsilon)Z\\
&\geq \frac{\sqrt{n}}{28}, \qquad (4.8)
\end{aligned}$$

where the last inequality follows since $n = 5k + r$, where $0 \leq r \leq 4$, implies $k = \sum_{j\in[4]} n_j = \lfloor n/5 \rfloor$, and $n \geq 18$.

Finally, consider $\mathcal{A}$ and let μ be the matching returned by $\mathcal{A}$ for the instance $\mathcal{I}$ when agents are asked queries in T. Next, we adversarially pick a $v \in \mathcal{V}_{\mathcal{I}}$ the following way:

i) for $i \in [k]$, if B_i is a Type-j block for some $j \in [4]$ and if $\mu(a_{2i-1}) = h_{i+1}$, then consider valuation profile in $\mathcal{V}_\mathcal{I}$ where a_{2i-1} has utility function u_{2j-1}

ii) for $i \in [k]$, if B_i is a Type-j block for some $j \in [4]$ and if $\mu(a_{2i}) = h_{i+1}$, then consider valuation profile in $\mathcal{V}_\mathcal{I}$ where a_{2i} has utility function u_{2j-1}

Given such a $v \in \mathcal{V}_\mathcal{I}$, we can now calculate $\mathrm{SW}(\mu)$. And for this, observe that from the definition of utility functions u_{2j-1} for $j \in [4]$, and since the utility of matching h_1 and all other objects to some agent is at most 1 and c_1^2, respectively, we have that,

$$\begin{aligned}
\mathrm{SW}(\mu) &\le 1 + c_1 n_1 + (c_1+\epsilon)n_2 + c_1^2 n_3 + c_1^2 n_4 + c_1^2 Z \\
&\le 2 + \frac{8}{\sqrt{n}}\left(\frac{1}{2}n_1 + (1/4-\epsilon)n_2 + \frac{c_1}{4}n_3 + (c_1-\epsilon)n_4\right) \quad \text{(since } Z \le n\text{)} \\
&\le \frac{56}{\sqrt{n}}\mathrm{SW}(\mu^*) + \frac{8}{\sqrt{n}}\mathrm{SW}(\mu^*). \quad \text{(using (4.7) and (4.8))} \qquad (4.9)
\end{aligned}$$

Rearranging (4.9) we have that $\frac{\mathrm{SW}(\mu^*)}{\mathrm{SW}(\mu)} \ge \frac{\sqrt{n}}{64}$, or in other words that, $\mathcal{L}(\mathcal{A}) \in \Omega(\sqrt{n})$. □

Finally, we also consider the unit-range case and show that any deterministic algorithm that asks at most one query per (agent, object) pair and produces a Pareto-optimal/rank-maximal/max-cardinality rank-maximal/fair matching has $\mathcal{L}(\mathcal{A}) \in \Omega(\sqrt{n})$. The proof of this appears in C.3.2.2 and is almost identical to the proof of Theorem 32, with the main difference being in the way utility functions $u_0, u_1, \ldots, u_8$ in the proof are defined.

4.5 Discussion

The focus of this chapter was on one-sided matching problems. While the usual assumption in such problems is that agents only submit ordinal preferences, it is not hard to imagine scenarios where agents might have some cardinal preferences which, for instance, indicate that they like object h_1 much more than object h_2. Although ignoring this information can lead to a lose in welfare, asking the agents for their cardinal utilities is not ideal, since determining their exact utilities can be a cognitively-involved task. Therefore, in this chapter we investigated the benefit of eliciting a small amount of extra information about agents' cardinal utilities. In particular, we designed algorithms that used simple threshold queries and returned a matching satisfying some desirable matching property, while also achieving a good approximation to the optimal welfare among all matchings satisfying that property. Overall, our results show how even asking agents for just one bit of extra information per object can improve welfare.

There are a number of future research directions that this work can take. First, the model in this chapter assumes that each agent needs to be matched to at most one object and that each object can be matched to at most one agent. However, there are several situations where more than one agent can be matched to the same object, like when assigning students to courses or schools. While our results do not directly hold when each object h_j has a capacity constraint c_j, only minimal modifications are needed. In particular, every time we construct a graph in any of the algorithms, all that needs to be done is to create c_j copies for the node that corresponds to object h_j. Other open algorithmic problems include addressing the gap between the upper and lower bounds for the non-adaptive algorithms, expanding the set of properties of interest to include, for example, popular matchings [Abr+07], or asking similar questions in the context of two-sided matching problems.

More broadly, a particularly interesting direction is to better understand the implications of deploying such an approach in practice. As mentioned in the introduction, we believe that in many settings non-adaptive algorithms that only ask the agents for a few number of queries with respect to an object might be the most practical approach to pursue, since it involves minimal communication overhead. Moreover, deploying something like that seems easier since the only change that needs to be made to the existing system which asks for ordinal preferences is to add checkboxes with respect to an object and the corresponding threshold queries. Nevertheless, there are still challenges to make this truly useful. A careful reader would have noticed that the thresholds used in our algorithms are very specific values (like $1/n^{1/3}$), which may not be easy to answer. While one potential way to mitigate some of this difficulty is by multiplying all the threshold values by a large enough constant so as to make them easier to comprehend, it's not clear if that would be enough. Therefore, it might be useful to have studies to better understand the kinds of queries that are easier to answer and the types of interface-design that can best support queries, as well as better understand what matching properties are deemed to be most important by users and designers of systems.

Part II

Making Decisions with Inaccurate Information

Chapter 5

Algorithmic Stability in Fair Allocation of Indivisible Goods

5.1 Introduction

Until now all the chapters in this thesis dealt with situations where agents submit incomplete preferences. However, there are many scenarios where it is assumed, and in fact expected, that the agents provide complete preferences. For example, the assumption that agents have complete cardinal preferences is crucial in the rent-division setting [Gal+17] and is typical in the vast literature on fair allocation of both divisible [BT96b] and indivisible goods [Lip+04]. While one could argue that assuming cardinal preferences is reasonable in such contexts, there is still the question on how to deal with agents who may not be completely aware of their preferences. In this chapter we deal with this broad question. Informally, we look at how to account for agents who, when asked to provide cardinal preferences, and owing to the fact that they are not precisely aware of their preferences, may report their preferences inaccurately.

We study this question in the context of fair allocation of indivisible goods where there are m indivisible goods that need to be fairly (for some notion of fairness) divided amongst $n \geq 2$ agents. Each agent is assumed to have a valuation function which assigns a value for each good and typically it is assumed that the valuation functions are *additive*—meaning that the value of a set of goods is the sum of values of the individual goods in the set. A popular algorithm for this problem is the maximum Nash welfare (MNW) solution which computes an allocation that is Pareto optimal (PO) and satisfies a fairness notion called

	A	B
g_1	104	162
g_2	273	250
g_3	186	240
g_4	437	348

(5.1a) Original instance

	A	B
g_1	105	162
g_2	271	250
g_3	186	240
g_4	438	348

(5.1b) Instance where agent A makes minor mistakes

EF1 [Car+16]. Informally, in an allocation that satisfies EF1 an agent i does not envy agent j after she removes some good from j's bundle, whereas Pareto optimality of an allocation implies that there is no other allocation where every agent receives at least as much utility and at least one of the agents strictly more. Given our current state of knowledge on fair and efficient allocations, the MNW solution essentially provides the best-known guarantees. However, as we will soon see, there is at least one aspect with respect to which it is lacking. The issue we will discuss is not specific to MNW but is something that can be raised with respect to different algorithms that assume access to cardinal preferences in different settings. Nevertheless, we use MNW here since our motivation to look at this issue stemmed from observing examples on Spliddit (`www.spliddit.org`), a popular fair division website which uses the MNW solution [GP15].

To illustrate our concern, consider an example with two agents (A and B) and four goods $(g_1, \ldots, g_4)$. As mentioned above, the agents have additive valuations and their values for the goods are in Table 5.1a. Spliddit uses the MNW solution to compute an EF1 and PO allocation for this instance, and it returns one where agent A receives $\{g_2, g_4\}$ and agent B receives $\{g_1, g_3\}$. What if, however, agent A made a minor 'mistake' while reporting the values and instead reported the values in Table 5.1b? Note that the two valuations are almost identical, with the value of each good being off by at most 2. Therefore, intuitively, it looks like we would, ideally, like to have similar outputs—and more so since the allocation for the original instance satisfies EF1 and PO even with respect to this new instance. However, does the MNW solution do this? No, and in fact the allocation returned in this case is one where agent A gets g_4 alone and agent B gets the rest. This in turn implies that agent A is losing roughly 38% of their utility for the 'mistake', which seems highly undesirable.

The example described above is certainly not one-off, and in fact we will show later how there are far worse examples for different fair division algorithms. More broadly, we believe that this is an issue that can arise in many problems where the inputs are assumed to be cardinal values. After all, it is not hard to imagine scenarios where many of us may find it hard to convert our preferences to precise numerical values. Now, of course, it is easy to see that if we insist on resolving this issue completely—meaning, if we insist that the agent making the 'mistake' should not experience any change in their outcome—then it cannot be done in any interesting way as long as we allow the 'mistakes' to be arbitrary and also insist that the algorithm be deterministic.[1] However, it is possible to impose some structure on the 'mistakes' made by agents. For example, in many settings, it might be reasonable to assume that agents are able to, at a minimum, maintain the underlying ordinal structure of their preferences. That is, if an agent considers good g to be the r-th highest valued good according to their true preference, then this information is also maintained in the 'mistake'. Note that this is indeed the case in the example above, and so, more broadly, this is the setting we consider. Our goal in this chapter is to try and address the issue that we observe in the example, and intuitively we want to design algorithms where an agent does not experience a large change in their utility as long as their report is only off by a little.

Before we make this more concrete, a reader who is familiar with the algorithmic game theory (AGT) literature might have the following question: "Why not just consider ordinal algorithms?" After all, these algorithms will have the property mentioned above when the underlying ordinal information is maintained, and moreover there is a body of literature that focuses on designing algorithms that only use ordinal information and still provide good guarantees with respect to the underlying cardinal values (e.g., see [Bou+15; AS16; AZ17; GKM17; AA18]). Additionally, and more specifically in the context of fair allocations, there is also a line of work that considers ordinal algorithms [BEL10; BKK14; Azi+15; SAH17]. While this is certainly a reasonable approach, there are a few reasons why this is inadequate: *i)* Constraining algorithms to use only ordinal information might be too restrictive. In fact, this is indeed the case here since we show that there are no ordinal algorithms that are EF1 and even approximately PO. Additionally, assuming that the agents only have ordinal preferences might be too pessimistic in certain situations. *ii)* There are systems like Spliddit that are used in practice and which explicitly elicit cardinal preferences, and so we believe that the approach here will be useful in such settings.

Given this, we believe that there is need for a new notion to address this issue. We term this *stability*, and informally our notion of stability captures the idea that the utility experienced by an agent should not change much as long as they make "small" or "innocuous" mistakes when reporting their preferences.[2] Although the general idea of algorithmic stability is certainly not new (see Section 5.3 for a discussion), to best of our knowledge, the notion

[1] Although there is work on randomized fair allocations (e.g., [BM01; Bud+13]), as pointed out by Caragiannis et al. [Car+16], randomization is not appropriate for many practical fair division settings where the outcomes are just used once.

of stability we introduce here (formally defined in Section 5.2.1) has not been previously considered. Therefore, we introduce this notion in the context of problems where cardinal preferences are elicited, and explicitly advocate for it to be considered during algorithm or mechanism design. This in turn constitutes what we consider as the main contribution of this chapter. We believe that if the algorithms for fair division—and in fact any problem where cardinal preferences are elicited—are to be truly useful in practice they need to have some guarantees on stability, and so towards this end we consider the problem of designing stable algorithms in the context of fair allocation of indivisible goods among two agents.

The rest of this chapter is organized as follows. We begin by formally defining the notion of stability and show how one cannot hope for stable algorithms that are EF1 and even approximately PO. As a result, we propose two relaxations, namely, approximate-stability and weak-approximate-stability, and show how existing algorithms that are fair and efficient perform poorly even in terms of these relaxations. This implies that one has to design new algorithms, and towards this end we present a simple, albeit exponential, algorithm for two agents that is approximately-stable and that guarantees pairwise maximin share (PMMS) and PO allocation; the algorithm is based on a general characterization result for PMMS allocations which we believe might be of independent interest. Finally, we show how a small change to the existing two-agent fair division algorithms can get us weak-approximate-stability along with the properties that these algorithms otherwise satisfy.

5.2 Preliminaries

Let $[n] = \{1, \ldots, n\}$ denote the set of agents and $\mathcal{G} = \{g_1, \ldots, g_m\}$ denote the set of indivisible goods that needs to divided among these agents. Throughout, we assume that every agent $i \in [n]$ has an additive valuation function $v_i : 2^{\mathcal{G}} \to \mathbb{Z}_{\geq 0}$,[3] where $\mathcal{V}$ denotes the set of all additive valuation functions, $v_i(\emptyset) = 0$, and additivity implies that for a $S \subseteq \mathcal{G}$ (which we often refer to as a bundle), $v_i(S) = \sum_{g \in S} v_i(\{g\})$. For ease of notation, we often

[2] In the AGT literature, the term *stability* is usually used in the context of stable algorithms in two-sided matching [GS62]. In fact, this is also the case in Chapter 2 where we talk about stability in two-sided matching. Additionally, the same term is used in many different contexts in the computer science literature broadly (e.g., in learning theory, or when talking about, say, stable sorting algorithms). Our choice of the term stability here stems from the usage of this term in learning theory (see Section 5.3 for an extended discussion) and therefore should not be confused with the notion of stability in two-sided matching.

[3] We assume valuations are integers to model practical deployment of fair division algorithms (e.g., adjusted winner protocol, Spliddit). All the results hold even if we assume that the valuations are non-negative real numbers.

omit $\{g\}$ and instead just write it as $v_i(g)$. We also assume throughout that $\forall i \in [n]$, $v_i(\mathcal{G}) = T$ for some $T \in \mathbb{Z}^+$, and that $\forall g \in \mathcal{G}$, $v_i(g) > 0$. Although the assumption that agents have positive value for a good may not be valid in certain situations, in Section 5.2.2 we argue why this is essential in order to obtain anything interesting in context of our problem. Finally, for $S \subseteq \mathcal{G}$ and $k \in [n]$, we use $\Pi_k(S)$ to denote the set of ordered partitions of S into k bundles, and for an allocation $A \in \Pi_n(\mathcal{G})$, where $A = (A_1, \ldots, A_n)$, use A_i to denote the bundle allocated to agent i.

We are interested in deterministic algorithms $\mathcal{M} : \mathcal{V}^n \to \Pi_n(\mathcal{G})$ that produce fair and efficient (i.e., Pareto optimal) allocations. For $i \in [n]$ and a profile of valuation functions $(v_i, v_{-i}) \in \mathcal{V}^n$, we use $v_i(\mathcal{M}(v_i, v_{-i}))$ to denote the utility that i obtains from the allocation returned by $\mathcal{M}$. Pareto optimality and the fairness notions considered are defined below.

Definition 14 (β-Pareto optimality (β-PO))**.** Given a $\beta \geq 1$, an allocation $A \in \Pi_n(\mathcal{G})$ is β-Pareto optimal if

$$\forall A' \in \Pi_n(\mathcal{G}) : \ (\exists i \in [n], v_i(A'_i) > \beta \cdot v_i(A_i)) \Rightarrow (\exists j \in [n], v_j(A'_i) < v_j(A_j)) \, .$$

In words, an allocation is β-Pareto-optimal if no agent can be made strictly better-off by a factor of β without making another agent strictly worse-off; Pareto optimality refers to the special case when $\beta = 1$. For allocations A, A', we say that A' β-Pareto dominates A if every agent receives at least as much utility in A' as in A and at least one agent is strictly better-off by a factor of β in A'.

We consider several notions of fairness, namely, pairwise maximin share (PMMS), envy-freeness up to least positively valued good (EFX), and envy-freeness up to one good (EF1). Among these, the main notion we talk about here (and design algorithms for) is PMMS, which is the strongest and which implies the other two notions. Informally, an allocation A is said to be a PMMS allocation if every agent i is guaranteed to get a bundle A_i that she values more than the bundle she receives when she plays *cut-choose* with any other agent j (i.e., she partitions the combined bundle of her allocation and the allocation A_j of j and receives the one she values less).

Definition 15 (pairwise maximin share (PMMS))**.** An allocation $A \in \Pi_n(\mathcal{G})$ is a pairwise maximin share (PMMS) allocation if

$$\forall i, \forall j \in [n] : \ v_i(A_i) \geq \max_{B \in \Pi_2(A_i \cup A_j)} \min\{v_i(B_1), v_i(B_2)\}.$$

Next, we define EFX and EF1, which as mentioned above are weaker than PMMS.

Definition 16 (envy free up to any positively valued good (EFX))**.** An allocation $A \in \Pi_n(\mathcal{G})$ is envy-free up to any positively valued good (EFX) if

$$\forall i, \forall j \in [n], \forall g \in A_j \text{ with } v_i(g) > 0 : \; v_i(A_i) \geq v_i(A_j \setminus \{g\}).$$

In words, an allocation is said to be EFX if agent i is no longer envious after removing any positively valued good from agent j's bundle.

Definition 17 (envy free up to one good (EF1))**.** An allocation $A \in \Pi_n(\mathcal{G})$ is envy-free up to one good (EF1) if

$$\forall i, \forall j \in [n], \exists g \in A_j : \; v_i(A_i) \geq v_i(A_j \setminus \{g\}).$$

In words, an allocation is said to be EF1 if agent i is no longer envious after removing some good from agent j's bundle.

In addition to the requirement that the algorithms be fair and efficient, we would also ideally like it to be stable. We define the notion of stability (and its relaxations) in the next section.

5.2.1 Stability

At a high-level, our notion of stability captures the idea that an agent should not experience a large change in utility as long as they make "small" or "innocuous" mistakes while reporting their preferences. Naturally, a more formal definition requires us to first define what constitutes a 'mistake' and what we mean by "small" or "innocuous" mistakes in the context of fair division. So, below, for an $i \in [n]$, $v_i \in \mathcal{V}$, and $\alpha > 0$, we define what we refer to as α-neighbours of v_i. According to this definition, the closer α is to zero, the "smaller" is the 'mistake', since smaller values of α indicate that agent i's report v_i' (which in turn is the 'mistake') is closer to their true valuation function v_i.

Definition 18 (α-neighbours of v_i (α-$N(v_i)$))**.** For $i \in [n]$, $T \in \mathbb{Z}^+$, $\alpha > 0$, and $v_i \in \mathcal{V}$, define α-$N(v_i)$, the set of α-neighbouring valuations of v_i, to be the set of all v_i' such that

- $\sum_{g \in \mathcal{G}} v'(g) = T$ (i.e., $v_i(\mathcal{G}) = v_i'(\mathcal{G})$),
- $\forall g, \forall g' \in \mathcal{G}$, $v(g) \geq v(g') \Leftrightarrow v'(g) \geq v'(g')$ (i.e., the ordinal information over the singletons is maintained), and

- $\|v_i' - v_i\|_1 = \sum_{g \in \mathcal{G}} |v'(g) - v(g)| \leq \alpha$.[4]

Throughout, we often refer to the valuation function v_i of agent i as its *true valuation function* or *true report* and, for some $\alpha > 0$, $v_i' \in \alpha\text{-}N(v_i)$ as its *mistake* or *misreport*. Note that although *true valuation function*, *true report*, and *misreport* are terms one often finds in the mechanism design literature which considers strategic agents, we emphasize that here we are not talking about strategic agents, but about agents who are just unsophisticated in the sense that they are unable to accurately convert their preferences into cardinal values. Also, although we write $\alpha > 0$, it should be understood that in the context of our results here, since the valuation functions of the agents are integral, the only valid values of α are when it is an integer. We use this notation for ease of exposition and because all our results hold even if we instead use real-valued valuations.

Now that we know what constitutes a mistake, we can define our notion of stability.

Definition 19 (α-stable algorithm)**.** For $\alpha > 0$, an algorithm $\mathcal{M}$ is said to be α-stable if $\forall i \in [n], \forall v_i, \forall v_{-i}$, and $\forall v_i' \in \alpha\text{-}N(v_i)$,

$$v_i\left(\mathcal{M}(v_i, v_{-i})\right) = v_i\left(\mathcal{M}(v_i', v_{-i})\right). \tag{5.1}$$

In words, for an $\alpha > 0$, an algorithm is said to be α-stable if for every agent $i \in [n]$ with valuation function v_i and all possible reports of the other agents, the utility agent i obtains is the same when reporting v_i and v_i'.Given the definition above, we have the following for what it means for an algorithm to be *stable*.

Definition 20 (stable algorithm)**.** An algorithm $\mathcal{M}$ is stable if it is α-stable for all $\alpha > 0$.

Although the "for all" in the definition above might seem like a strong requirement at first glance, it is not, for one can easily show that the following observation holds.

Observation 1. An algorithm $\mathcal{M}$ is stable if and only if there exists an $\alpha > 0$ such that $\mathcal{M}$ is α-stable.

Proof. The forward part (i.e., the 'if' part) is obvious since by definition an algorithm is stable if it is α-stable for all $\alpha > 0$. To see the 'only if' part, let us assume that $\mathcal{M}$ is α-stable for some $\alpha > 0$. Next, for an arbitrary $\alpha' \neq \alpha$, let us consider an arbitrary agent

[4] We present our results using the L_1-norm, although qualitatively they do not change if we use, say, the L_∞-norm.

$i_1 \in [n]$, an arbitrary v_{i_1}, an arbitrary $v'_{i_1} \in \alpha'\text{-}N(v_{i_1})$, and arbitrary reports of the other agents denoted by v_{-i_1}.

First, note that for any $\alpha' < \alpha$, we have that

$$v'_{i_1} \in \alpha'\text{-}N(v_{i_1}) \Rightarrow v'_{i_1} \in \alpha\text{-}N(v_{i_1}),$$

which in turn implies that $\mathcal{M}$ is α'-stable since it is α-stable.

Next, let us consider an $\alpha' > \alpha$. Now, observe that we can construct a (finite) series of valuations functions $v_{i_2}, \ldots, v_{i_n}$, where for all $j \in [n]$, $v_{i_{j+1}} \in \alpha\text{-}N(v_{i_j})$ and where $v_{i_{n+1}} = v'_{i_1}$. This along with the fact that $\mathcal{M}$ is α-stable implies that, for all $j \in [n]$,

$$v_{i_1}(\mathcal{M}(v_{i_j}, v_{-i_1})) = v_{i_1}(\mathcal{M}(v_{i_{j+1}}, v_{-i_1})).$$

So, using the equalities above, we have,

$$v_{i_1}(\mathcal{M}(v_{i_1}, v_{-i_1})) = v_{i_1}(\mathcal{M}(v_{i_{n+1}}, v_{-i_1})) = v_{i_1}(\mathcal{M}(v'_{i_1}, v_{-i_1})),$$

thus proving that $\mathcal{M}$ is α'-stable. $\square$

It is important to note that the definition for an α-stable algorithm is only saying that the utility agent i obtains (and not the allocation itself) when moving from v_i to v'_i is the same. Additionally, although the notion of stability in general may look too strong, it is important to note that there are several algorithms (e.g., the well-known EF1 *draft* mechanism [Car+16]) that satisfy this definition. In particular, one can immediately see from the definition of stability that every ordinal fair division algorithm—i.e., an algorithm that produces the same output for input profiles $(v_1, \ldots, v_n)$ and $(v'_1, \ldots, v'_n)$ as long as $\forall g, g' \in \mathcal{G}, v_i(g) \geq v_i(g') \Leftrightarrow v'_i(g) \geq v'_i(g')$—is stable.

However, in general, and as we will see in Section 5.4.2, the equality in (5.1) can be too strong a requirement. Therefore, in the next section we propose two relaxations to the strong requirement of stability.

5.2.1.1 Approximate notions of stability

We first introduce the weaker relaxation which we refer to as weak-approximate-stability. Informally, weak-approximate-stability basically says that the utility that an agent experiences as a result of reporting a neighbouring instance is not too far away from the what would have been achieved if the reports were exact.

Definition 21 ((ϵ, α)-weakly-approximately-stable algorithm)**.** For an $\alpha > 0$ and $\epsilon \geq 1$, an algorithm $\mathcal{M}$ is said to be (ϵ, α)-weakly-stable if $\forall i \in [n], \forall v_i, \forall v_{-i}$, and $\forall v_i' \in \alpha\text{-}N(v_i)$,

$$\frac{1}{\epsilon} \leq \frac{v_i\left(\mathcal{M}(v_i', v_{-i})\right)}{v_i\left(\mathcal{M}(v_i, v_{-i})\right)} \leq \epsilon. \tag{5.2}$$

Although the definition above might seem like a natural relaxation of the notion of stability, as it will become clear soon, it is a bit weak. Therefore, below we introduce the stronger notion which we refer to as approximate-stability. However, before this, we introduce the following, which, for a given valuation function v, defines the set, equiv(v), of valuation functions v' such that the ordinal information over the bundles is the same in both v and v'.

Definition 22 (equiv(v))**.** For a valuation function $v : 2^{\mathcal{G}} \rightarrow \mathbb{Z}_{\geq 0}$, equiv($v$) refers to the set of all valuation functions v' such that for all $S_1, S_2 \subseteq \mathcal{G}$,

$$v(S_1) \geq v(S_2) \Leftrightarrow v'(S_1) \geq v'(S_2).$$

In words, equiv(v) refers to set all of valuation functions v' such that v and v' induce the same weak order over the set of all bundles (i.e., over the set $2^{\mathcal{G}}$). Throughout, for an $i \in [n]$, we say that two instances (or profiles) (v_i, v_{-i}) and (v_i', v_{-i}) are equivalent if $v_i' \in \text{equiv}(v_i)$. Also, we say that v_i and v_i' are *ordinally equivalent* if $v_i' \in \text{equiv}(v_i)$.

Equipped with this notion, we can now define approximate-stability. Informally, an algorithm is approximately-stable if it is weakly-approximately-stable and if with respect to every instance that is equivalent to the true reports, it is stable.

Definition 23 ((ϵ, α)-approximately-stable algorithm)**.** For an $\alpha > 0$ and $\epsilon \geq 1$, an algorithm $\mathcal{M}$ is said to be (ϵ, α)-approximately-stable if $\forall i \in [n], \forall v_i, \forall v_{-i}$,

- $\forall v_i' \in \text{equiv}(v_i)$, $v_i\left(\mathcal{M}(v_i, v_{-i})\right) = v_i\left(\mathcal{M}(v_i', v_{-i})\right)$, and
- $\mathcal{M}$ is (ϵ, α)-weakly-approximately-stable.

Note that when $\epsilon = 1$ the definitions for both the relaxations (i.e., weak-approximate-stability and approximate-stability) collapse to the one for α-stable algorithms. Also, throughout, we say that an algorithm is ϵ-approximately-stable for $\alpha \leq K$ if it is (ϵ, α)-approximately-stable for all $\alpha \in (0, K]$ (similarly for ϵ-weakly-stable).

Although the definition of approximate-stability might be seem a bit contrived at first glance, it is important to note that this is not the case. The requirement that the algorithm be stable on equivalent instances is natural because of the following observation which states that with respect to all the notions that we talk about here any algorithm that satisfies such a notion can always output the same allocation for two instances that are equivalent.[5]

Observation 2. Let P be a property that is one of EF1, EFX, PMMS, or PO. For all $i \in [n]$, if an allocation $(A_1, \ldots, A_n)$ satisfies property P with respect to the profile (v_i, v_{-i}), then $(A_1, \ldots, A_n)$ also satisfies property P with respect to the profile (v'_i, v_{-i}), where $v'_i \in \text{equiv}(v_i)$.

Proof. Consider an arbitrary agent $i \in [n]$, and recall from Definition 22 that for any $v'_i \in \text{equiv}(v_i)$, and for any two sets $S_1, S_2 \subseteq \mathcal{G}$, $v'_i(S_1) \geq v'_i(S_2)$ if and only $v_i(S_1) \geq v_i(S_2)$. Given this, the observation follows by using the definitions of the stated properties. □

5.2.2 Some Q & A on assumptions and definitions

Why the assumption of positive values for the goods? To see why we make this assumption, consider the fair division instance in Table 5.2a. Next, consider an arbitrary algorithm $\mathcal{M}$ that is EF1 and PO, and let us assume w.l.o.g. that the allocation returned by the algorithm is $(\{g_2\}, \{g_1\})$. Now, let us consider the instance in Table 5.2b. Note that since $\mathcal{M}$ returns an allocation that is PO and EF1, therefore the output with respect to the instance in Table 5.2b has to be either $(\{g_1\}, \{g_2\})$ or $(\emptyset, \{g_1, g_2\})$.

	A	B
g_1	T-1	T-1
g_2	1	1

(5.2a) Original instance

	A	B
g_1	T	T-1
g_2	0	1

(5.2b) Instance where agent A makes a mistake

Given this, consider a scenario where the valuation function mentioned in Table 5.2a is the true valuation (v_1) of agent A. When reporting correctly, she receives the good g_2. Table 5.2b shows agent A's misreport (v'_1), where $\alpha = 2$ and in which case she receives the good g_1 or none of the goods. Therefore, now, for any algorithm that is EF1 and PO, we either have $\frac{v_1(g_1)}{v_1(g_2)} = \frac{T}{1}$, or $\frac{v_1(\emptyset)}{v_1(g_2)} = 0$, and both of these are not informative or useful.

Is there any direct connection by the notion of stability and strategyproofness? At first glance, the definition of stable algorithms (or its relaxations defined in Section 5.2.1.1)

[5]Note that Observation 2 is only valid for the exact versions of these notions and not for their approximate counterparts.

might seem very similar to the definition for (approximately) strategyproof algorithms (i.e., algorithms where truthful reporting is an (approximately) weakly-dominant strategy for all the agents). Although there is indeed some similarity, it is important to note that these are different notions and neither does one imply the other. For instance, there is a stable algorithm that is EF1 (one can easily see that the well-known EF1 *draft* algorithm [Car+16, Sec. 3] is stable), but there is no strategyproof algorithm that is EF1 [Ama+17, App. 4.6].

5.3 Related Work

Now that we have defined our notion, we can better discuss related work. There are several lines of research that are related to the topic of this chapter. Some of the connections we discuss are in very different contexts and are by themselves very active areas of research. In such cases we only provide some pointers to the relevant literature.

Connections to algorithmic stability, differential privacy, and algorithmic fairness. Algorithmic stability captures the idea of stability by employing the principle that the output of an algorithm should not "change much" when a "small change" is made to the input. To the best of our knowledge, notions of stability have not been considered in the computational social choice/algorithmic game theory literature. However, the computational learning theory literature has considered various notions of stability and has, for instance, used them to draw connections between the stability of a learning algorithm and its ability to generalize (e.g., [BE02; Sha+10]). Although the notion of stability here is based on the same principle, it is defined differently from the ones in this literature. Here we are concerned about the change in utility than an agent experiences when she perturbs her input and deem an algorithm to be approximately-stable if this change is small.

Algorithmic stability can in turn be connected to differential privacy [Dwo+06; Dwo06]. Informally, differential privacy (DP) requires that the probability of an outcome does not "change much" on "small changes" to the input. Therefore, essentially, DP can be considered as a notion of algorithmic stability, albeit a very strong one as compared to the ones studied in the learning theory literature (see the discussion in [Dwo+15, Sec. 1.4]) and the one we consider. In particular, if we were considering randomized algorithms, it is indeed the case that an ϵ-differentially private algorithm is $\exp(\epsilon)$-stable, just like how an ϵ-differentially private algorithm is a $(\exp(\epsilon) - 1)$-dominant strategy mechanism [MT07]. Nevertheless, we believe that the notion we introduce here is independently useful, and is different from DP in a few ways. First, the motivation is completely different. We believe that our notion may be important even in situations where privacy is not a concern. Second, in this chapter we are only concerned with deterministic algorithms and one can easily see that DP is too strong a notion for this case as there are no deterministic and differentially private algorithms that have a range of at least two.

Finally, the literature on algorithmic (individual-based) fairness captures the idea of fairness by employing the principle that "similar agents" should be "treated similarly" [Dwo+12]. Although this notion is employed in contexts where one is talking about two different individuals, note that one way to think of our stability requirement is to think of it as a fairness requirement where two agents are considered similar if and only if they have similar inputs (i.e., say, if one's input is a perturbation of the other's). Therefore, thinking this way, algorithmic fairness can be considered as a generalization of stability, and just like DP it is much stronger and only applicable in randomized settings. In fact, algorithmic fairness can be seen as a generalization of DP (see the discussion in [Dwo+12, Sec. 2.3]) and so our argument above as to why our notion is useful is relevant even in this case.

Connections to robust algorithm and mechanism design. Informally, an algorithm is said to be robust if it "performs-well" even under "slightly different" inputs or if the underlying model is different from the one the designer has access to. This notion has received considerable amount of attention in the algorithmic game theory and social choice literature. For instance—and although this line of work does not explicitly term their algorithms as "(approximately) robust"—the flurry of work that takes the implicit utilitarian view considers scenarios where the agents have underlying cardinal preferences but only provide ordinal preferences to the designer. The goal of the designer in these settings is to then use these ordinal preferences in order obtain an algorithm or mechanism that "performs well" (in the approximation sense, with respect to some objective function) with respect to all the possible underlying cardinal preferences (e.g., [Bou+15; AS16; AZ17; GKM17; AA18]). Additionally, and more explicitly, robust algorithm design has been considered, for instance, in the context of voting (e.g., [SYE13; Bre+17]) and the stable marriage problem [ML18; MV18; CSS19], and robust mechanism design has been considered in the context of auctions [CMA12; CMA14; CMA15] and facility location [ML19]. Although, intuitively, the concepts of robustness and stability might seem quite similar, it is important to note that they are different. Stability requires that the outcome of an algorithm does not "change much" if one of the agents slightly modifies its input. Therefore, the emphasis here is to make sure that the outcomes are not very different as long as there is a small change to the input associated with one of the agents. Robustness, on the other hand, requires the outcome of an algorithm to remain "good" (in the approximation sense) even if the underlying inputs are different from what the algorithm had access to. Therefore, in this case the emphasis is on making sure that the same output (i.e., one that is computed with the input the algorithm has access to) is not-too-bad with respect to a set of possible underlying true inputs (but ones the algorithm does not have access to). More broadly, one can think of robustness as a feature that a designer aspires to to ensure that the outcome of their algorithm is not-too-bad even if the model assumed, or the input they have access to, is slightly inaccurate, whereas stability in the context that we use here is more of a feature that is in service of unsophisticated agents who are prone to making mistakes when converting their preferences to cardinal values.

Related work on fair division of indivisible goods. The problem of fairly allocating indivisible goods has received considerable attention, with several works proposing different notions of fairness [Lip+04; Bud11; Car+16] and computing allocations that satisfy these notions, sometimes along with some efficiency requirements [Car+16; BKV18; PR18; CGH19]. Our focus in this chapter is also on the problem of computing fair and efficient allocations, but in contrast to previous work our focus is on coming up with algorithms that are also (approximately) stable. While many of these papers address the general case of $n \geq 2$ agents, the results here are restricted to the case of two agents.

Although a restricted case, the two agent case is an important one and has been explicitly considered in several previous works [BKK12; Ram13; BKK14; VK14; Azi15; PR18]. Among these, the work that is most relevant to our results here is that of Ramaekers [Ram13]. In particular, and although the results here we derived independently, Ramaekers's paper contains two results that are similar to the ones we have here—first, a slightly weaker version of the $n = 2$ case of Theorem 34, and second, a slightly weaker version of Theorem 35. The exact differences are outlined in Sections 5.4.3 and 5.4.4 since we need to introduce a few more notions to make them clear.

In addition to the papers mentioned above—all of which adopt the model as in this chapter where the assumption is that the agents have cardinal preferences—there is also work that considers the case when agents have ordinal preferences [BEL10; BKK14; Azi+15; SAH17]. Although this line of work is related in that ordinal algorithms are stable, it is also quite different since usually the goal in these papers is to compute fair allocations if they exist or study the complexity of computing notions like possibly-fair or necessarily-fair allocations.

5.4 Approximate-Stability in Fair Allocation of Indivisible Goods

Our aim is to design (approximately) stable algorithms for allocating a set of indivisible goods among two agents that guarantee pairwise maximin share (PMMS) and Pareto

	A	B
g_1	T-1	T-2
g_2	1	2

(5.3a) Original instance

	A	B
g_1	T-3	T-2
g_2	3	2

(5.3b) Instance where agent A makes a mistake

optimal (PO) allocations. However, before we try to design new algorithms, the first question that arises is: *How do the existing algorithms fare? How stable are they?* We address this below.

5.4.1 How (approximately) stable are the existing algorithms?

We consider the following well-studied algorithms that guarantee PO and at least EF1. (We refer the reader to Appendix D.1 for brief descriptions of these algorithms.)

i) Adjusted winner protocol [BT96a; BT96b]; returns an EF1 and PO allocation for two agents.

ii) Leximin solution [PR18]; returns a PMMS and PO allocation for two agents.

iii) Maximum Nash Welfare solution [Car+16]; returns an EF1 and PO allocation for any number of agents.

iv) Fisher-market based algorithm [BKV18]; returns an EF1 and PO allocation for any number of agents.

All the algorithms mentioned above perform poorly even in terms of the weaker relaxation of stability, i.e., weak-approximate-stability. To see this, consider the instance in Table 5.3a, and note that the allocation that is output by any of these algorithms is agent A getting g_1 and agent B getting g_2. Next, consider the instance in Table 5.3b. In this case, if we use any of these algorithms, then the allocation that is output is agent A getting g_2 and agent B getting g_1.

Given this, let the values mentioned in Table 5.3a constitute the true valuation function (v_1) of agent A. When reporting these, she receives the good g_1. Table 5.3b shows agent A's misreport (v_1') in which case she receives the good g_2. Recall from the definition of α-neighbours of v_i (Definition 18) that $v_1' \in \alpha\text{-}N(v_i)$, where $\alpha = 4$. Therefore, we now have, $\frac{v_1(g_2)}{v_1(g_1)} = \frac{1}{T-1}$, or in other words, all the four algorithms mentioned above are $(T-1)$-weakly-approximately-stable, even when $\alpha = 4$.

	A	B
g_1	$\frac{T}{3}$	$\frac{T}{3}$
g_2	$\frac{2T}{9}$	$\frac{2T}{9}$
g_3	$\frac{2T}{9}$	$\frac{2T}{9}$
g_4	$\frac{2T}{9}$	$\frac{2T}{9}$

(5.4a) Original instance

	A	B
g_1	$\frac{T}{3}$	$T-3$
g_2	$\frac{2T}{9}$	1
g_3	$\frac{2T}{9}$	1
g_4	$\frac{2T}{9}$	1

(5.4b) Instance where agent B makes a mistake

5.4.2 Are there fair and efficient algorithms that are stable?

The observation that previously studied algorithms perform poorly even in terms of the weaker relaxation of stability implies that we need to look for new algorithms. So, now, a natural question that arises here is: *Is there any hope at all for algorithms that are fair, PO, and stable?* Note that without the requirement of PO the answer to this question is a "Yes"—at least when the fairness notion that is being considered is EF1, since it is easy to observe that the well-known *draft* algorithm (where agents take turns picking their favourite good among the remaining goods) that is EF1 [Car+16, Sec. 3] is stable. However, if we require PO, then we show that the answer to the question above is a "No," in that there are no stable algorithms that always return an EF1 and even approximately-PO allocation.

Theorem 33. *Let $\mathcal{M}$ be an algorithm that is stable and always returns an EF1 allocation. Then, for any β in $[1, \frac{T-3}{2}]$, $\mathcal{M}$ cannot be β-PO.*

Proof. Consider the instance in Table 5.4a which represents agents' true valuation functions. For simplicity we assume that $T \mod 9 \equiv 0$. Next, since the agents are symmetric and it is easy to verify that in every EF1 allocation each of the agents have to get exactly two goods, let us assume w.l.o.g. that agent A receives g_1. Given this, now consider the instance in Table 5.4b where agent B makes a mistake. Let us denote agent B's true utility function as v, misreport as $v' \in \frac{4T}{3}\text{-}N(v)$, and (S_1, S_2) and (S_1', S_2') as outcomes for the instances in Tables 5.4a, 5.4b, respectively. From our discussion above, we know that $v(S_2) = \frac{4T}{9}$, since agent B gets two goods from the set $\{g_2, g_3, g_4\}$.

Since $\mathcal{M}$ is stable, we know that $v(S_2) = v(S_2')$. Now, it is easy to see that this is only possible if the set S_2' has exactly two goods from $\{g_2, g_3, g_4\}$. So, let us assume w.l.o.g. that $S_2' = \{g_2, g_3\}$. If this is the case, then note that the allocation $(\{g_2, g_3, g_4\}, \{g_1\})$ Pareto dominates the allocation (S_1', S_2') by a factor of $\frac{v'(g_1)}{v'(S_2')} = \frac{T-3}{2}$, which in turn proves our theorem. $\square$

Given this result and our observation in Section 5.4.1 that previously studied algorithms perform poorly even in terms of weak-approximate-stability, it is clear that the best one can hope for is to design new algorithms that are fair, efficient, and approximately-stable. In Section 5.4.4 we show that this is possible. However, before we do that, in the next section we first present a necessary and sufficient condition for PMMS allocations when there are $n \geq 2$ agents.

5.4.3 A necessary and sufficient condition for existence of PMMS allocations

The general characterization result for PMMS allocations presented here will be useful in the next section to design an approximately-stable algorithm that produces a PMMS and PO allocation for the case of two agents. Additionally, we also believe that the result might potentially be of independent interest.

For an agent $\ell \in [n]$, a set $Q \subseteq \mathcal{G}$, and a set $S \subseteq Q$, the result uses a notion of rank of S, denoted by $r_\ell^Q(S)$, and defined as the number of subsets of Q that have value at most $v_\ell(S)$. More formally,

$$r_\ell^Q(S) = \Big|\{P \mid P \subseteq Q, v_\ell(P) \leq v_\ell(S)\}\Big|. \tag{5.3}$$

The notion of rank has been previously considered in the fair division literature by Ramaekers [Ram13]. In particular, according to our notation, they talk about $r_\ell^{\mathcal{G}}(S)$ in the context of fair division among two agents who have a strict preference orders over the subsets of $\mathcal{G}$, and one of their results is a weaker (since they assume a strict order over subsets of $\mathcal{G}$ which is not assumed here) version of the $n = 2$ case of the theorem below [Ram13, Thm. 1(2)].

Theorem 34. *Given an instance with m indivisible goods and $n \geq 2$ agents with additive valuation functions, an allocation $A = (A_1, \dots, A_n)$ is a pairwise maximin share allocation if and only if $\forall i \in [n], \forall j \in [n]$, and $K_{ij} = A_i \cup A_j$,*

$$\min\left\{r_i^{K_{ij}}(A_i),\ r_j^{K_{ij}}(A_j)\right\} \geq 2^{|K_{ij}|-1}.$$

Before we present a formal argument to prove this theorem, we present a brief overview. Overall, the proof uses some observations about the ranking function. In particular, for a set Q and $S \subseteq Q$, one key observation is that the rank of S is high enough (more precisely, greater than $2^{|Q|-1}$) if and only if the value of this set is at least half that of Q. Once we have this, then the proof essentially follows by combining it with a few other observations about the ranking function and some simple counting arguments.

More formally, we first state the following claims about the ranking function. The proofs of these directly follow from the way the ranking function is defined.

Claim 6. Let $Q \subseteq \mathcal{G}$ and $i \in [n]$ be some agent. Then,

i) for $A \subseteq Q, B \subseteq Q$, $v_i(A) < v_i(B) \Leftrightarrow r_i^Q(A) < r_i^Q(B)$

ii) $r_i^Q(A) < r_i^Q(B) \Leftrightarrow r_i^{\mathcal{G}}(A) < r_i^{\mathcal{G}}(B)$.

Proof. To prove the first part, consider the sets $H_A = \{P \mid P \subseteq Q, v_i(P) \leq v_i(A)\}$ and $H_B = \{P \mid P \subseteq Q, v_i(P) \leq v_i(B)\}$. First, observe that $v_i(A) < v_i(B)$ if and only if $|H_A| < |H_B|$. Also, from the definition of the ranking function we know that $|H_A| < |H_B|$ if and only if $r_i^Q(A) < r_i^Q(B)$. Combining these we have our claim.

To prove the second part, observe that from the first part we know that $r_i^Q(A) < r_i^Q(B)$ if and only if $v_i(A) < v_i(B)$. Next, again using the first part with $Q = \mathcal{G}$, we have that $v_i(A) < v_i(B)$ if and only if $r_i^{\mathcal{G}}(A) < r_i^{\mathcal{G}}(B)$. Combining these we have our claim. ☐

Claim 7. Let $Q \subseteq \mathcal{G}$, $i \in [n]$ be some agent, $\ell \in \mathbb{Z}_{\geq 0}$, and $T_\ell = \{P \mid P \subseteq Q, r_i^Q(P) \leq \ell\}$. Then,

i) for $S \subseteq Q$, if $r_i^Q(S) = \ell$, then $|T_\ell| = \ell$

ii) $|T_\ell| \leq \ell$.

Proof. To prove the first part, consider the set $H = \{P \mid P \subseteq Q, v_i(P) \leq v_i(S)\}$. First, note that from Claim 6(i) we can see that $H = T_\ell$. Next, from the definition of $r_i^Q(S)$, we know that $|H| = \ell$. Also, every element in H will have a rank at most ℓ (since for each such set S', $v_i(S') \leq v_i(S)$) and every element outside of H will have a rank larger than ℓ (since for each such set S', $v_i(S') > v_i(S)$). Hence, i) follows.

To prove the second part, consider the largest $\ell' \leq \ell$ such that there exists some $S \subseteq Q$ with $r_i^Q(S) = \ell'$. Now, from i) we know that the number of subsets of Q with rank at most ℓ' is exactly ℓ' and hence from our choice of ℓ' the statement follows. ☐

Claim 8. Let $A = (A_1, \ldots, A_n)$ be an allocation, and $i, j \in [n]$ be some agents. If $K_{ij} = A_i \cup A_j$, then

$$r_i^{K_{ij}}(A_i) > 2^{|K_{ij}|-1} \Leftrightarrow v_i(A_i) \geq \frac{v_i(K_{ij})}{2}.$$

Proof. (⇒) Let us assume for the sake of contradiction that $\ell = r_i^{K_{ij}}(A_i) > 2^{|K_{ij}|-1}$, but $v_i(A_i) < \frac{v_i(K_{ij})}{2}$. Since $\ell > 2^{|K_{ij}|-1}$, we know from Claim 7(i) that there exists S and $S^c = K_{ij} \setminus S$, such that $r_i^{K_{ij}}(S) \leq r_i^{K_{ij}}(A_i)$ and $r_i^{K_{ij}}(S^c) \leq r_i^{K_{ij}}(A_i)$. This in turn implies that using Claim 6(i) we have that $v_i(S) \leq v_i(A_i) < \frac{v_i(K_{ij})}{2}$ and $v_i(S^c) \leq v_i(A_i) < \frac{v_i(K_{ij})}{2}$, which is impossible since the valuation functions are additive.

(⇐) Let A be an allocation such that $v_i(A_i) \geq \frac{v_i(K_{ij})}{2}$, but $r_i^{K_{ij}}(A_i) \leq 2^{|K_{ij}|-1}$. Now, consider the set $H_1 = \{P \mid P \subseteq K_{ij}, r_i^{K_{ij}}(P) > r_i^{K_{ij}}(A_i)\}$. From Claim 7(i) and using the fact that $r_i^{K_{ij}}(A_i) \leq 2^{|K_{ij}|-1}$ we know that $|H_1| \geq 2^{|K_{ij}|-1}$. Also, note that every set in H_1 has, by Claim 6(i), value greater than $v_i(A_i)$, which in turn implies that $A_j \notin H_1$. Next, consider $H_2 = \{P^c \mid P \in H_1, P^c = K_{ij} \setminus P\}$, where additivity implies that every set in H_2 has value less than $v_i(A_1)$. Note that $|H_1| + |H_2| \geq 2^{|K_{ij}|}$ and A_i is neither in H_1 nor H_2, which is impossible. ☐

Equipped with the claims above, we are now ready to prove our theorem.

Proof of Theorem 34. (⇒) Let us assume for the sake of contradiction that A is a PMMS allocation and that there exists i, j such that $\min\{r_i^{K_{ij}}(A_i), r_j^{K_{ij}}(A_j)\} < 2^{|K_{ij}|-1}$. W.l.o.g., let us assume that $r_i^{K_{ij}}(A_i) < 2^{|K_{ij}|-1}$. Next, consider the set $H = \{B \mid B \subseteq K_{ij}, r_i^{K_{ij}}(B) > r_i^{K_{ij}}(A_i)\}$. Since $r_i^{K_{ij}}(A_i) < 2^{|K_{ij}|-1}$, we know from Claim 7(ii) that $|H| > 2^{|K_{ij}|-1}$ (since there are $2^{|K_{ij}|}$ subsets of K_{ij}). This implies that there is a set S and its complement $S^c = K_{ij} \setminus S$ such that, $r_i^{K_{ij}}(A_i) < \min\{r_i^{K_{ij}}(S), r_i^{K_{ij}}(S^c)\}$, which in turn using Claim 6(i) implies that $v_i(A_i) < \min\{v_i(S), v_i(S^c)\}$. However, note that this contradicts the fact that i has an MMS partition w.r.t. j in A.

(⇐) Let us assume for the sake of contradiction that there exists an agent i such that i does not have an MMS partition w.r.t. j, but $\min\{r_i^{K_{ij}}(A_i), r_j^{K_{ij}}(A_j)\} \geq 2^{|K_{ij}|-1}$. This implies that $v_i(A_i) < \frac{v_i(K_{ij})}{2}$, which in turn using Claim 8 and the fact that $\min\{r_i^{K_{ij}}(A_i), r_j^{K_{ij}}(A_j)\} \geq 2^{|K_{ij}|-1}$, implies that $r_i^{K_{ij}}(A_i) = 2^{|K_{ij}|-1}$. Next, since i does not perceive (A_i, A_j) to be an MMS partition w.r.t. j, there must exist a partition (A'_i, A'_j) such that $A'_i \cup A'_j = K_{ij}$ and $\min\{v_i(A'_i), v_i(A'_j)\} > v_i(A_i)$. This implies that, using Claim 6(i), we have that $r_i^{K_{ij}}(A'_i) > r_i^{K_{ij}}(A_i)$ and $r_i^{K_{ij}}(A'_j) > r_i^{K_{ij}}(A_i)$, or in other words that a set (i.e., A'_i) and its complement (i.e., $A'_j = K_{ij} \setminus A'_i$) both have rank greater than $r_i^{K_{ij}}(A_i)$. Now, if this is case, then one can see that, since $r_i^{K_{ij}}(A_i) = 2^{|K_{ij}|-1}$, this implies there exists a set $S \subseteq K_{ij}$ and its complement $S^c = K_{ij} \setminus S$ such that $r_i^{K_{ij}}(S) \leq r_i^{K_{ij}}(A_i)$ and $r_i^{K_{ij}}(S^c) \leq r_i^{K_{ij}}(A_i)$.

However, this is impossible because we can now use Claim 8 to see that both $v_i(S)$ and $v_i(S^c)$ have value less than $\frac{v_i(K_{ij})}{2}$, which in turn contradicts the fact that v_i is an additive valuation function. $\square$

In the next section we use this result to show an approximately-stable algorithm that is PMMS and PO when there are two agents.

5.4.4 rank-leximin: An approximately-stable PMMS and PO algorithm for two agents

The idea of our algorithm is simple. Instead of the well-known Leximin algorithm where one aims to maximize the minimum utility any agents gets, then the second minimum utility, and so on, our approach, which we refer to as the rank-leximin algorithm (Algorithm 7), is to do a leximin-like allocation, but based on the ranks of the bundles that the agents receive. Here for an agent i and a bundle $B \subseteq \mathcal{G}$, by rank we mean $r_i^{\mathcal{G}}(B)$, as defined in (5.3). That is, the rank-leximin algorithm maximizes the minimum rank of the bundle that any agent gets, then it maximizes the second minimum rank, then the third minimum rank, and so on. Note that this in turn induces a comparison operator $\prec$ between two partitions and this is formally specified as rank-leximinCMP in Algorithm 7. Although the original leximin solution also returns a PMMS and PO allocation for two agents, recall that we observed in Section 5.4.1 that it does not provide any guarantee even in terms of weak-approximate-stability, even when $\alpha = 4$. Rank-leximin on the other hand is PMMS and PO for two agents and, as we will show, is also $(2 + \frac{12\alpha}{T})$-approximately-stable for all $\alpha \in (0, \frac{T}{3}]$. Additionally, it also returns an allocation that is PMMS and PO for any number of agents as long as they report ordinally equivalent valuation functions (see Definition 22).

Remark: Given the characterization result for PMMS allocations, one can come up with several algorithms that satisfy PMMS and PO. However, we consider rank-leximin here since it is a natural counterpart to the well-known leximin algorithm [PR18]. Additionally, it turns out that contrary to our initial belief the idea of rank-leximin is not new. Ramaekers [Ram13] considered it in the context of algorithms for two-agent fair division that satisfy Pareto-optimality, *anonymity*, *the unanimity bound*, and *preference monotonicity* (see [Ram13, Sec. 3] for definitions), where the *unanimity bound* is a notion that one can show is equivalent to the notion of Maximin share (MMS) that is used in the computational fair division literature. So, with caveat that Ramaekers [Ram13] assumes that the agents have a strict preference orders over the subsets of $\mathcal{G}$ (which is not assumed here), the result of Ramaekers [Ram13, Thm. 2] already proves that rank-leximin is PMMS and PO for two

Procedure: rank-leximinCMP(P, T)
Input: two partitions $P, T \in \Pi_n(\mathcal{G})$
Output: returns true if $P \prec T$, i.e., if P is before T in the rank-leximin sorted order

$R_P \leftarrow$ agents sorted in non-decreasing order of the rank of their bundles in P, i.e., based on $r_i^{\mathcal{G}}(P_i)$, with ties broken in some arbitrary but consistent way throughout
$R_T \leftarrow$ similar ordering as in R_P above, but based on $r_i^{\mathcal{G}}(T_i)$
for each $\ell \in [n]$ **do**
  $i \leftarrow R_P^{\ell}$ ▷ ℓ-th agent in R_P
  $j \leftarrow R_T^{\ell}$ ▷ ℓ-th agent in R_T
  if $r_i^{\mathcal{G}}(P_i) \neq r_j^{\mathcal{G}}(T_j)$ **then**
    return $r_i^{\mathcal{G}}(P_i) < r_j^{\mathcal{G}}(T_j)$
  end if
end for
return false

Main:
Input: for each agent $i \in [n]$, their valuation function $v_i : 2^{\mathcal{G}} \rightarrow \mathbb{R}_{\geq 0}$
Output: an allocation $A = (A_1, \ldots, A_n)$ that is PMMS and PO

$\mathcal{L} \leftarrow$ perform a rank-Lexmin sort on $\Pi_n(\mathcal{G})$ based on the rank-leximinCMP operator defined above
return $A = (A_1, \ldots, A_n)$ that is the last element in $\mathcal{L}$

Algorithm 7: rank-leximin algorithm

agents (MMS is equivalent to PMMS in the case of two agents), which is the result we show in Theorem 35. However, we still include our proof because of the indifference issue mentioned above, and since it almost follows directly from Theorem 34.

Below we first show that the rank-leximin algorithm always returns an allocation that is PMMS and PO for the case of two agents. The fact that it is PO can be seen by using some of the properties that we proved about the ranking function in the previous section, while the other property follows from combining Theorem 34 along with a simple pigeonhole argument. Following this, we also show that rank-leximin always returns a PMMS and PO allocation when there are $n \geq 2$ agents with ordinally equivalent valuation functions. The proof of this is slightly more involved and it proceeds by first showing how rank-leximin returns such an allocation when all the agents have identical valuation functions. Once we have this, then the theorem follows by repeated application of Observation 2.

Theorem 35. *Given an instance with m indivisible goods and two agents with additive valuation functions, the rank-leximin algorithm (Algorithm 7) returns an allocation that is*

PMMS and PO.

Proof. Let $A = (A_1, \ldots, A_n)$ be the allocation that is returned by the rank-leximin algorithm (Algorithm 7). Below we will first show that A is PO and subsequently argue why it is PMMS.

Suppose A is not Pareto optimal. Then there exists another allocation A' such that for all $i \in [n]$, $v_i(A'_i) \geq v_i(A_i)$, and the inequality is strict for at least one of the agents, say j. This in turn implies that using Claim 6 we have that for all $i \in [n]$, $r_i^{\mathcal{G}}(A'_i) \geq r_i^{\mathcal{G}}(A_i)$ and $r_j^{\mathcal{G}}(A'_j) > r_j^{\mathcal{G}}(A_j)$. However, this implies that from the procedure rank-leximinCMP in Algorithm 7 we have that $A \prec A'$, and this in turn directly contradicts the fact that A was the allocation that was returned.

To show that A is PMMS, consider agent 1 and all the sets S such that $r_1^{\mathcal{G}}(S) \geq 2^{m-1}$. From Claim 7 we know that there are at least $2^{m-1} + 1$ such sets. Therefore, if $S^c = \mathcal{G} \setminus S$, then there is at least one S such that $r_2^{\mathcal{G}}(S^c) \geq 2^{m-1}$. This implies (S, S^c) is an allocation such that $\min\{r_1^{\mathcal{G}}(S), r_2^{\mathcal{G}}(S^c)\} \geq 2^{m-1}$. Now since rank-leximin maximizes the minimum rank that any agent receives, we have that $\min\{r_1^{\mathcal{G}}(A_1), r_2^{\mathcal{G}}(A_2)\} \geq 2^{m-1}$, and so now we can use Theorem 34 to see that A is a PMMS allocation. $\square$

Theorem 36. *Given an instance with m indivisible goods and $n \geq 2$ agents with additive valuation functions, the rank-leximin algorithm (Algorithm 7) returns an allocation A that is PMMS and PO if all the agents report ordinally equivalent valuation functions.*

Proof. To prove this we first show how rank-Leximin always returns a PMMS and PO allocation $(A_1, \ldots, A_n)$ for agents with identical valuation functions. Once we have that, then the theorem follows by repeated application of Observation 2 and by observing that the rank-leximin algorithm produces the same output for two instances that are equivalent.

First, note that we already know from the proof of Theorem 35 that it is PO with respect to any instance with n agents. So we just need to argue that it produces a PMMS allocation when the reports are identical. To see this, recall from Theorem 34 that we need to show that for all $i, j \in [n]$, $\min\{r_i^Q(A_i), r_j^Q(A_j)\} \geq 2^{k-1}$, where $Q = A_i \cup A_j$ and $k = |Q|$.

Suppose this was not the case and there exists agents i, j such that $r_i^Q(A_i) < 2^{k-1}$. Now, let us consider the set $H = \{S \mid S \subseteq Q \text{ and } r_j^Q(S) > r_i^Q(A_i)\}$. Since $r_i^Q(A_i) < 2^{k-1}$, we know from Claim 7(ii) that $|H| > 2^{k-1}$ (since there are 2^k subsets of Q in total). This in turn implies that, there is a set S and its complement S^c, such that both $r_j^Q(S)$ and $r_j^Q(S^c)$ are greater than $r_i^Q(A_i)$. So, now, consider these sets S and S^c, and ask agent i to pick the one she values the most. Let us assume without loss of generality that this is S. Note that since the valuations are additive and hence $v_i(S) \geq \frac{1}{2}v_i(Q) > v_i(A_i)$, we know from Claim 8 that $r_i^Q(S) \geq 2^{k-1} > r_i^Q(A_i)$. This in turn implies that,

$$\begin{aligned}
&\min\{r_i^Q(S), r_j^Q(S^c)\} > r_i^Q(A_i) \geq \min\{r_i^Q(A_i), r_j^Q(A_j)\} \\
\Rightarrow &\min\{r_i^Q(S), r_i^Q(S^c)\} > r_i^Q(A_i) \geq \min\{r_i^Q(A_i), r_i^Q(A_j)\},
\end{aligned}$$

where the implication follows from the fact that i and j have identical valuation functions and so for any $S \subseteq \mathcal{G}$ and $A \subseteq S$, we have $r_i^S(A) = r_j^S(A)$.

The observation above implies that we can use Claim 6(ii) to see that,

$$\begin{aligned}
&\min\{r_i^{\mathcal{G}}(S), r_i^{\mathcal{G}}(S^c)\} > \min\{r_i^{\mathcal{G}}(A_i), r_i^{\mathcal{G}}(A_j)\} \\
\Rightarrow &\min\{r_i^{\mathcal{G}}(S), r_j^{\mathcal{G}}(S^c)\} > \min\{r_i^{\mathcal{G}}(A_i), r_j^{\mathcal{G}}(A_j)\},
\end{aligned}$$

where the implication again follows from the fact that i and j have identical valuation functions.

Now, consider the allocation $A' = (A_1, \ldots, A_{i-1}, S, A_{i+1}, \ldots, A_{j-1}, S^c, A_{j+1}, \ldots, A_n)$. From the discussion above we know that $A \prec A'$ according to the rank-LeximinCMP operator in Algorithm 7. However, this contradicts the fact that A was the allocation that was returned by the rank-Leximin algorithm. $\square$

Now that we know rank-leximin produces a PMMS and PO allocation, we will move on to see how approximately-stable it is in the next section. However, before that we make the following remark.

Remark: The rank-leximin algorithm takes exponential time. Note that this is not surprising since finding PMMS allocations is NP-hard even for two identical agents—one can see this by a straightforward reduction from the well-known Partition problem.

5.4.4.1 rank-leximin is $\left(2 + \mathcal{O}(\frac{\alpha}{T})\right)$-approximately-stable for $\alpha \leq \frac{T}{3}$

Before we show how approximately-stable rank-leximin is, we prove the following claims which will be useful in order to prove our result.

Claim 9. Let v be an additive utility function and for some $\alpha > 0$, let $v' \in \alpha\text{-}N(v)$. Then, for any $S_1, S_2 \subseteq G$,

i) $|v(S_1) - v'(S_1)| \leq \frac{\alpha}{2}$

ii) if $v(S_1) > v(S_2)$ and $v'(S_1) \leq v'(S_2)$, then $v(S_1) - v(S_2) \leq \alpha$.

Proof. To prove the first part, suppose $|v(S_1) - v'(S_1)| > \frac{\alpha}{2}$. Let $S_1^c = \mathcal{G} \setminus S_1$. Since v, v' are additive, we know that $v(S_1) + v(S_1^c) = T$ (and similarly for v'), and so this in turn implies that $|v'(S_1^c) - v(S_1^c)| > \frac{\alpha}{2}$. So, using this, we have,

$$\begin{aligned}\sum_{g \in \mathcal{G}} |v(g) - v'(g)| &= \sum_{g \in S_1} |v(g) - v'(g)| + \sum_{g \in S_1^c} |v'(g) - v(g)| \\ &\geq |\textstyle\sum_{g \in S_1} (v(g) - v'(g))| + |\textstyle\sum_{g \in S_1^c} (v'(g) - v(g))| \\ &= |v(S_1) - v'(S_1)| + |v'(S_1^c) - v(S_1^c)| \\ &> \alpha,\end{aligned}$$

which is a contradiction since $v' \in \alpha\text{-}N(v)$.

To prove the second part, observe that from the first part we have,

$$\begin{aligned}v(S_1) - v(S_2) &\leq v'(S_1) + \frac{\alpha}{2} - (v'(S_2) - \frac{\alpha}{2}) \\ &= v'(S_1) - v'(S_2) + \alpha \\ &\leq \alpha,\end{aligned}$$

where the last inequality follows from the fact that $v'(S_1) \leq v'(S_2)$. □

Claim 10. Given an instance with m indivisible goods and two agents with additive valuation functions, let (A_1, A_2) be a PMMS allocation. If for an agent $i \in [2]$, $k = |A_j| \geq 2$, where $j \neq i$, then

i) $v_i(A_i) \geq v_i(M_i^j)$, where M_i^j is the maximum valued good of agent i in the bundle A_j

ii) $v_i(A_j) - v_i(A_i) \leq v_i(m_i^j)$, where m_i^j is the minimum valued good of agent i in the bundle A_j

iii) $v_i(A_i) \geq \frac{k-1}{2k-1}T$.

Proof. To prove the first part, let us assume that $v_i(A_i) < v_i(M_i^j)$. Since $k \geq 2$, we have another good $g \in A_j$ such that $g \neq M_i^j$. So, now, consider the allocation $(A_i \cup \{g\}, \{M_i^j\})$. Note that using additivity we have that $\min\{v_i(A_i \cup \{g\}), v_i(M_i^j)\} > v_i(A_i)$, which in turn contradicts the fact that (A_i, A_j) is a PMMS allocation.

To prove the second part, let us assume that $v_i(A_j) - v_i(A_i) > v_i(m_i^j)$. Next, consider the allocation $(A_i \cup \{m_i^j\}, A_j \setminus \{m_i^j\})$. Note that using additivity we have that $\min\{v_i(A_i \cup \{m_i^j\}), v_i(A_j \setminus \{m_i^j\})\} > v_i(A_i)$, which in turn contradicts the fact that (A_i, A_j) is a PMMS allocation.

To prove the third part, observe that from the second part we know that $v_i(A_i) \geq v_i(A_j) - v_i(m_i^j)$. Also, if $k = |A_j|$, then we can use the fact that the valuation functions are additive to see that $v_i(m_i^j) \leq \frac{v_i(A_j)}{k}$. So, using these, we have,

$$v_i(A_i) \geq v_i(A_j) - v_i(m_i^j) \geq v_i(A_j) - \frac{v_i(A_j)}{k} = (T - v_i(A_i))\left(\frac{k-1}{k}\right),$$

where the last inequality follows from the fact that $v_i(A_i) + v_i(A_j) = T$.

Finally, rearranging the term above we have our claim. □

Claim 11. Given an instance with m indivisible goods and two agents with additive valuation functions, let (A_1, A_2) be the allocation that is returned by the rank-leximin algorithm for this instance. If $k_1 = r_1^{\mathcal{G}}(A_1)$ and $k_2 = r_2^{\mathcal{G}}(A_2)$, then $\min\{k_1, k_2\} \geq 2^{m-1}$ and $\max\{k_1, k_2\} > 2^{m-1}$.

Proof. Since rank-leximin produces a PMMS allocation (Theorem 35), we know from Theorem 34 that $\min\{k_1, k_2\} \geq 2^{m-1}$. Also, if $\max\{k_1, k_2\} \leq 2^{m-1}$, then using Claim 8 we have that $v_1(A_1) < \frac{T}{2}$ and $v_2(A_2) < \frac{T}{2}$. However, this in turn contradicts the fact that rank-leximin is Pareto-optimal (Theorem 35) since swapping the bundles improves the utilities of both the agents. □

Equipped with the claims above, we can now prove the approximate-stability bound for rank-leximin. The proof here proceeds by first arguing about how rank-leximin is stable with respect to equivalent instances. This is followed by showing upper and lower bounds on how weakly-approximately-stable it is, which in turn involves looking at several different cases and using several properties (some of them proved above and some that we will introduce as we go along) about the allocation returned by the rank-leximin algorithm.

Theorem 37. *Given an instance with m indivisible goods and two agents with additive valuation functions, the rank-leximin algorithm is $(2 + \frac{12\alpha}{T})$-approximately-stable for all $\alpha \leq \frac{T}{3}$.*

Proof. Let us consider two agents with valuation functions v_1, v_2. Since the rank-leximin is symmetric, we can assume w.l.o.g. that agent 1 is the one making a mistake. Throughout, let us denote this misreport by v_1'. Also, let rank-leximin$(v_1, v_2) = (A_1, A_2)$, and rank-leximin$(v_1', v_2) = (S_1, S_2)$. Next, let us introduce the following notation that we will use throughout. For some arbitrary valuations y_1, y_2, if an allocation (C_1, C_2) precedes an allocation (B_1, B_2) in the rank-leximin order with respect to (y_1, y_2) (i.e., according to the rank-leximinCMP function in Algorithm 7), then we denote this by $(B_1, B_2) \succ_{y_1,y_2} (C_1, C_2)$. In the case they are equivalent according to the rank-leximin operator, then we denote it by $(B_1, B_2) \equiv_{y_1,y_2} (C_1, C_2)$. Additionally, we use $(B_1, B_2) \succeq_{y_1,y_2} (C_1, C_2)$ to denote that (C_1, C_2) either precedes or is equivalent to (B_1, B_2).

Equipped with the notations above, we now prove our theorem. To do this, first recall from the definition of approximate-stability (see Definition 23) that we first need to show that if $v_1' \in \text{equiv}(v_1)$, then $v_1(A_1) = v_1(S_1)$. To see why this is true, consider the rank-leximinCMP function in Algorithm 7 and observe that since $v_1' \in \text{equiv}(v_1)$, v_1, v_1' have the same bundle rankings, and hence for any two partitions $P, T \in \pi_n(\mathcal{G})$, $P \prec_{v_1,v_2} T$ (i.e., P appears before T in the rank-leximin order) if and only if $P \prec_{v_1',v_2} T$. This in turn along with the fact that rank-leximin uses a deterministic tie-breaking rule implies that rank-leximin$(v_1, v_2) = (A_1, A_2) =$ rank-leximin(v_1', v_2), thus showing that it is stable with respect to equivalent instances.

Having shown the above, let us move to the second part where we show how weakly-approxmiately-stable rank-leximin is. For the rest of this proof, let $v_1' \in \alpha\text{-}N(v_i)$, for some $\alpha \in (0, \frac{T}{3}]$. Also, for $i \in [2]$, let the ranking function associated with valuation functions v_i and v_1' be $r_i^{\mathcal{G}}(\cdot)$ and $r_1'^{\mathcal{G}}(\cdot)$, respectively. Throughout, in order to keep the notations simple, we use $r_1(\cdot)$, $r_1'(\cdot)$, and $r_2(\cdot)$ to refer to $r_1^{\mathcal{G}}(\cdot)$, $r_1'^{\mathcal{G}}(\cdot)$, and $r_2^{\mathcal{G}}(\cdot)$, respectively.

In order to prove the bound on weak-approximate-stability, we show two lemmas. The first one shows an upper bound of 2 for the ratio $\frac{v_1(S_1)}{v_1(A_1)}$ and the second one shows a lower bound of $\frac{1}{2+\frac{12\alpha}{T}}$.

Lemma 38. *For $\alpha > 0$, if $v_1' \in \alpha\text{-}N(v_1)$, rank-leximin$(v_1, v_2) = (A_1, A_2)$, and rank-leximin$(v_1', v_2) = (S_1, S_2)$, then*

$$\frac{v_1(S_1)}{v_1(A_1)} \leq 2.$$

Proof. To prove this, let us consider the following three cases. Note that since rank-leximin returns a PMMS allocation (Theorem 35) we know from Theorem 34 that these are the only three cases. Also, note that the bound is trivially true when $v_1(A_1) - v_1(S_1) \geq 0$. So below we directly consider the case when $v_1(S_1) > v_1(A_1)$.

Case 1. $r_1(A_1) > 2^{m-1}$**:** Since $r_1(A_1) > 2^{m-1}$, we know using Claim 8 that $v_1(A_1) \geq \frac{T}{2}$. Therefore, using this and the fact that $v(S_1) \leq T$, we have,

$$\frac{v_1(S_1)}{v_1(A_1)} \leq 2. \tag{5.4}$$

Case 2. $r_1(A_1) = 2^{m-1}$ **and** $|S_1| = 1$**:** First, observe that since $|S_1| = 1$ and $v_1(A_1) < v_1(S_1)$, we have that $S_1 \subseteq A_2$. This in turn implies $v_1(S_1) \leq v_1(A_2)$ and so using this we have

$$\frac{v_1(S_1)}{v_1(A_1)} \leq \frac{v_1(A_2)}{v_1(A_1)}. \tag{5.5}$$

Now, in order to upper-bound (5.5), consider the following two cases.

i) $|A_2| \geq 2$**:** Let $k = |A_2|$. Since $k \geq 2$, we know from Claim 10 that $v_1(A_1) \geq \frac{k-1}{2k-1}T$. So, using this, we have

$$\frac{v_1(A_2)}{v_1(A_1)} = \frac{T - v_1(A_1)}{v_1(A_1)} \leq \frac{k}{k-1} \leq 2. \tag{5.6}$$

ii) $|A_2| = 1$**:** Since $|A_2| = 1$ and from above we know that $S_1 \subseteq A_2$, we have that $S_1 = A_2$. This in turn implies that rank-leximin$(v_1, v_2) = (A_1, A_2)$ and rank-leximin$(v_1', v_2) = (A_2, A_1)$. Next, consider the following observations.

   a) $r_1(A_1) = 2^{m-1}$ (recall that this is the case we are considering)

   b) $r_2(A_2) = 2^{m-1} + 1$ (since $|A_2| = 1$, we know that $r_2(A_2) \leq 2^{m-1} + 1$, because valuation functions are additive and so any singleton set can have rank at most $2^{m-1} + 1$. However, $r_2(A_2) \geq 2^{m-1} + 1$, using Claim 11.)

   c) $r_1'(A_1) = 2^{m-1}$ (since $|A_1| = m - 1$, we know that $r_1'(A_1) \geq 2^{m-1}$. However, $r_1'(A_1) < 2^{m-1} + 1$, for if not then $(A_1, A_2) \succ_{v_1', v_2} (A_2, A_1)$, thus contradicting the fact that rank-leximin$(v_1', v_2) = (A_2, A_1)$)

   d) $r_1(A_2) = 2^{m-1} + 1$ (since $|A_2| = 1 \Rightarrow r_1(A_2) \leq 2^{m-1} + 1$, and $r_1(A_2) > r_1(A_1)$ since we are considering the case when $v_1(S_1) = v_1(A_2) > v_1(A_1)$)

   e) $r_2(A_1) = 2^{m-1}$ (since $|A_1| = m - 1 \Rightarrow r_2(A_1) \geq 2^{m-1}$, and $r_2(A_1) < 2^{m-1} + 1$, for if not then $(A_2, A_1) \succ_{v_1, v_2} (A_1, A_2)$, thus contradicting the fact that rank-leximin$(v_1, v_2) = (A_1, A_2)$)

f) $r'_1(A_2) = 2^{m-1} + 1$ (since $|A_2| = 1 \Rightarrow r'_1(A_2) \leq 2^{m-1} + 1$, and $r_2(A_1) = 2^{m-1} \Rightarrow r'_1(A_2) \geq 2^{m-1} + 1$ using Claim 11)

Now, combining all the observations above, we can see that $(A_1, A_2) \equiv_{v_1,v_2} (A_2, A_1)$ and $(A_1, A_2) \equiv_{v'_1,v_2} (A_2, A_1)$. However, note that this in turn is a contradiction because the rank-leximin algorithm breaks ties deterministically and so it couldn't have been the case that rank-leximin$(v_1, v_2) = (A_1, A_2)$ and rank-leximin$(v'_1, v_2) = (A_2, A_1)$. Hence, this case is impossible.

Case 3. $r_1(A_1) = 2^{m-1}$ **and** $|S_1| \geq 2$**:** First, note that since $v_1(S_1) > v_1(A_1)$, $r_1(A_1) = 2^{m-1}$, and rank-leximin$(v_1, v_2) = (A_1, A_2)$, we have that, for all $g \in S_1$, $v_1(S_1 \setminus \{g\}) \leq v_1(A_1)$. Why? Suppose if not, then consider the allocation $(S_1 \setminus \{g\}), S_2 \cup \{g\})$, and observe that $(S_1 \setminus \{g\}), S_2 \cup \{g\}) \succ_{v_1,v_2} (A_1, A_2)$, which in turn contradicts the fact that rank-leximin$(v_1, v_2) = (A_1, A_2)$. So, now, by adding up all the inequalities with respect all $g \in S_1$, we have that

$$(|S_1| - 1) \cdot v_1(S_1) \leq |S_1| \cdot v_1(A_1).$$

This implies that we can use the fact that $|S_1| \geq 2$ to see that

$$\frac{v_1(S_1)}{v_1(A_1)} \leq \frac{|S_1|}{|S_1| - 1} \leq 2. \tag{5.7}$$

Finally, combining (5.4), (5.6), and (5.7), we have our lemma. $\square$

Next, we show a lower bound for $\frac{v_1(S_1)}{v_1(A_1)}$.

Lemma 39. *For $\alpha \in (0, \frac{T}{3}]$, if $v'_1 \in \alpha\text{-}N(v_1)$, rank-leximin$(v_1, v_2) = (A_1, A_2)$, and rank-leximin$(v'_1, v_2) = (S_1, S_2)$, then*

$$\frac{v_1(S_1)}{v_1(A_1)} \geq \frac{1}{2 + \frac{12\alpha}{T}}.$$

Proof. Since we are trying to show a lower bound for $\frac{v_1(S_1)}{v_1(A_1)}$, below we consider the case when $v_1(A_1) > v_1(S_1)$, since otherwise it is trivially lower-bounded by 1. Also, note we can directly consider the case when $v_1(S_2) > v_1(S_1)$, for otherwise $v_1(S_1) = T - v_1(S_2) \geq \frac{T}{2}$ and so $\frac{v_1(S_1)}{v_1(A_1)} \geq \frac{1}{2}$. Therefore, throughout this proof, we have $v_1(S_2) > \frac{T}{2}$.

Next, let us consider the following cases. Note that, by using Claim 11 with respect to the allocation (S_1, S_2), these are the only three cases.

Case 1. $r_1'(S_1) > 2^{m-1}$**:** Since $r_1'(S_1) > 2^{m-1}$, we know from Claim 8 that $v_1'(S_1) \geq \frac{T}{2}$. Also, since $v_1' \in \alpha\text{-}N(v_1)$, from Claim 9(i) we have that $v_1'(S_1) - v_1(S_1) \leq |v_1'(S_1) - v_1(S_1)| \leq \frac{\alpha}{2}$. So, using these, we have,

$$\frac{v_1(S_1)}{v_1(A_1)} \geq \frac{\frac{T}{2} - \frac{\alpha}{2}}{v_1(A_1)} \geq \frac{\frac{T}{2} - \frac{\alpha}{2}}{T} \geq \frac{1}{2 + \frac{4\alpha}{T}}, \tag{5.8}$$

where the last inequality follows from the fact that $\alpha \in (0, \frac{T}{3}]$.

Case 2. $r_1'(S_1) = 2^{m-1}$ **and** $\min\{r_1(A_1), r_2(A_2)\} = r_1(A_1)$**:** Since $\min\{r_1(A_1), r_2(A_2)\} = r_1(A_1)$ and rank-leximin$(v_1', v_2) = (S_1, S_2)$, we have that $v_1'(A_1) \leq v_1'(S_1)$, for if not, then it is easy to see that $(A_1, A_2) \succ_{(v_1', v_2)} (S_1, S_2)$, thus contradicting the fact that rank-leximin$(v_1', v_2) = (S_1, S_2)$. Also, since $v_1(A_1) > v_1(S_1)$ and from above we have $v_1'(A_1) \leq v_1'(S_1)$, from Claim 9(ii) we have know that $v_1(A_1) - v_1(S_1) \leq \alpha$. So, now, let us consider the following two cases.

i) $|S_2| = 1$**:** First, note that since $|S_2| = 1$ and $v_1(A_1) > v_1(S_1)$, we have that $S_2 \subseteq A_1$. This in turn implies that $v_1(A_1) \geq v_1(S_2) > \frac{T}{2}$. So, now, using these we have,

$$\frac{v_1(S_1)}{v_1(A_1)} \geq \frac{v_1(A_1) - \alpha}{v_1(A_1)} > 1 - \frac{2\alpha}{T} \geq \frac{1}{2 + \frac{4\alpha}{T}}, \tag{5.9}$$

where the last inequality follows from the fact that $\alpha \in (0, \frac{T}{3}]$.

ii) $|S_2| \geq 2$**:** Let $k = |S_2|$. Now, since $v_1(A_1) - v_1(S_1) \leq \alpha$, we have,

$$\begin{aligned}
\frac{v_1(S_1)}{v_1(A_1)} \geq \frac{v_1(S_1)}{v_1(S_1) + \alpha} &= 1 - \frac{\alpha}{v_1(S_1) + \alpha} \\
&\geq 1 - \frac{\alpha}{v_1'(S_1) - \frac{\alpha}{2} + \alpha} && \text{(using Claim 9(i))} \\
&\geq 1 - \frac{\alpha}{\frac{k-1}{2k-1}T + \frac{\alpha}{2}} && \text{(using Claim 10(iii))} \\
&\geq \frac{1}{2 + \frac{4\alpha}{T}},
\end{aligned} \tag{5.10}$$

where the last inequality follows from the fact that $\alpha \in (0, \frac{T}{3}]$.

Case 3. $r_1'(S_1) = 2^{m-1}$ **and** $\min\{r_1(A_1), r_2(A_2)\} = r_2(A_2)$**:** First, note that we can directly consider the case when $\min\{r_1(A_1), r_2(A_2)\} = r_2(A_2) = 2^{m-1}$, for if not then we either have $v'_1(A_1) \leq v'_1(S_1)$, which can be handled as in Case 2, or we have that $v'_1(A_1) > v'_1(S_1)$, which is impossible since this would imply $\min\{r'_1(A_1), r_2(A_2)\} > 2^{m-1} = r'_1(S_1) = \min\{r'_1(S_1), r_2(S_2)\}$, thus contradicting the fact that rank-leximin$(v'_1, v_2) = (S_1, S_2)$. Additionally, since we are considering the case when $r_2(A_2) = 2^{m-1}$, we have from Claim 11 that $r_1(A_1) \geq 2^{m-1} + 1$, which in turn implies, using Claim 8, that $v_1(A_1) \geq \frac{T}{2}$.

Given the observations above, let us now consider the following cases.

i) $|S_2| \geq 2$ **and** $|S_1 \cap A_1| \geq 1$**:** Let $k = |S_2|$ and let $X \subset \mathcal{G}$ such that $r_1(X) = 2^{m-1}$. Since rank-leximin$(v_1, v_2) = (A_1, A_2)$ and $\forall g \in \mathcal{G}$, and $\forall i \in [2]$, $v_i(g) > 0$, we have that $r_1(A_1 \setminus \{g\}) \leq 2^{m-1}$, for if not then $\min\{r_1(A_1 \setminus \{g\}), r_2(A_2 \cup \{g\})\} > 2^{m-1} = r_2(A_2) = \min\{r_1(A_1), r_2(A_2)\}$, thus contradicting the fact that rank-leximin$(v_1, v_2) = (A_1, A_2)$. So, next, let us consider a good $g \in S_1 \cap A_1$. Using the observations above, we have,

$$\begin{aligned} v_1(A_1) &\leq v_1(X) + v_1(g) && \text{(since } r_1(A_1 \setminus \{g\}) \leq 2^{m-1}\text{)} \\ &\leq v_1(X) + v_1(S_1) && \text{(since } v_1(g) \leq v_1(S_1)\text{)} \\ &\leq 2v_1(S_1) + \alpha. && (5.11) \end{aligned}$$

The last inequality is true because either $v_1(S_1) \geq v_1(X)$, or, if not, then it is easy to see that, since $r'_1(S_1) = 2^{m-1}$ and $r_1(S_1) < r_1(X_1) = 2^{m-1}$, there exists an $H \subset \mathcal{G}$ such that $v_1(H) \geq v_1(X) > v_1(S_1)$ and $v'_1(H) \leq v'_1(S_1)$. This in turn implies, we can use Claim 9(ii), to see that $v_1(H) - v_1(S) \leq \alpha$.

So, now, using (5.11), we have,

$$\begin{aligned} \frac{v_1(S_1)}{v_1(A_1)} \geq \frac{1}{2} - \frac{\alpha}{2v_1(A_1)} &\geq \frac{1}{2} - \frac{\alpha}{T} && \left(\text{since } v_1(A_1) \geq \frac{T}{2}\right) \\ &\geq \frac{1}{2 + \frac{12\alpha}{T}}, && (5.12) \end{aligned}$$

where the last inequality follows from the fact that $\alpha \in (0, \frac{T}{3}]$.

ii) $|S_2| \geq 2$ **and** $|S_1 \cap A_1| = 0$**:** Let $k = |S_2|$. First, note that since $|S_1 \cap A_1| = 0$, we have that $A_1 \subseteq S_2$. Therefore, using this, we have,

$$\begin{aligned} \frac{v_1(S_1)}{v_1(A_1)} \geq \frac{v_1(S_1)}{v_1(S_2)} &\geq \frac{v'_1(S_1) - \frac{\alpha}{2}}{v'_1(S_2) + \frac{\alpha}{2}} \\ &\geq \frac{\frac{k-1}{2k-1}T - \frac{\alpha}{2}}{\frac{k}{2k-1}T + \frac{\alpha}{2}} && \text{(using Claim 10(iii))} \end{aligned}$$

$$\geq \frac{1}{2 + \frac{12\alpha}{T}}, \tag{5.13}$$

where the last inequality follows from the fact that $k \geq 2$ and $\alpha \in (0, \frac{T}{3}]$.

iii) $|S_2| = 1$**:** First, note that since $|S_2| = 1$, $r_2(S_2) \leq 2^{m-1} + 1$, as additivity implies that any singleton set can have rank at most $2^{m-1} + 1$. Also, recall that we are considering the case when $r'_1(S_1) = r_2(A_2) = 2^{m-1}$. Now, since rank-lexmin$(v'_1, v_2) = (S_1, S_2)$, we need to have $v'_1(S_1) \geq v'_1(A_1)$, for if not, then the fact that $r_2(A_2) = 2^{m-1}$ and $v'_1(A_1) > v'_1(S_1)$ implies that $(A_1, A_2) \succeq_{v_1, v_2} (S_1, S_2)$ and $(A_1, A_2) \succeq_{v'_1, v_2} (S_1, S_2)$. However, this is impossible since the rank-leximin algorithm uses a deterministic tie-breaking rule and hence if this is case, then rank-leximin(v_1, v_2) and rank-leximin(v'_1, v_2) cannot be different. Therefore, we have that $v'_1(S_1) \geq v'_1(A_1)$, which in turn can again be handled as in Case 2.

Finally, combining (5.8)–(5.13), we have that $\frac{v_1(S_1)}{v_1(A_1} \geq \frac{1}{2+\frac{12\alpha}{T}}$. □

Since rank-leximin is stable with respect to equivalent instances, and combining the lemmas above we have that it is $(2 + \frac{12\alpha}{T})$-weakly-approximately-stable for all $\alpha \leq \frac{T}{3}$, our theorem follows. □

Note that for $\alpha \in (0, \frac{T}{3}]$, our approximate-stability bound is independent of the choice of T, whereas for all the previously studied algorithms that we had considered in Section 5.4.1 it was dependent on T, and in fact was equal to $(T-1)$ even when $\alpha = 4$. Additionally, none of those algorithms are stable on equivalent instances either (i.e., the bound of $(T-1)$ was with respect to weak-approximate-stability and not the stronger notion of approximate-stability). Finally, it is interesting to note that although our model assumes that the misreport v' always maintains the ordinal information (over the singletons) of the true valuation v (see the second condition in Definition 18), the proof above does not use this fact. This implies that rank-leximin is stronger than what the theorem indicates since it is $(2 + \frac{12\alpha}{T})$-stable for all $\alpha \leq \frac{T}{3}$ and for all possible misreports that are within an L_1 distance of α from the original valuation.[6]

[6]Seeing this, one might wonder "why have the condition of maintaining the ordinal information over the singletons (i.e., the second condition in Definition 18) then?" The reason is, recall that our original goal was look for stable algorithms, and so this condition was a natural one to impose since, as mentioned in the introduction, one cannot hope for stable algorithms without assuming anything about the kind of mistakes the agents might commit. However, as mentioned in Section 5.4.2, even with this assumption designing stable algorithms turned out to be impossible, and hence we relaxed the strong requirement of stability to approximate-stability. And under this relaxed definition, the condition turns out to be not as crucial.

Input: for each agent $i \in [2]$, their valuation function v_i
Output: an allocation $A = (A_1, A_2)$ that satisfies the same properties as $\mathcal{M}$

1: $g_1^{\max} \leftarrow \arg\max_{g \in \mathcal{G}} v_1(g)$ ▷ in case of tie, pick the one with the lowest index
2: **if** $v_1(g_1^{\max}) > \frac{T}{2}$ **then** ▷ **Phase1**
3: **return** $A = (\{g_1^{\max}\}, \mathcal{G} \setminus \{g_1^{\max}\})$
4: **end if**
5: **return** $\mathcal{M}(v_1, v_2)$. ▷ **Phase 2**

Algorithm 8: Framework for modified-$\mathcal{M}$

5.5 Weak-Approximate-Stability in Fair Allocation of Indivisible Goods

In this section we consider weak-approximate-stability, the weaker notion of approximate stability defined in Definition 21 and we show how a very simple change to the existing algorithms can ensure more stability. In particular, we show that any algorithm that returns a PMMS and PO allocation for two agents can be made $(4 + \frac{6\alpha}{T})$-weakly-approximately-stable for $\alpha \leq \frac{T}{3}$. The main idea needed to achieve this is to modify the algorithm that produces a PMMS and PO allocation and convert it into a two-phase procedure where the first phase only handles instances where agent 1 has a good of value greater than $\frac{T}{2}$. In the case of such instances, we just output the allocation where agent 1 gets the highly-valued good and all the other goods are given to agent 2. Below, in Algorithm 8, we formally describe the framework and subsequently show how this very minor change leads to better stability overall.

Theorem 40. *Given an instance with m indivisible goods and two agents with additive valuation functions, let $\mathcal{M}$ be an algorithm that returns a PMMS and PO allocation. Then, for $\alpha \leq \frac{T}{3}$, modified-$\mathcal{M}$, which refers to the change as described in Algorithm 8, is $(4 + \frac{6\alpha}{T})$-weakly-approximately-stable, and returns an allocation that is PMMS and PO.*

Proof. We begin by arguing that if $\mathcal{M}$ produces a PMMS and PO allocation then so does modified-$\mathcal{M}$. To see this, note that we just need to show that the output in Phase 1 is PMMS and PO, since otherwise modified-$\mathcal{M}$ returns the same outcome as $\mathcal{M}$. The fact that it is PMMS is in turn easy to see because agent 1 is envy-free (since they get a good of value greater than $\frac{T}{2}$) and the other agent gets $m - 1$ goods. As for PO, note that agent 1 gets only one good with value greater than $\frac{T}{2}$, so they cannot be made better-off without making agent 2 worse-off.

To show the bound on weak-approximate-stability, let us consider two agents with utility functions v_1, v_2. Next, for some $\alpha \in (0, \frac{T}{3}]$, we denote the misreport by agent $i \in [2]$ as v'_i, and let $(-i)$ denote the agent $(i+1) \mod 2$. Also, let rank-leximin$(v_1, v_2) = (A_1, A_2)$, and, depending on which agent $i \in [2]$ is misreporting, let rank-leximin$(v'_1, v_2) = (S_1, S_2)$ or rank-leximin$(v_1, v'_2) = (S_1, S_2)$. We refer to (v_1, v_2) as the true report, and, depending on which agent $i \in [2]$ is misreporting, refer to (v'_1, v_2) or (v_1, v'_2) as the misreport. Given an instance, we say that the instance is in 'Phase 1' if agent 1 has a good of value greater than $\frac{T}{2}$ (this corresponds to the 'if' statement in Algorithm 8). Otherwise, we say that the instance is in 'Phase 2.' Also, throughout, for $i \in [2]$, we use $g_i^{\max}$ to denote the highest valued good in $\mathcal{G}$ according to agent i (if there are multiple goods with the highest value, pick the one with the lowest index). Recall that, since $v'_i \in \alpha\text{-}N(v_i)$, $g_i^{\max}$ is the same in v_i and v'_i.

Equipped with the notations above, let us now consider the following cases. Note that since the allocation returned is identical whenever an instance is in Phase 1, we only need to analyze these cases.

Case 1. True report is in Phase 1, misreport is in Phase 2: First, since there is a change in phase due to misreporting and the choice of phase only depends on agent 1's valuation function, note that agent 1 is the one misreporting. Given this, below we derive an upper and lower bound for the ratio $\frac{v_1(S_1)}{v_1(A_1)}$.

Note that since (v_1, v_2) in Phase 1, we have that $A_1 = \{g_1^{\max}\}$ and $v_1(g_1^{\max}) > \frac{T}{2}$. Therefore,

$$\frac{v_1(S_1)}{v_1(A_1)} = \frac{v_1(S_1)}{v_1(g_1^{\max})} \leq \frac{T}{v_1(g_1^{\max})} < 2. \tag{5.14}$$

Next, to lower bound the ratio, let us consider the following two cases.

i) $|S_2| = 1$**:** Since $|S_2| = 1$, we have that $v'_1(S_1) \geq v'_1(\mathcal{G} \setminus \{g_1^{\max}\})$. Also, since (v'_1, v_2) in Phase 2, we know that $v'_1(g_1^{\max}) \leq \frac{T}{2}$. So, now, using these and Claim 9(i), we have

$$\frac{v_1(S_1)}{v_1(A_1)} \geq \frac{v'_1(S_1) - \frac{\alpha}{2}}{v_1(A_1)} \geq \frac{v'_1(\mathcal{G} \setminus \{g_1^{\max}\}) - \frac{\alpha}{2}}{v_1(A_1)} \geq \frac{\frac{T}{2} - \frac{\alpha}{2}}{T} \geq \frac{1}{2 + \frac{4\alpha}{T}}, \tag{5.15}$$

where the last inequality follows since $\alpha \in (0, \frac{T}{3}]$.

ii) $|S_2| \geq 2$**:** Let $k = |S_2|$. Since $k \geq 2$, we can use Claim 9(i) and Claim 10(iii), to see that

$$\frac{v_1(S_1)}{v_1(A_1)} \geq \frac{v'_1(S_1) - \frac{\alpha}{2}}{v_1(A_1)} \geq \frac{\frac{k-1}{2k-1}T - \frac{\alpha}{2}}{T} \geq \frac{1}{4 + \frac{6\alpha}{T}}, \tag{5.16}$$

where the last inequality follows from the facts that $k \geq 2$ and $\alpha \in (0, \frac{T}{3}]$.

Case 2. True report is in Phase 2, misreport is in Phase 1: Similar to the previous case, here again we have that agent 1 is the one misreporting. Also, since (v_1', v_2) in Phase 1, we have that $S_1 = \{g_1^{\max}\}$ and $v_1'(g_1^{\max}) > \frac{T}{2}$. Just like in the previous case, we again upper and lower bound the ratio $\frac{v_1(S_1)}{v_1(A_1)}$.

For the upper bound, first note that since (v_1, v_2) is in Phase 2, we have $v_1(g_1^{\max}) \leq \frac{T}{2}$. Next, note that we only need to consider the case when $g_1^{\max} \in A_2$,since otherwise the ratio is upper-bounded by 1. Also, if $|A_2| = 1$, then note that $v_1(A_1) \geq T - v_1(g_1^{\max}) \geq \frac{T}{2}$, and so if this is the case then again the ratio is upper-bounded by 1. Finally, if $k = |A_2| \geq 2$, then we can use Claim 10(iii) to see that,

$$\frac{v_1(S_1)}{v_1(A_1)} = \frac{v_1(g_1^{\max})}{v_1(A_1)} \leq \frac{\frac{T}{2}}{\frac{k-1}{2k-1}T} \leq \frac{3}{2}. \tag{5.17}$$

For the lower bound, since $v_1'(g_1^{\max}) > \frac{T}{2}$ and $\alpha \in (0, \frac{T}{3}]$, we can use Claim 9(i) to see that,

$$\frac{v_1(S_1)}{v_1(A_1)} = \frac{v_1(g_1^{\max})}{v_1(A_1)} \geq \frac{\frac{T}{2} - \frac{\alpha}{2}}{T} \geq \frac{1}{2 + \frac{4\alpha}{T}}. \tag{5.18}$$

Case 3. True report is in Phase 2, misreport is in Phase 2: Let $i \in [2]$ be the agent who is misreporting. First, for the lower bound, we can proceed exactly as in Cases 1(i) and 1(ii) above. For the upper bound, let us consider the three cases.

i) $i = 1$ **and** $|A_2| = 1$**:** Since $|A_2| = 1$, we know that $v_1(A_1) \geq v_1(\mathcal{G} \setminus \{g_i^{\max}\})$. Also, since (v_1, v_2) is in Phase 2, $v_1(g_1^{\max}) \leq \frac{T}{2}$. So, using this, we have,

$$\frac{v_1(S_1)}{v_1(A_1)} \leq \frac{T}{T - v(g_1^{\max})} \leq 2. \tag{5.19}$$

ii) $i = 2$ **and** $|A_1| = 1$**:** First, note that $v_2(A_2) \geq \frac{T}{2}$, for, if otherwise, when the agents report (v_1, v_2), (A_2, A_1) Pareto dominates (A_1, A_2), which is a contradiction. This is because, since $|A_1| = 1$ and we are in Phase 2, $v_1(A_1) \leq \frac{T}{2}$, and so the allocations can be swapped to (weakly) improve the utilities of both the agents. Therefore, using this, we have,

$$\frac{v_2(S_2)}{v_2(A_2)} \leq \frac{v_2(S_2)}{\frac{T}{2}} \leq 2. \tag{5.20}$$

iii) $|A_{-i}| \geq 2$**:** Let $k = |A_{-i}|$. Since $k \geq 2$, we can use Claim 10(iii), to see that

$$\frac{v_i(S_i)}{v_i(A_i)} \leq \frac{T}{\frac{k-1}{2k-1}T} \leq 3. \tag{5.21}$$

Finally, combining all the observations, (5.14)–(5.21), from above we have our theorem. □

Note that the simple framework presented in Algorithm 8 can be used with respect to any algorithm $\mathcal{M}$ that is fair and efficient, and so one can derive, perhaps better bounds, when considering modified versions of specific algorithms (likes the ones in Section 5.4.1) that provide a PMMS, EFX, or EF1 guarantee. Therefore, although the overall change that is required to any algorithm is minor, the observation is important if we care about having algorithms that are more stable, since without this recall that we observed in Section 5.4.1 that all the four previously studied algorithms are $(T-1)$-weakly-approximately-stable even when $\alpha = 4$.

5.6 Discussion

The problem considered here was born as a result of an example we came across on the fair division website, Spliddit [GP15], where it seemed like arguably innocuous mistakes by an agent had significant ramifications on their utility for the outcome. Thus, the question arose as to whether such consequences were avoidable, or more broadly if we could have more stable algorithms in the context of fair allocations. In this chapter we focused on algorithmic stability in fair and efficient allocation of indivisible goods among two agents, and towards this end, we introduced a notion of stability and showed how it is impossible to achieve exact stability along with even a weak notion of fairness like EF1 and even approximate efficiency. This raised the question of how to relax the strong requirement of stability, and here we proposed two relaxations, namely, approximate-stability and weak-approximate-stability, and showed how the previously studied algorithms in fair division that are fair and efficient perform poorly with respect to these relaxations. This lead to looking for new algorithms, and here we proposed an approximately-stable algorithm (rank-leximin) that guarantees a pairwise maximin share and Pareto optimal allocation, and presented a simple general framework for any fair algorithm to achieve weak-approximate stability. Along the way, we also provided a characterization result for pairwise maximin share allocations and showed how (in addition to the two-agent case) such allocations always exist as long as the agents report ordinally equivalent valuation functions. Overall, while the results demonstrate how one can do better in the context of two-agent fair allocations, our main contribution is in introducing the notion of stability and its relaxations, and in explicitly advocating for it to be used in the design of algorithms or mechanisms that elicit cardinal preferences.

Moving forward, we believe that there is a lot of scope for future work. Especially when given the observation that humans may often find it hard to attribute exact numerical values to their preferences, we believe that some notion of stability should be considered when designing algorithms or mechanisms in settings where cardinal values are elicited. This opens up the possibility of asking similar questions like the ones we have in other settings where such issues may perhaps be more crucial. Additionally, specific to the problem considered here, there are several unanswered questions. For instance, the most obvious one is about the case when there are more than two agents. A careful reader would have noticed that all the algorithms we talked about here heavily relied on the fact that there were only two agents, and therefore none of them work when there are more. As another example—this one in the context of two-agent fair allocations—one question that could be interesting is to see if there are polynomial-time algorithms that are approximately-stable and can guarantee EF1/EFX and PO. Given the fact that polynomial time EF1 and PO algorithms exist for two agents, we believe that it would interesting to see if we can also additionally guarantee approximate-stability, or prove otherwise—in which case it would demonstrate a clear separation between unstable and approximately-stable algorithms. The rank-leximin algorithm presented here is approximately-stable and provides the stated guarantees in terms of fairness and efficiency, but it is exponential, and we could not answer this question either in the positive or otherwise. Therefore, this, too, remains to be resolved by future work.

Chapter 6

Conclusion

The focus of this thesis was on collective decision-making settings where the agents interacting with the concerned decision-making procedure do not provide complete or accurate preference information. In particular, we looked at four settings, each of them highlighting different aspects of working with incomplete or inaccurate preference information. We provide a brief summary of our contributions below and connect them with the goals mentioned in Chapter 1.

The first two technical chapters (Chapters 2 and 3) of the thesis dealt with the question of designing robust algorithms or mechanisms in settings where agents have incomplete preferences. Here by robust we meant algorithms or mechanisms that produce solutions that are good (for specific definitions of good) with respect to all the possible underlying complete preferences. Chapter 2 dealt with what can perhaps be considered as the most basic setting when designing robust algorithms in the presence of incomplete preferences. Here we looked at a version of the two-sided matching problem where agents are allowed to submit weak orders and the objective was to design algorithms that find matchings that minimize the maximum number of blocking pairs with respect to all the possible underlying linear orders. To this end, we provided some approximation algorithms and some hardness of approximation results in different settings for this problem. Although the issue of incomplete information had been previously studied in the context of this problem, the usual approach had been to work with extremes—i.e., to look at matchings that have no blocking pairs with at least one, or all, of the underlying linear orders. Our work here can be considered as an extension to this line of work, since here we moved away from these extremes and tried to find a middle-ground when it comes to working with incomplete preference information.

Continuing with theme of designing robust algorithms or mechanisms, in Chapter 3 we looked at a version of the facility location problem which allowed us to focus on incentive issues that arise when working with incomplete preference information. In contrast to the standard model for this problem, here we introduced a model where the agents only provide coarse preference information to the mechanism—in particular, agents only indicate that their preferred location for the facility lies in some interval. This in turn raised the question on what solution concepts to consider for such a setting, and here we looked at two natural solution concepts, namely, very weak dominance and minimax dominance. The first one was previously considered in the context of auction design and the other introduced here. Following this, we designed mechanisms that, under each of the two solution concepts, approximately implement the optimal minimax value with respect to two commonly studied objective functions, and also showed some corresponding lower bounds.

In Chapter 4 we looked at a different aspect when it comes to working with incomplete preference information. While Chapters 2 and 3 considered the case where algorithms just use the given incomplete information, here we considered a scenario where the algorithm can query the agents to get more information. Although, as mentioned in the chapter, there has been other work that have considered similar settings, our work here complemented these by looking at a different choice of objective to optimize for and by considering an alternative type of query the algorithm could use. This line of inquiry was, in turn, motivated by some applications in the context of school choice—the problem of assigning students to public schools. In particular, in this chapter, we considered one-sided matching problems where the objective is to find an assignment of agents to objects that satisfies some desirable property like Pareto optimality. While most work on this problem assumes that agents only have ordinal preferences, here we considered a setting where agents may have some underlying cardinal utilities. We first argued why not taking these into account can result in significant loss in welfare and subsequently designed algorithms that can query the agents using threshold queries about their cardinal utilities. Overall, our results showed how even asking the agents for one bit of extra information per object can result in better social welfare.

Finally, motivated by some observations on a popular fair division website, in Chapter 5 we considered a scenario where the agents are unsophisticated in the sense that they are prone to making mistakes while reporting their preferences. In particular, we considered the problem of fairly allocating a set of indivisible goods among two agents, and introduced a notion of algorithmic stability which, informally, states that the utility experienced by an agent should not change much as long as they make of "small" or "innocuous" mistakes while reporting their preferences. We first showed that it impossible for algorithms to be fair, Pareto optimal, and stable. Following this, we focused on two relaxations of stability, namely, approximate-stability and weak-approximate-stability, and showed how the rank-leximin algorithm satisfies the maximin share fairness property, is Pareto optimal, and approximately-stable. Following this, we also presented a simple framework to modify existing fair and Pareto optimal algorithms in order to ensure that they also achieve weak-approximate-stability. Although the results in this chapter work only for the limited case of two agent fair allocation, we believe that our main contribution here was in highlighting the issue of stability and in explicitly advocating for it be considered in the design of algorithms, especially ones which ask the agents for cardinal preference information.

6.1 Future Directions

In Chapters 2–5 we discussed future work in the context of specific questions considered in those chapters. Here we talk about some broad directions that could potentially be interesting.

Beyond worst-case analysis. All the results in this thesis are with respect to absolute worst-case notions where we try to find a desirable outcome (e.g., a matching with the least number of blocking pairs, or one with maximum welfare) and the bounds presented are with respect to all possible instances and all possible underlying complete information that is consistent with the given partial information. While we believe that this is a reasonable first step, it might be the case that these worst-cases almost never occur in most applications we care about. Hence, it would be interesting to see if one can get better algorithms by assuming certain structure—ones which are ideally motivated from real applications—on the kinds of input an algorithm will face. For instance, suppose there is reason to believe that agents' preferences will be correlated. In this case, one question that might potentially be interesting is to see if we can define a suitable parameter to describes how correlated they are and then come up with algorithms that achieve better bounds for different values of this parameter. More broadly, it would be interesting to see if, based on some properties, we can classify "plausible inputs" (i.e., the kinds of input an algorithm is likely to encounter in an application) into certain classes and then see if it is possible to design class-specific algorithms that do well on instances from the corresponding class.

Combining theoretical analysis with experiments. The focus of this thesis was on theoretical analysis of different settings. While such analysis is useful in proving that the algorithms presented guarantee certain outcomes, there might still be challenges in realising these guarantees if one we were to actually deploy these in some application. For instance, while the theoretical analysis from Chapter 4 says that one can improve welfare by even just asking one bit of extra information per object, the algorithms presented there ask for very specific threshold queries which might perhaps be hard to answer in practice. Therefore, it would be interesting to study, for example, what kind of queries are easier to answer and the types of interface-design that can best support queries. More broadly, we believe that it would be interesting to start with a theoretical model and analysis, run experiments to see the challenges in deploying these, or the challenges in realising these guarantees, and then tweak the model and analysis based on such observations.

Randomization. All the results in this thesis are on deterministic algorithms or mechanisms. Since all the problems considered here were ones that were not previously studied, it was natural as a first step to understand the limits of determinism, as determinism can in fact be a hard constraint in many applications. However, there might be settings where randomization is allowed and where in fact it might be beneficial in terms of getting better outcomes. Therefore, it would be interesting to extend all the results here to this case, and in the process perhaps get much better bounds than the ones presented here.

References

[AA18] Ben Abramowitz and Elliot Anshelevich. "Utilitarians Without Utilities: Maximizing Social Welfare for Graph Problems Using Only Ordinal Preferences". In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*. 2018, pp. 894–901 (cit. on pp. 75, 78, 104, 113).

[AAZ19] Ben Abramowitz, Elliot Anshelevich, and Wennan Zhu. "Awareness of Voter Passion Greatly Improves the Distortion of Metric Social Choice". In: *Proceedings of the Fifteenth International Conference on Web and Internet Economics (WINE)*. 2019, pp. 3–16 (cit. on p. 78).

[Abd+05] Atila Abdulkadiroğlu, Parag A Pathak, Alvin E Roth, and Tayfun Sönmez. "The Boston Public School Match". In: *American Economic Review* 95.2 (2005), pp. 368–371 (cit. on p. 1).

[ABM05] David J. Abraham, Péter Biró, and David Manlove. ""Almost stable" Matchings in the Roommates Problem". In: *Proceedings of the Third International Workshop on Approximation and Online Algorithms (WAOA)*. 2005, pp. 1–14 (cit. on p. 13).

[Abr+04] David J. Abraham, Katarína Cechlárová, David F Manlove, and Kurt Mehlhorn. "Pareto Optimality in House Allocation Problems". In: *Proceedings of the Fifteenth International Symposium on Algorithms and Computation (ISAAC)*. 2004, pp. 3–15 (cit. on pp. 73, 78, 80).

[Abr+06] David J. Abraham, Ning Chen, Vijay Kumar, and Vahab S Mirrokni. "Assignment Problems in Rental Markets". In: *Proceedings of the Second International Workshop on Internet and Network Economics (WINE)*. 2006, pp. 198–213 (cit. on pp. 78, 80).

[Abr+07] David J Abraham, Robert W Irving, Telikepalli Kavitha, and Kurt Mehlhorn. "Popular Matchings". In: *SIAM Journal on Computing* 37.4 (2007), pp. 1030–1045 (cit. on p. 100).

[ACY11] Atila Abdulkadiroğlu, Yeon-Koo Che, and Yosuke Yasuda. "Resolving Conflicting Preferences in School Choice: The "Boston Mechanism" Reconsidered". In: *American Economic Review* 101.1 (2011), pp. 399–410 (cit. on pp. 74, 78).

[ACY15] Atila Abdulkadiroğlu, Yeon-Koo Che, and Yosuke Yasuda. "Expanding "Choice" in School Choice". In: *American Economic Journal: Microeconomics* 7.1 (2015), pp. 1–42 (cit. on pp. 74, 78).

[Ama+17] Georgios Amanatidis, Georgios Birmpas, George Christodoulou, and Vangelis Mar-kakis. "Truthful Allocation Mechanisms Without Payments: Characterization and Implications on Fairness". In: *Proceedings of the Eighteenth ACM Conference on Economics and Computation (EC)*. 2017, pp. 545–562 (cit. on p. 112).

[Ama+20] Georgios Amanatidis, Georgios Birmpas, Aris Filos-Ratsikas, and Alexandros A. Voudouris. "Peeking Behind the Ordinal Curtain: Improving Distortion via Cardinal Queries". In: *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*. 2020, pp. 1782–1789 (cit. on p. 78).

[Ama+21] Georgios Amanatidis, Georgios Birmpas, Aris Filos-Ratsikas, and Alexandros A. Voudouris. "A Few Queries Go a Long Way: Information-Distortion Tradeoffs in Matching". In: *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*. 2021 (cit. on pp. 75, 78, 89, 180).

[APR05] Atila Abdulkadiroğlu, Parag A Pathak, and Alvin E Roth. "The New York City High School Match". In: *American Economic Review* 95.2 (2005), pp. 364–367 (cit. on p. 1).

[AS03] Atila Abdulkadiroğlu and Tayfun Sönmez. "School choice: A Mechanism Design Approach". In: *American economic review* 93.3 (2003), pp. 729–747 (cit. on p. 1).

[AS16] Elliot Anshelevich and Shreyas Sekar. "Blind, Greedy, and Random: Algorithms for Matching and Clustering Using Only Ordinal Information". In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI)*. 2016, pp. 383–389 (cit. on pp. 75, 78, 84, 104, 113).

[AS98] Atila Abdulkadiroğlu and Tayfun Sönmez. "Random Serial Dictatorship and the Core From Random Endowments in House Allocation Problems". In: *Econometrica* 66.3 (1998), pp. 689–701 (cit. on pp. 73, 78).

[AS99] Atila Abdulkadiroğlu and Tayfun Sönmez. "House Allocation With Existing Tenants". In: *Journal of Economic Theory* 88.2 (1999), pp. 233–260 (cit. on pp. 73, 78).

[AWX15] Haris Aziz, Toby Walsh, and Lirong Xia. "Possible and necessary allocations via sequential mechanisms". In: *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*. 2015, pp. 468–474 (cit. on p. 12).

[AZ17] Elliot Anshelevich and Wennan Zhu. "Tradeoffs Between Information and Ordinal Approximation for Bipartite Matching". In: *Proceedings of the Tenth International Symposium on Algorithmic Game Theory (SAGT)*. 2017, pp. 267–279 (cit. on pp. 75, 78, 104, 113).

[Azi+15] Haris Aziz, Serge Gaspers, Simon Mackenzie, and Toby Walsh. "Fair Assignment of Indivisible Objects Under Ordinal Preferences". In: *Artificial Intelligence* 227 (2015), pp. 71–92 (cit. on pp. 104, 114).

[Azi+16] Haris Aziz, Péter Biró, Serge Gaspers, Ronald de Haan, Nicholas Mattei, and Baharak Rastegari. "Stable Matching With Uncertain Linear Preferences". In: *Proceedings of the Ninth International Symposium on Algorithmic Game Theory (SAGT)*. 2016, pp. 195–206 (cit. on p. 16).

[Azi+19] Haris Aziz, Péter Biró, Jérôme Lang, Julien Lesca, and Jérôme Monnot. "Efficient reallocation under additive and responsive preferences". In: *Theoretical Computer Science* 790 (2019), pp. 1–15 (cit. on p. 12).

[Azi15] Haris Aziz. "A Note on the Undercut Procedure". In: *Social Choice and Welfare* 45.4 (Dec. 2015), pp. 723–728 (cit. on p. 114).

[Bar01] Salvador Barberà. "An Introduction to Strategy-Proof Social Choice Functions". In: *Social Choice and Welfare* 18.4 (2001), pp. 619–653 (cit. on p. 50).

[Bau+12] Dorothea Baumeister, Piotr Faliszewski, Jérôme Lang, and Jörg Rothe. "Campaigns for Lazy Voters: Truncated Ballots". In: *Proceedings of the Eleventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 2012, pp. 577–584 (cit. on pp. 12, 26).

[BB04] Mourad Baïou and Michel Balinski. "Student Admissions and Faculty Recruitment". In: *Theoretical Computer Science* 322.2 (2004), pp. 245–265 (cit. on p. 16).

[BE02] Olivier Bousquet and André Elisseeff. "Stability and Generalization". In: *Journal of Machine Learning Research (JMLR)* 2.Mar (2002), pp. 499–526 (cit. on p. 112).

[BEL10] Sylvain Bouveret, Ulle Endriss, and Jérôme Lang. "Fair Division Under Ordinal Preferences: Computing Envy-Free Allocations of Indivisible Goods". In: *Proceedings of the Nineteenth European Conference on Artificial Intelligence (ECAI)*. 2010, pp. 387–392 (cit. on pp. 104, 114).

[Ben+20] Gerdus Benade, Swaprava Nath, Ariel D Procaccia, and Nisarg Shah. "Preference Elicitation for Participatory Budgeting". In: *Management Science, Articles in Advance* (2020), pp. 1–15 (cit. on p. 78).

[BJ94] Salvador Barberà and Matthew Jackson. "A Characterization of Strategy-Proof Social Choice Functions for Economies With Pure Public Goods". In: *Social Choice and Welfare* 11.3 (1994), pp. 241–252 (cit. on p. 50).

[BKK12] Steven J Brams, D Marc Kilgour, and Christian Klamler. "The Undercut Procedure: An Algorithm for the Envy-Free Division of Indivisible Items". In: *Social Choice and Welfare* 39.2-3 (2012), pp. 615–631 (cit. on p. 114).

[BKK14] Steven J Brams, M Kilgour, and Christian Klamler. "Two-Person Fair Division of Indivisible Items: An Efficient, Envy-Free Algorithm". In: *Notices of the AMS* 61.2 (2014), pp. 130–141 (cit. on pp. 104, 114).

[BKV18] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. "Finding Fair and Efficient Allocations". In: *Proceedings of the Nineteenth ACM Conference on Economics and Computation (EC)*. 2018, pp. 557–574 (cit. on pp. 114–115, 186).

[BM01] Anna Bogomolnaia and Hervé Moulin. "A New Solution to the Random Assignment Problem". In: *Journal of Economic Theory* 100.2 (2001), pp. 295–328 (cit. on p. 104).

[BMM10] Péter Biró, David Manlove, and Shubham Mittal. "Size versus Stability in the Marriage Problem". In: *Theoretical Computer Science* 411.16-18 (2010), pp. 1828–1841 (cit. on p. 13).

[Bou+15] Craig Boutilier, Ioannis Caragiannis, Simi Haber, Tyler Lu, Ariel D Procaccia, and Or Sheffet. "Optimal Social Choice Functions: A Utilitarian View". In: *Artificial Intelligence* 227 (2015), pp. 190–213 (cit. on pp. 75, 78, 104, 113).

[Bre+17] Robert Bredereck, Piotr Faliszewski, Andrzej Kaczmarczyk, Rolf Niedermeier, Piotr Skowron, and Nimrod Talmon. "Robustness Among Multiwinner Voting Rules". In: *Proceedings of the Tenth International Symposium on Algorithmic Game Theory (SAGT)*. 2017, pp. 80–92 (cit. on pp. 78, 113).

[BT96a] Steven J Brams and Alan D Taylor. "A Procedure for Divorce Settlements". In: *Mediation Quarterly* 13.3 (1996), pp. 191–205 (cit. on pp. 115, 185).

[BT96b] Steven J Brams and Alan D Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, 1996 (cit. on pp. 102, 115, 185).

[Bud+13] Eric Budish, Yeon-Koo Che, Fuhito Kojima, and Paul Milgrom. "Designing Random Allocation Mechanisms: Theory and Applications". In: *American Economic Review* 103.2 (2013), pp. 585–623 (cit. on p. 104).

[Bud11] Eric Budish. "The Combinatorial Assignment Problem: Approximate Competitive Equilibrium from Equal Incomes". In: *Journal of Political Economy* 119.6 (2011), pp. 1061–1103 (cit. on p. 114).

[CAR] CARMS: *Canadian Resident Matching Service*. URL: https://www.carms.ca/ (visited on 06/22/2021) (cit. on pp. 1, 16).

[Car+16] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D Procaccia, Nisarg Shah, and Junxing Wang. "The Unreasonable Fairness of Maximum Nash Welfare". In: *Proceedings of the Seventeenth ACM Conference on Economics and Computation (EC)*. 2016, pp. 305–322 (cit. on pp. 103–104, 109, 112, 114–116, 186).

[CCP19] Ioannis Caragiannis, George Christodoulou, and Nicos Protopapas. "Impartial selection with additive approximation guarantees". In: *Proceedings of the Twelfth International Symposium on Algorithmic Game Theory (SAGT)*. Springer. 2019, pp. 269–283 (cit. on p. 48).

[CFT16] Qingpeng Cai, Aris Filos-Ratsikas, and Pingzhong Tang. "Facility Location With Minimax Envy". In: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*. 2016, pp. 137–143 (cit. on p. 42).

[CGH19] Ioannis Caragiannis, Nick Gravin, and Xin Huang. "Envy-freeness up to Any Item with High Nash Welfare: The Virtue of Donating Items". In: *Proceedings of the Twentieth ACM Conference on Economics and Computation (EC)*. 2019, pp. 527–545 (cit. on p. 114).

[CMA12] Alessandro Chiesa, Silvio Micali, and Zeyuan Allen-Zhu. "Mechanism Design with Approximate Valuations". In: *Proceedings of the Third Innovations in Theoretical Computer Science (ITCS)*. 2012, pp. 34–38 (cit. on pp. 13, 45, 47, 49, 54, 78, 113).

[CMA14] Alessandro Chiesa, Silvio Micali, and Zeyuan Allen-Zhu. "Knightian Self Uncertainty in the VCG Mechanism for Unrestricted Combinatorial Auctions". In: *Proceedings of the Fifteenth ACM Conference on Economics and Computation (EC)*. 2014, pp. 619–620 (cit. on pp. 49, 53, 78, 113).

[CMA15] Alessandro Chiesa, Silvio Micali, and Zeyuan Allen-Zhu. "Knightian Analysis of the Vickrey Mechanism". In: *Econometrica* 83.5 (2015), pp. 1727–1754. ISSN: 1468-0262 (cit. on pp. 47–49, 113).

[Con85] Marquis De Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. L'imprimerie royale, 1785 (cit. on p. 1).

[CSS19] Jiehua Chen, Piotr Skowron, and Manuel Sorge. "Matchings Under Preferences: StrenGth of Stability and Trade-Offs". In: *Proceedings of the Twentieth ACM Conference on Economics and Computation (EC)*. 2019, pp. 41–59 (cit. on pp. 78, 113).

[DB13] Joanna Drummond and Craig Boutilier. "Elicitation and Approximately Stable Matching With Partial Preferences". In: *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI)*. 2013, pp. 97–105 (cit. on pp. 13, 16).

[DB14] Joanna Drummond and Craig Boutilier. "Preference Elicitation and Interview Minimization in Stable Matchings". In: *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI)*. 2014, pp. 645–653 (cit. on p. 17).

[Dok+12] Elad Dokow, Michal Feldman, Reshef Meir, and Ilan Nehama. "Mechanism Design on Discrete Lines and Cycles". In: *Proceedings of the Thirteenth ACM Conference on Electronic Commerce (EC)*. ACM. 2012, pp. 423–440 (cit. on p. 50).

[Dwo+06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. "Calibrating Noise to Sensitivity in Private Data Analysis". In: *Theory of Cryptography Conference (TCC)*. 2006, pp. 265–284 (cit. on p. 112).

[Dwo+12] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. "Fairness Through Awareness". In: *Proceedings of the Third Innovations in Theoretical Computer Science Conference (ITCS)*. 2012, pp. 214–226 (cit. on p. 113).

[Dwo+15] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. "Preserving Statistical Validity in Adaptive Data Analysis". In: *Proceedings of the Forty-Seventh ACM Symposium on Theory of Computing (STOC)*. 2015, pp. 117–126 (cit. on p. 112).

[Dwo06] Cynthia Dwork. "Differential Privacy". In: *Proceedings of the Thirty-Third International Colloquium on Automata, Languages and Programming (ICALP)*. 2006, pp. 1–12 (cit. on p. 112).

[FFZ14] Aris Filos-Ratsikas, Søren Kristoffer Stiil Frederiksen, and Jie Zhang. "Social Welfare in One-Sided Matchings: Random Priority and Beyond". In: *Proceedings of the Seventh International Symposium on Algorithmic Game Theory (SAGT)*. 2014, pp. 1–12 (cit. on p. 78).

[FSY16] Itai Feigenbaum, Jay Sethuraman, and Chun Ye. "Approximately Optimal Mechanisms for Strategyproof Facility Location: Minimizing Lp Norm of Costs". In: *Mathematics of Operations Research* 42.2 (2016), pp. 434–447 (cit. on p. 51).

[FT11] Uriel Feige and Moshe Tennenholtz. "Mechanism Design With Uncertain Inputs: (To Err Is Human, to Forgive Divine)". In: *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing (STOC)*. 2011, pp. 549–558 (cit. on p. 50).

[FT16] Dimitris Fotakis and Christos Tzamos. "Strategyproof Facility Location for Concave Cost Functions". In: *Algorithmica* 76.1 (2016), pp. 143–167 (cit. on p. 51).

[FW13] Michal Feldman and Yoav Wilf. "Strategyproof Facility Location and the Least Squares Objective". In: *Proceedings of the Fourteenth ACM Conference on Electronic Commerce (EC)*. 2013, pp. 873–890 (cit. on pp. 42, 51).

[Gal+17] Ya'akov Gal, Moshe Mash, Ariel D Procaccia, and Yair Zick. "Which is the fairest (rent division) of them all?" In: *Journal of the ACM (JACM)* 64.6 (2017), pp. 1–22 (cit. on p. 102).

[Gho+11] Ali Ghodsi, Matei Zaharia, Benjamin Hindman, Andy Konwinski, Scott Shenker, and Ion Stoica. "Dominant Resource Fairness: Fair Allocation of Multiple Resource Types". In: *Proceedings of the 8th USENIX conference on Networked systems design and implementation*. 2011, pp. 323–336 (cit. on p. 1).

[GI89] Dan Gusfield and Robert W. Irving. *The Stable Marriage Problem - Structure and Algorithms*. Foundations of computing series. MIT Press, 1989 (cit. on pp. 11, 16).

[GKM17] Ashish Goel, Anilesh K Krishnaswamy, and Kamesh Munagala. "Metric Distortion of Social Choice Rules: Lower Bounds and Fairness Properties". In: *Proceedings of the Eighteenth ACM Conference on Economics and Computation (EC)*. 2017, pp. 287–304 (cit. on pp. 75, 78, 104, 113).

[Goe+19] Ashish Goel, Anilesh K Krishnaswamy, Sukolsak Sakshuwong, and Tanja Aitamurto. "Knapsack Voting for Participatory Budgeting". In: *ACM Transactions on Economics and Computation* 7.2 (2019), pp. 1–27 (cit. on p. 78).

[GP15] Jonathan Goldman and Ariel D Procaccia. "Spliddit: Unleashing Fair Division Algorithms". In: *ACM SIGecom Exchanges* 13.2 (2015), pp. 41–46 (cit. on pp. 103, 135).

[GS62] David Gale and Lloyd S Shapley. "College Admissions and the Stability of Marriage". In: *The American Mathematical Monthly* 69.1 (1962), pp. 9–15 (cit. on pp. 11, 16, 105).

[GT17] Iddan Golomb and Christos Tzamos. *Truthful Facility Location With Additive Errors*. 2017. arXiv: 1701.00529 [cs.GT] (cit. on pp. 43–44, 48, 51, 68–69, 71).

[Haz+12] Noam Hazon, Yonatan Aumann, Sarit Kraus, and Michael Wooldridge. "On the evaluation of election outcomes under uncertainty". In: *Artificial Intelligence* 189 (2012), pp. 1–18 (cit. on p. 13).

[HB04] Nathanael Hyafil and Craig Boutilier. "Regret Minimizing Equilibria and Mechanisms for Games With Strict Type Uncertainty". In: *Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence (UAI)*. 2004, pp. 268–277 (cit. on p. 14).

[HB07a] Nathanaël Hyafil and Craig Boutilier. "Mechanism Design With Partial Revelation". In: *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI)*. 2007, pp. 1333–1340 (cit. on pp. 49–50).

[HB07b] Nathanaël Hyafil and Craig Boutilier. "Partial Revelation Automated Mechanism Design". In: *Proceedings of the Twenty-Second National Conference on Artificial Intelligence (AAAI)*. 2007, pp. 72–77 (cit. on pp. 49–50).

[HIM16] Koki Hamada, Kazuo Iwama, and Shuichi Miyazaki. "The Hospitals/Residents Problem with Lower Quotas". In: *Algorithmica* 74.1 (2016), pp. 440–465 (cit. on p. 34).

[Hua+13] Chien-Chung Huang, Telikepalli Kavitha, Kurt Mehlhorn, and Dimitrios Michail. "Fair Matchings and Related Problems". In: *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*. 2013 (cit. on pp. 78, 80, 82).

[HZ79] Aanund Hylland and Richard Zeckhauser. "The Efficient Allocation of Individuals to Positions". In: *Journal of Political economy* 87.2 (1979), pp. 293–314 (cit. on pp. 73, 78).

[IM08] Robert W. Irving and David Manlove. "Approximation Algorithms for Hard Variants of the Stable Marriage and Hospitals/Residents Problems". In: *Journal of Combinatorial Optimization* 16.3 (2008), pp. 279–292 (cit. on p. 26).

[Irv+06] Robert W. Irving, Telikepalli Kavitha, Kurt Mehlhorn, Dimitrios Michail, and Katarzyna E. Paluch. “Rank-Maximal Matchings”. In: *ACM Transactions on Algorithms* 2.4 (Oct. 2006), pp. 602–610 (cit. on pp. 78, 80, 82).

[Irv04] Robert W. Irving. *Greedy Matchings*. 2004. University of Glasgow, Computing Science Department Research Report: TR-2004-177 (cit. on pp. 78, 80, 82).

[Irv85] Robert W. Irving. “An Efficient Algorithm for the “Stable Roommates” Problem”. In: *Journal of Algorithms* 6.4 (1985), pp. 577–595 (cit. on p. 11).

[Irv94] Robert W. Irving. “Stable Marriage and Indifference”. In: *Discrete Applied Mathematics* 48.3 (1994), pp. 261–272 (cit. on pp. 12, 21).

[Kar72] Richard M. Karp. “Reducibility Among Combinatorial Problems”. In: *Proceedings of the Symposium on the Complexity of Computer Computations*. 1972, pp. 85–103 (cit. on p. 33).

[LB11] Tyler Lu and Craig Boutilier. “Robust Approximation and Incremental Elicitation in Voting Protocols”. In: *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*. 2011, pp. 287–293 (cit. on p. 14).

[Lip+04] Richard J Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. “On Approximately Fair Allocations of Indivisible Goods”. In: *Proceedings of the Fifth ACM conference on Electronic Commerce (EC)*. 2004, pp. 125–131 (cit. on pp. 102, 114).

[Lu+10] Pinyan Lu, Xiaorui Sun, Yajun Wang, and Zeyuan Allen Zhu. “Asymptotically Optimal Strategy-Proof Mechanisms for Two-Facility Games”. In: *Proceedings of the Eleventh ACM Conference on Electronic Commerce (EC)*. 2010, pp. 315–324 (cit. on pp. 42, 51).

[Man+02] David Manlove, Robert W. Irving, Kazuo Iwama, Shuichi Miyazaki, and Yasufumi Morita. “Hard Variants of Stable Marriage”. In: *Theoretical Computer Science* 276.1-2 (2002), pp. 261–279 (cit. on pp. 14, 22).

[Man+19] Debmalya Mandal, Ariel D. Procaccia, Nisarg Shah, and David P. Woodruff. “Efficient and Thrifty Voting by Any Means Necessary”. In: *Proceedings of the Thirty-Third Conference on Neural Information Processing Systems (NeurIPS)*. 2019, pp. 7178–7189 (cit. on p. 78).

[Man13] David F. Manlove. *Algorithmics of Matching Under Preferences*. Vol. 2. Series on Theoretical Computer Science. WorldScientific, 2013 (cit. on pp. 11, 16).

[Man16] David F. Manlove. “Hospitals/Residents Problem”. In: *Encyclopedia of Algorithms*. 2016, pp. 926–930 (cit. on pp. 1, 11, 16).

[MD11] Jordi Massó and Inés Moreno De Barreda. “On Strategy-Proofness and Symmetric Single-Peakedness”. In: *Games and Economic Behavior* 72.2 (2011), pp. 467–484 (cit. on pp. 44, 50, 55, 63–64, 66).

[Mic07] Dimitrios Michail. “Reducing Rank-Maximal to Maximum Weight Matching”. In: *Theoretical Computer Science* 389.1-2 (2007), pp. 125–132 (cit. on p. 82).

[ML18] Vijay Menon and Kate Larson. “Robust and Approximately Stable Marriages Under Partial Information”. In: *Proceedings of the Fourteenth International Conference on Web and Internet Economics (WINE)*. 2018, pp. 341–355 (cit. on pp. iv, 78, 113).

[ML19] Vijay Menon and Kate Larson. “Mechanism Design for Locating a Facility Under Partial Information”. In: *Proceedings of the Twelfth International Symposium on Algorithmic Game Theory (SAGT)*. 2019, pp. 49–62 (cit. on pp. iv, 78, 113).

[ML20] Vijay Menon and Kate Larson. *Algorithmic Stability in Fair Allocation of Indivisible Goods Among Two Agents*. 2020. arXiv: `2007.15203 [cs.GT]` (cit. on p. iv).

[MM05] Kurt Mehlhorn and Dimitrios Michail. “Network Problems With Non-Polynomial Weights and Applications”. Manuscript. 2005 (cit. on pp. 78, 80, 82).

[MML21] Thomas Ma, Vijay Menon, and Kate Larson. “Improving Welfare in One-Sided Matching Using Simple Threshold Queries”. In: *Proceedings of the Thirteeth International Joint Conference on Artificial Intelligence (IJCAI)*. 2021 (cit. on p. iv).

[Mou80] Hervé Moulin. “On Strategy-Proofness and Single Peakedness”. In: *Public Choice* 35.4 (1980), pp. 437–455 (cit. on pp. 50, 55).

[MSW20] Debmalya Mandal, Nisarg Shah, and David P. Woodruff. “Optimal Communication-Distortion Tradeoff in Voting”. In: *Proceedings of the Twenty-First ACM Conference on Economics and Computation (EC)*. 2020, pp. 795–813 (cit. on p. 78).

[MT07] Frank McSherry and Kunal Talwar. “Mechanism Design via Differential Privacy”. In: *Proceedings of the Forty-Eighth IEEE Symposium on Foundations of Computer Science (FOCS)*. 2007, pp. 94–103 (cit. on p. 112).

[MV18] Tung Mai and Vijay V. Vazirani. "Finding Stable Matchings That Are Robust to Errors in the Input". In: *Proceedings of the Twenty-Sixth European Symposium on Algorithms (ESA)*. 2018, 60:1–60:11 (cit. on pp. 78, 113).

[Nis+07] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007 (cit. on pp. 41, 50, 55).

[NRM] NRMP: *National Resident Matching Program*. URL: https://www.nrmp.org/ (visited on 06/22/2021) (cit. on pp. 1, 16).

[PR06] Ariel D Procaccia and Jeffrey S. Rosenschein. "The Distortion of Cardinal Preferences in Voting". In: *International Workshop on Cooperative Information Agents*. 2006, pp. 317–331 (cit. on pp. 75, 78, 84).

[PR18] Benjamin Plaut and Tim Roughgarden. "Almost Envy-freeness with General Valuations". In: *Proceedings of the Twenty-Ninth ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2018, pp. 2584–2603 (cit. on pp. 114–115, 120, 185).

[PT13] Ariel D Procaccia and Moshe Tennenholtz. "Approximate Mechanism Design Without Money". In: *ACM Transactions on Economics and Computation* 1.4 (2013), pp. 18:1–18:26 (cit. on pp. 42, 48, 50–51).

[Ram13] Eve Ramaekers. "Fair Allocation of Indivisible Goods: The Two-Agent Case". In: *Social Choice and Welfare* 41.2 (2013), pp. 359–380 (cit. on pp. 114, 117, 120).

[Ras+13] Baharak Rastegari, Anne Condon, Nicole Immorlica, and Kevin Leyton-Brown. "Two-Sided Matching With Partial Information". In: *Proceedings of the Fourteenth ACM Conference on Electronic Commerce (EC)*. 2013, pp. 733–750 (cit. on p. 17).

[Ras+14] Baharak Rastegari, Anne Condon, Nicole Immorlica, Robert Irving, and Kevin Leyton-Brown. "Reasoning About Optimal Stable Matchings Under Partial Information". In: *Proceedings of the Fifteenth ACM Conference on Economics and Computation (EC)*. 2014, pp. 431–448 (cit. on pp. 12, 16).

[Rom98] Antonio Romero-Medina. "Implementation of Stable Solutions in a Restricted Matching Market". In: *Review of Economic Design* 3.2 (1998), pp. 137–147 (cit. on p. 16).

[Rot84] Alvin E Roth. "The Evolution of the Labor Market for Medical Interns and Residents: A Case Study in Game Theory". In: *Journal of political Economy* 92.6 (1984), pp. 991–1016 (cit. on pp. 1, 16).

[Rot86] Alvin E Roth. "On the Allocation of Residents to Rural Hospitals: A General Property of Two-Sided Matching Markets". In: *Econometrica: Journal of the Econometric Society* (1986), pp. 425–427 (cit. on pp. 1, 16).

[RP77] Alvin E. Roth and Andrew Postlewaite. "Weak Versus Strong Domination in a Market With Indivisible Goods". In: *Journal of Mathematical Economics* 4.2 (1977), pp. 131–137 (cit. on pp. 73, 78).

[RS90] Alvin E. Roth and Marilda A. Oliveira Sotomayor. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*. Econometric Society Monographs. Cambridge University Press, 1990 (cit. on p. 16).

[SAH17] Erel Segal-Halevi, Haris Aziz, and Avinatan Hassidim. "Fair Allocation Based on Diminishing Differences". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI)*. 2017, pp. 1254–1261 (cit. on pp. 104, 114).

[Sha+10] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. "Learnability, Stability and Uniform Convergence". In: *Journal of Machine Learning Research (JMLR)* 11.Oct (2010), pp. 2635–2670 (cit. on p. 112).

[SS13] Daniela Saban and Jay Sethuraman. "House allocation with indifferences: a generalization and a unified view". In: *Proceedings of the Fourteenth ACM conference on Electronic Commerce (EC)*. 2013, pp. 803–820 (cit. on pp. 87–90, 92, 95, 182–183).

[SS74] Lloyd Shapley and Herbert Scarf. "On Cores and Indivisibility". In: *Journal of Mathematical Economics* 1.1 (1974), pp. 23–37 (cit. on pp. 73, 78, 80).

[SÜ10] Tayfun Sönmez and M Utku Ünver. "House Allocation With Existing Tenants: A Characterization". In: *Games and Economic Behavior* 69.2 (2010), pp. 425–445 (cit. on p. 78).

[SV02] James Schummer and Rakesh V Vohra. "Strategy-Proof Location on a Network". In: *Journal of Economic Theory* 104.2 (2002), pp. 405–428 (cit. on p. 50).

[SYE13] Dmitry Shiryaev, Lan Yu, and Edith Elkind. "On Elections With Robust Winners". In: *Proceedings of the Twelfth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. 2013, pp. 415–422 (cit. on pp. 78, 113).

[Tan90] Jimmy J. M. Tan. "A Maximum Stable Matching for the Roommates Problem". In: *BIT Numerical Mathematics* 30.4 (1990), pp. 631–640 (cit. on p. 29).

[VK14] Rudolf Vetschera and D Marc Kilgour. "Fair Division of Indivisible Items Between Two Players: Design Parameters for Contested Pile Methods". In: *Theory and Decision* 76.4 (2014), pp. 547–572 (cit. on p. 114).

[XC11] Lirong Xia and Vincent Conitzer. "Determining Possible and Necessary Winners Given Partial Orders". In: *Journal of Artificial Intelligence Research (JAIR)* 41 (2011), pp. 25–67 (cit. on p. 12).

APPENDICES

Appendix A

Omitted material from Chapter 2

A.1 Missing proofs from Section 2.3.3

Lemma 7. *Let $\mathcal{I}_1$ denote some instance and $\mathcal{I}_2$ denote the instance returned by the procedure proposeWith$(A, \mathcal{I}_1)$, where the set A represents the proposing side. If there exists a matching of size t in $\mathcal{I}_1$ that is internally super-stable with respect to $\mathcal{I}$, then there exists a matching of size t in $\mathcal{I}_2$ that is internally super-stable with respect to $\mathcal{I}$.*

Proof. In order to prove this, we need a few claims. However, before that let us introduce the following terminology. When the procedure proposeWith$(A, \mathcal{I}_1)$ is executed, for every run of the while loop in line 2 with respect to an agent a, we can track the instance that is currently being used. That is, initially we have the instance $\mathcal{I}_1$ and this is referred to as the instance that is "currently being used" by the agent a_1, where a_1 is the first agent with respect whom the while loop is executed. Now, after the first run of the while loop (w.r.t. a_1) we have an updated instance (because of some delete operations that happened in lines 3–10), say, $\mathcal{I}_{curr}$. Therefore, the next time the while loop is run with respect to some agent a_2, this is the instance that is "currently being used" with respect to a_2. We use this terminology in the following claim.

Claim 12. Let $a \in A$ be an agent who is assigned to be 'free' in proposeWith$(A, \mathcal{I}_1)$, $\mathcal{I}_{curr}$ denote the instance that is currently being used with respect to a, and $\mathcal{I}_1'$ be the instance obtained by running lines 3–10 with respect to agent a. If there exists a matching of size t in $\mathcal{I}_{curr}$ that is internally super-stable with respect to $\mathcal{I}$, then there exists a matching of size t in $\mathcal{I}_1'$ that is internally super-stable with respect to $\mathcal{I}$.

Proof. First, note that every time the procedure proposeWith() is called with the set A, the agents in this set do not have any ties. This is because, when proposeWith() is called with the set of women W for the first time in line 18, all ties are already broken due to the first call of proposeWith() with the set of men in line 17. Therefore, in all the arguments below we do not need to concern ourselves with issues that can arise as a result of the agents in A having ties.

Now, to prove this claim we consider the following two cases separately.

1. when b, the first agent in a's list, is either engaged to p, but prefers a to p, or is not engaged currently

2. when b is engaged to $p \in A$ and finds p and a incomparable

Case 1: In this case the only change that happens to the instance $\mathcal{I}_{curr}$ are due to deletions of the form (c, b) where $a \succ_b c$. Let the resulting instance be $\mathcal{I}'_1$. So, in order to prove that $\mathcal{I}'_1$ has a matching of size t that is internally super-stable with respect to $\mathcal{I}$, we just need to prove that there exists a matching of size t in $\mathcal{I}_{curr}$ that is internally super-stable matching with respect to $\mathcal{I}$ and does not have any matches of the form (c, b). Now, let us suppose that's not the case and that in every matching μ of size t in $\mathcal{I}_{curr}$ that is internally super-stable with respect to $\mathcal{I}$ there exists such a pair. This in turn implies that a is unmatched in μ, for if otherwise (a, b) will form an obvious-blocking pair. Hence, we can form the matching $\mu' = (\mu \setminus (c, b)) \cup (a, b)$. Note that μ' is internally-stable with respect to $\mathcal{I}$ as this does not introduce any new super-blocking pair (since b is the first agent on a's list and b does not block any other agent in μ' as he prefers the new partner over his old partner c) and has the same size as μ. This in turn is a contradiction and hence we have that $\mathcal{I}'_1$ has a matching of size t that is internally super-stable with respect to $\mathcal{I}$.

Case 2: In this case, the only change that happens to the instance $\mathcal{I}_{curr}$ is the deletion of (a, b). Let the resulting instance be $\mathcal{I}'_1$. So, in order to prove that $\mathcal{I}'_1$ has a matching of size t that is internally super-stable with respect to $\mathcal{I}$, we just need to prove that there exists a matching of size t in $\mathcal{I}_{curr}$ that is internally super-stable matching with respect to $\mathcal{I}$ and does not have any matches of the form (a, b). Suppose that's not the case and that every matching μ of size t in $\mathcal{I}_{curr}$ that is internally super-stable matching with respect to $\mathcal{I}$ has (a, b). This in turn implies that p is unmatched in μ, for if otherwise (p, b) will form a super-blocking pair (since b is the first agent on p's list and b finds p and a incomparable). Hence, we can now form the matching $\mu' = (\mu \setminus (a, b)) \cup (p, b)$, which is internally super-stable with respect to $\mathcal{I}$ as this does not introduce any new super-blocking pair (since *i)* b is the first agent on p's list and *ii)* b does not block any other agent in μ' as it finds a and b incomparable and so if he blocks someone now he also blocked them when

(a, b) was present, thus contradicting the fact that μ was internally super-stable) and has the same size as μ. This in turn is a contradiction and hence, again, we have that $\mathcal{I}_1'$ has a matching of size t that is internally super-stable with respect to $\mathcal{I}$. □

Next, we have the following claim whose proof is very similar to Case 2 in the proof of Claim 12.

Claim 13. Let m be a man, w be the first woman on m's list, $\mathcal{I}_3$ be some instance obtained after line 11, and $\mathcal{I}_4$ be the instance obtained after deleting each (m', w) in line 14. If there exists a matching of size t in $\mathcal{I}_3$ that is internally super-stable with respect to $\mathcal{I}$, then there exists a matching of size t in $\mathcal{I}_4$ that is internally super-stable with respect to $\mathcal{I}$.

Given the two claims above, we are now ready to prove Lemma 7. First, starting with $\mathcal{I}_1$ and by repeatedly using Claim 12 with respect to each free agent $a \in A$, we can see that the instance $\mathcal{I}_1'$ that we obtain at the end of the while loop (i.e., line 11) has a matching of size t that is internally super-stable matching with respect to the initial instance $\mathcal{I}$. Second, starting with $\mathcal{I}_1'$ and by repeatedly using Claim 13 with respect to each man, we can see that the instance $\mathcal{I}_2$ that is returned from the procedure proposeWith$(A, \mathcal{I}_1)$ has a matching of size t that is internally super-stable with respect to the instance $\mathcal{I}$. This in turn proves our lemma. □

Lemma 8. *Let $\mathcal{I}_1$ denote some instance that does not contain any ties, $(m_1, w_1), (m_2, w_2), \ldots, (m_r, w_r)$ be a rotation that is exposed in $\mathcal{I}_1$, and $\mathcal{I}_2$ be the instance that is obtained by deleting the entries (m_i, w_i) for all $i \in \{1, \ldots, r\}$ from $\mathcal{I}_1$. If there exists a matching of size t in $\mathcal{I}_1$ that is internally super-stable with respect to $\mathcal{I}$, then there exists a matching of size t in $\mathcal{I}_2$ that is internally super-stable with respect to $\mathcal{I}$.*

Proof. Let μ be a matching of size t in $\mathcal{I}_1$ that is internally super-stable with respect to $\mathcal{I}$. We will consider the following two cases separately and show that in each case there exists a matching of size t in $\mathcal{I}_2$ that is internally super-stable with respect to $\mathcal{I}$.

1. for all $i \in \{1, \ldots, r\}$, $(m_i, w_i) \in \mu$
2. there exists some $j \in \{1, \ldots, r\}$ such that $(m_j, w_j) \notin \mu$

Case 1: Consider the matching μ' that is obtained by removing (m_i, w_i) and instead adding (m_i, w_{i+1}) for all $i \in \{1, \ldots, r\}$ (here $i + 1$ is done modulo r). Now, it is easy to see that this does not lead to any new internal super-blocking pairs with respect to $\mathcal{I}$. Hence, μ' is internally super-stable with respect to $\mathcal{I}$, has the same size as μ (as every agent matched in μ is also matched in μ'), and all the matched pairs in μ' are in $\mathcal{I}_2$.

Case 2: First, note that if $(m_i, w_i) \notin \mu$ for all $i \in \{1, \ldots, r\}$, then we are done as the only deleted entries in $\mathcal{I}_2$ are (m_i, w_i) for all $i \in \{1, \ldots, r\}$. So this implies that we can, without loss of generality, consider the smallest $k \in \{1, \ldots, r\}$ such that $(m_j, w_j) \notin \mu$ for all $j \in \{1, \ldots, k\}$, but $(m_{k+1}, w_{k+1}) \in \mu$ (since there is at least one j such that $(m_j, w_j) \notin \mu$, this can always be done because we can re-index the rotation so that the first element (m_1, w_1) of the rotation is not in μ). Let $\mathcal{I}'_2$ be the instance that is formed by deleting the entries (m_i, w_i) for all $i \in \{1, \ldots, k+1\}$. Below, we will show that $\mathcal{I}'_2$ satisfies the conditions of the lemma, i.e., we show that $\mathcal{I}'_2$ has a matching of size t that is internally super-stable with respect to $\mathcal{I}$. And so once we have that, we can just repeat this argument until we get to the instance $\mathcal{I}_2$.

To see why $\mathcal{I}'_2$ satisfies the conditions of the lemma, notice that m_k needs to be unmatched in μ, for if otherwise then (m_k, w_{k+1}) will be an obvious blocking-pair. This in turn implies that we can construct another matching μ' such that $\mu' = (\mu \setminus (m_{k+1}, w_{k+1})) \cup (m_k, w_{k+1})$, both of them have the same size, and all the matched pairs in μ' are in $\mathcal{I}'_2$. Also, one can see that this does not lead to any new super-blocking pairs since *i)* w_{k+1} improved and so will not be part of any new super-blocking pair and *ii)* m_k does not form a super-blocking pair as it is matched to the agent who is second in his list and w_k who is first in his list, if matched in μ, has to be matched to someone better (as m_k is the last agent in w_k's list). ☐

Lemma 9. *If $\mathcal{I}_1$ is an instance that does not contain any exposed rotation, then the list of every man in $\mathcal{I}_1$ has only one woman and vice versa.*

Proof. First, note that, throughout, an agent a is in b's list if and only if b is in a's list. Second, we prove the following claim.

Claim 14. Let $\mathcal{I}_2$ be some initial instance, $\mathcal{I}_3$ be the instance that is obtained after running the procedure proposeWith() with the men's side proposing, and $\mathcal{I}_4$ be the instance that is obtained after running the procedure proposeWith() with the women's side proposing. For any two agents a and b in $\mathcal{I}_4$, if a is the only agent in b's list, then b is the only agent in a's list.

Proof. To prove our claim let us consider the following two cases in the instance $\mathcal{I}_3$.

1. there exists a woman w_1 such that m_1 is the only man on w_1's list, but m_1 has at least one other woman other than w_1 in his list

2. there exists a man m_1 such that w_1 is the only woman on m_1's list, but w_1 has at least one other man other than m_1 in her list

Case 1: Since we are looking at $\mathcal{I}_3$, note that w_1 must be the first woman on m_1's list, for if otherwise there is some other man, say, m_k who is not engaged (because w_1 has only m_1 in her list). However, we know that this is not possible and so w_1 is the first woman on m_1's list. Therefore, now, when we run the procedure proposeWith() with the women's side proposing, w_1 will propose to m_1 and as a result m_1 will delete all other women from his list. And so, in $\mathcal{I}_4$, w_1 is the only agent in m_1's list and vice versa.

Case 2: For this case, let us suppose that even after running the procedure proposeWith() with the women's side proposing, w_1 still has at least one other man other than m_1 in her list (if not, then we are already done). Let this man be m_2. This in turn implies that in the engagement relation that results, at least one of the women, say, w_k, is not engaged (because m_1 has only w_1 in his list). However, we know that this is not possible as this would imply that the engagement relation has an obvious blocking pair (m_k, w_k), where m_k is the last agent on w_k's list (and we know this cannot happen since we are just running the same proposal-rejection sequence as in the Gale-Shapley algorithm). □

Given the observations above, let us assume for the sake of contradiction that there exists an instance $\mathcal{I}_1$ such that it does not have any exposed rotation but has at least one agent who has a list of size greater than one. Using Claim 14 and the fact that an agent a is in b's list if and only if b is in a's list, we can assume without loss of generality that there exists a man m_1 such that he has at least two women in his list. Let w_1 and w_2 be the first and second woman, respectively, in m_1's list. Since $\mathcal{I}_1$ is obtained after running the procedure proposeWith() twice, once each with the men's and women's side proposing, from Claim 14 we know that m_2, who is the last man on w_2's list, too has at least two women in his list as otherwise w_2 would also have just one man m_2 in her list. Let w_3 be the second woman in m_2's list. Now, we can see that we can just inductively keep on applying the above argument and form the following pairs $\rho = (m_1, w_1), (m_2, w_2), (m_3, w_3), \ldots, (m_r, w_r)$ for some $r \in \{2, \ldots, n\}$, where m_i is the last man on w_i's list and w_{i+1} is the second woman on m_i's list. However, note that ρ is an exposed rotation, and hence we have a contradiction. □

A.2 An example of a "bad" weakly-stable matching in the case of one-sided top-truncated preferences

Consider the instance $\mathcal{I}$ as shown in Figure A.1, where ties appear only on the women's side. Furthermore, we define the following:

Men

$$
\begin{aligned}
m_1 &: w_1 \succ W_{F\setminus\{1\}} \succ W_{B_1} \succ \ldots \succ W_{B_z} \\
m_2 &: w_1 \succ w_2 \succ [\ldots] \\
m_3 &: w_2 \succ w_3 \succ W_{F\setminus\{2,3\}} \succ W_S \\
m_4 &: w_2 \succ w_4 \succ W_{F\setminus\{2,4\}} \succ W_S \\
&\vdots \\
m_{\frac{n}{2}} &: w_2 \succ w_{\frac{n}{2}} \succ W_{F\setminus\{2,\frac{n}{2}\}} \succ W_S \\
m_{b_0} &: w_1 \succ W_{B_1\setminus\{b_0\}} \succ w_{b_0} \succ W_{S\setminus B_1} \succ W_{F\setminus\{1\}} \\
&\vdots \\
m_{b_1-1} &: w_1 \succ W_{B_1\setminus\{b_1-1\}} \succ w_{b_1-1} \succ W_{S\setminus B_1} \succ W_{F\setminus\{1\}} \\
m_{b_1} &: w_1 \succ W_{B_2\setminus\{b_1\}} \succ w_{b_1} \succ W_{S\setminus B_2} \succ W_{F\setminus\{1\}} \\
&\vdots \\
m_{b_2-1} &: w_1 \succ W_{B_2\setminus\{b_2-1\}} \succ w_{b_2-1} \succ W_{S\setminus B_2} \succ W_{F\setminus\{1\}} \\
&\vdots \\
m_{b_{z-1}} &: w_1 \succ W_{B_z\setminus\{b_{z-1}\}} \succ w_{b_{z-1}} \succ W_{S\setminus B_z} \succ W_{F\setminus\{1\}} \\
&\vdots \\
m_{b_z-1} &: w_1 \succ W_{B_z\setminus\{b_z-1\}} \succ w_{b_z-1} \succ W_{S\setminus B_z} \succ W_{F\setminus\{1\}}
\end{aligned}
$$

Women

$$
\begin{aligned}
w_1 &: m_2 \succ m_1 \succ [\ldots] \\
w_2 &: m_2 \succ m_1 \succ [\ldots] \\
w_3 &: m_1 \succ m_3 \succ [\ldots] \\
w_4 &: m_1 \succ m_4 \succ [\ldots] \\
&\vdots \\
w_{\frac{n}{2}} &: m_1 \succ m_{\frac{n}{2}} \succ [\ldots] \\
w_{b_0} &: M_{S\setminus B_1} \succ m_1 \succ m_{b_0} \succ M_{F\setminus\{1\}} \succ M^T_{B_1\setminus\{b_0\}} \\
&\vdots \\
w_{b_1-1} &: M_{S\setminus B_1} \succ m_1 \succ m_{b_1-1} \succ M_{F\setminus\{1\}} \succ M^T_{B_1\setminus\{b_1-1\}} \\
w_{b_1} &: M_{S\setminus B_1} \succ m_1 \succ m_{b_1} \succ M_{F\setminus\{1\}} \succ M^T_{B_1\setminus\{b_1\}} \\
&\vdots \\
w_{b_2-1} &: M_{S\setminus B_1} \succ m_1 \succ m_{b_2-1} \succ M_{F\setminus\{1\}} \succ M^T_{B_1\setminus\{b_2-1\}} \\
&\vdots \\
w_{b_{z-1}} &: M_{S\setminus B_1} \succ m_1 \succ m_{b_{z-1}} \succ M_{F\setminus\{1\}} \succ M^T_{B_1\setminus\{b_{z-1}\}} \\
&\vdots \\
w_{b_z-1} &: M_{S\setminus B_1} \succ m_1 \succ m_{b_z-1} \succ M_{F\setminus\{1\}} \succ M^T_{B_1\setminus\{b_z-1\}}
\end{aligned}
$$

Figure A.1: The instance $\mathcal{I}$ that is used to illustrate that there can be weakly-stable matchings with $\mathcal{O}(n^2\sqrt{\delta})$ super-blocking pairs even in the case of one-sided top-truncated preferences

- $\delta \in [\frac{16}{n^2}, \frac{1}{4}]$, $y = \frac{n\sqrt{\delta}}{2}$, $z = \frac{n}{2y}$ (for simplicity we assume that y and z are integers; we can appropriately modify the proof if that is not the case)
- $b_j = \frac{n}{2} + jy + 1, \forall j \in [0, \ldots, z]$
- $B_i = \{b_{i-1}, \ldots, b_i - 1\}, \forall i \in [1, \ldots, z]$, $F = \{1, \ldots, \frac{n}{2}\}, S = \{\frac{n}{2} + 1, \ldots, n\}$
- $W_X\,(M_X)$: for some set X, place all the women (men) with index in X in the increasing order of their indices
- $W^T_X\,(M^T_X)$: for some set X, place all the women (men) with index in X as tied
- $[\ldots]$: place all the remaining alternatives in some strict order.

First thing is to see that the optimal solution $\mathcal{M}_{opt}$ associated with the instance is

$$\mathcal{M}_{opt} = \{(m_1, w_1), (m_2, w_2), \ldots, (m_n, w_n)\},$$

where (m_2, w_1) is the only super-blocking pair (and it is an obvious blocking pair). Also, it can be verified that the total amount of missing information in $\mathcal{I}$ is at most δ.

Now, consider the matching $\mathcal{M}$, where

$$\mathcal{M} = \Big\{ (m_1, w_2), (m_2, w_1), (m_3, w_3), (m_4, w_4), \ldots, (m_{\frac{n}{2}}, w_{\frac{n}{2}}),$$
$$(m_{b_0}, w_{b_0+1}), (m_{b_0+1}, w_{b_0+2}), \ldots, (m_{b_1-2}, w_{b_1-1}), (m_{b_1-1}, w_{b_0}),$$
$$(m_{b_1}, w_{b_1+1}), (m_{b_1+1}, w_{b_1+2}), \ldots, (m_{b_2-2}, w_{b_2-1}), (m_{b_2-1}, w_{b_1}), \ldots,$$
$$\ldots, (m_{b_{z-1}}, w_{b_{z-1}+1}), (m_{b_{z-1}+1}, w_{b_{z-1}+2}), \ldots, (m_{b_z-2}, w_{b_z-1}), (m_{b_z-1}, w_{b_{z-1}}) \Big\}.$$

It is easy to check that $\mathcal{M}$ is weakly-stable. Also, it can be verified that it has $\mathcal{O}(n^2\sqrt{\delta})$ super-blocking pairs (this is because with respect to each block B_j one can see that $\mathcal{M}$ has $\mathcal{O}(|B_j|^2)$ super-blocking pairs). □

Appendix B

Omitted material from Chapter 3

B.1 Revelation principle for minimax dominant strategies

Below we show that in the setting under consideration the revelation principle holds with respect to minimax dominant strategies.

Lemma 41. *Let $\mathcal{M}$ be a mechanism that implements a social choice function f in minimax dominant strategies. Then, there exists a direct mechanism $\mathcal{M}'$ that implements f and where for every agent i reporting her candidate locations K_i is a minimax dominant strategy.*

Proof. Let $(s_1, \ldots, s_n)$ be the minimax dominant strategy in $\mathcal{M}$ such that $f(K_1, \ldots, K_n) = F(s_1(K_1), \ldots, s_n(K_n))$, where $F(\cdot)$ is the outcome function associated with $\mathcal{M}$. Next, let us define the outcome function, F', associated with $\mathcal{M}'$ as

$$F'(K_1, \ldots, K_n) = F(s_1(K_1), \ldots, s_n(K_n)). \tag{B.1}$$

Now, using the fact that $(s_1, \ldots, s_n)$ is a minimax dominant strategy in $\mathcal{M}$, we have that $\forall K_i', \forall K_{-i}$,

$$\begin{aligned}\max_{\ell_i \in K_i} \max_{\sigma_i \in \Delta(\Sigma_i)} & C(\ell_i, F(s_i(K_i), s_{-i}(K_{-i})) - C(\ell_i, F(\sigma_i(\ell_i), s_{-i}(K_{-i})) \\ &\leq \max_{\ell_i \in K_i} \max_{\sigma_i \in \Delta(\Sigma_i)} C(\ell_i, F(s_i(K_i'), s_{-i}(K_{-i})) - C(\ell_i, F(\sigma_i(\ell_i), s_{-i}(K_{-i}))).\end{aligned} \tag{B.2}$$

Additionally, if ℓ_i is the true location of agent i, then, again, using the fact that $(s_1, \ldots, s_n)$ is a minimax dominant strategy, we have that $\forall s_i' \in \Sigma_i, \forall K_{-i}$,

$$C(\ell_i, F(s_i(\ell_i), s_{-i}(K_{-i})) \leq C(\ell_i, F(s_i'(\ell_i), s_{-i}(K_{-i})). \tag{B.3}$$

Therefore,

$$\min_{\sigma_i \in \Delta(\Sigma_i)} C(\ell_i, F(\sigma_i(\ell_i), s_{-i}(K_{-i})) = C(\ell_i, F(s_i(\ell_i), s_{-i}(K_{-i})),$$

and using this in (B.2) we have that $\forall K_i', \forall K_{-i}$,

$$\begin{aligned}\max_{\ell_i \in K_i} C(\ell_i, F(s_i(K_i), s_{-i}(K_{-i}))) - C(\ell_i, F(s_i(\ell_i), s_{-i}(K_{-i}))) \\ \leq \max_{\ell_i \in K_i} C(\ell_i, F(s_i(K_i'), s_{-i}(K_{-i}))) - C(\ell_i, F(s_i(\ell_i), s_{-i}(K_{-i}))).\end{aligned} \tag{B.4}$$

This in turn implies that using (B.1) we have that $\forall K_i', \forall K_{-i}$,

$$\max_{\ell_i \in K_i} C(\ell_i, F'(K_i, K_{-i})) - C(\ell_i, F'(\ell_i, K_{-i})) \leq \max_{\ell_i \in K_i} C(\ell_i, F'(K_i', K_{-i})) - C(\ell_i, F'(\ell_i, K_{-i})),$$

or in other words that reporting the candidate locations K_i is a minimax dominant strategy in $\mathcal{M}'$. $\square$

B.2 Minimax optimal solution for avgCost

Given the candidate locations $K_i = [a_i, b_i]$ for all i, where, for some $\delta \in [0, B]$, $b_i - a_i \leq \delta$, in this section we are concerned with computing the minimax optimal solution p_{opt} such that $p_{opt} = \arg\min_{p \in [0,B]} \max_{I \in \mathcal{I}} \left(S(I, p) - \min_{p' \in [0,B]} S(I, p') \right)$, where $\mathcal{I} = [a_1, b_1] \times \ldots \times [a_n, b_n]$ and S is the average cost function. Note that from the discussion in Section 3.4 we know that for any $I \in \mathcal{I}$, $\min_{p' \in [0,B]} S(I, p') = S(I, p_I)$, where p_I is the median of the points in the vector I. Therefore, we can rewrite the definition of p_{opt} as $p_{opt} = \arg\min_{p \in [0,B]} \max_{I \in \mathcal{I}} S(I, p) - S(I, p_I)$. Next, we introduce the following notation.

Notation: Consider the left endpoints associated with all the agents, i.e., the set $\{a_i\}_{i \in [n]}$. We denote the sorted order of these points as $L_1, \ldots, L_n$ (throughout, by sorted order we mean sorted in non-decreasing order). Similarly, we denote the sorted order of the right endpoints, i.e., the points in the set $\{b_i\}_{i \in [n]}$, as $R_1, \ldots, R_n$. Additionally, for $i \in [n]$, we use M_i to denote the mean of L_i and R_i (i.e., $M_i = \frac{L_i + R_i}{2}$). Throughout, for $\mathcal{I} = K_1 \times \ldots \times K_n$, we refer to an element of $\mathcal{I}$ as an "input" and whenever we refer to an input $I \in \mathcal{I}$, where

$I = (\ell_1, \ldots, \ell_n)$, we assume without loss of generality that the l_is are in sorted order (because the agents can always be re-indexed). Also, given a point p, let $\mathcal{I}_1(p) \subseteq \mathcal{I}$ be the set of all inputs $I_1 = (\ell_1, \ldots, \ell_n)$ such that $\ell_{k+1} \geq p$. Similarly, let $\mathcal{I}_2(p) \subseteq \mathcal{I}$ be the set of all inputs $I_2 = (\ell'_1, \ldots, \ell'_n)$ such that $\ell'_{k+1} < p$. Often when the point p is clear from the context, we write $\mathcal{I}_1$ and $\mathcal{I}_2$ to refer to $\mathcal{I}_1(p)$ and $\mathcal{I}_2(p)$, respectively.

Armed with the notation defined above, we can now prove the following lemma (which is also stated in Section 3.4), which gives a concise formula to find the maximum regret associated with a point p for the average cost objective.

Lemma 16. *Given a point p, the maximum regret associated with p for the average cost objective can be written as* $\max(obj_1^{AC}(p), obj_2^{AC}(p))$, *where*

- $obj_1^{AC}(p) = \frac{1}{n}\left(2\sum_{i=j}^{k}(R_i - p) + (n-2k)(R_{k+1} - p)\right)$, *where j is the smallest index such that $R_j > p$ and $j \leq k$*
- $obj_2^{AC}(p) = \frac{1}{n}\left(2\sum_{i=k+2}^{h}(p - L_i) + (n-2k)(p - L_{k+1})\right)$, *where h is the largest index such that $L_h < p$ and $h \geq k+2$.*

Proof. To prove this, consider the maximum regret associated with locating the facility at p which is given by $\max_{I \in \mathcal{I}} S(I, p) - S(I, p_I)$. Given the fact that $\mathcal{I} = \mathcal{I}_1 \cup \mathcal{I}_2$, we can rewrite this as $\max(\max_{I_1 \in \mathcal{I}_1} S(I_1, p) - S(I_1, p_{I_1}), \max_{I_2 \in \mathcal{I}_2} S(I_2, p) - S(I_2, p_{I_2}))$. Now, let us consider each of these terms separately in the cases below.

Case 1: $\max_{I_1 \in \mathcal{I}_1} S(I_1, p) - S(I_1, p_{I_1})$**.** Let us consider an arbitrary input $I_1 = (\ell_1, \ldots, \ell_n)$ that belongs to $\mathcal{I}_1$ (if $\mathcal{I}_1 = \emptyset$, then we define $\max_{I_1 \in \mathcal{I}_1} S(I_1, p) - S(I_1, p_{I_1}) = 0$). Now, the regret associated with I_1 is given by

$$
\begin{aligned}
\text{regret}(p, I_1) &= S(I_1, p) - S(I_1, p_{I_1}) \\
&= \frac{1}{n}\left(\sum_{i=1}^{n} |\ell_i - p| - \sum_{i=1}^{n} |\ell_i - \ell_{k+1}|\right) && \big(\text{as } p_{I_1} = \ell_{k+1}\big) \\
&= \frac{1}{n}\left(\sum_{i=1}^{k} |\ell_i - p| + \sum_{i=1}^{n-k}(\ell_{k+i} - p) - \left(\sum_{i=1}^{k}(\ell_{k+1} - \ell_i) + \sum_{i=2}^{n-k}(\ell_{k+i} - \ell_{k+1})\right)\right) \\
& && \big(\text{as } \ell_{k+1} \geq p \text{ and } \ell_1 \leq \ldots \leq \ell_n\big) \\
&= \frac{1}{n}\left(\sum_{i=1}^{k} (|\ell_i - p| + \ell_i - p) + (n-2k)(\ell_{k+1} - p)\right)
\end{aligned}
$$

$$= \frac{1}{n}\left(\sum_{i=1}^{j-1}((p-\ell_i)+\ell_i-p)+\sum_{i=j}^{k}((\ell_i-p)+\ell_i-p)+(n-2k)(\ell_{k+1}-p)\right)$$

$$\big(\text{where } j \text{ is the smallest index such that } l_j > p \text{ and } j \le k\big)$$

$$= \frac{1}{n}\left(2\sum_{i=j}^{k}(\ell_i-p)+(n-2k)(\ell_{k+1}-p)\right).$$

Therefore, if $obj_1^{AC}(p) = \max_{I_1 \in \mathcal{I}_1} S(I_1, p) - S(I_1, p_{I_1})$, then we have that

$$obj_1^{AC}(p) = \max_{I_1 \in \mathcal{I}_1} S(I_1, p) - S(I_1, p_{I_1}) = \max_{I_1 \in \mathcal{I}_1} \frac{1}{n}\left(2\sum_{i=j}^{k}(\ell_i-p)+(n-2k)(\ell_{k+1}-p)\right).$$

And so, now since $\ell_i \in [L_i, R_i]$ (this is proved in Claim 16, which is in Appendix B.3), we have that

$$obj_1^{AC}(p) = \frac{1}{n}\left(2\sum_{i=j}^{k}(R_i-p)+(n-2k)(R_{k+1}-p)\right),$$

where j is smallest index such that $R_j > p$ and $j \le k$.

Case 2: $\max_{I_2 \in \mathcal{I}_2} S(I_2, p) - S(I_2, p_{I_2})$**.** Like in the previous case, consider an arbitrary input $I_2 = (\ell'_1, \dots, \ell'_n)$ that belongs to $\mathcal{I}_2$ (if $\mathcal{I}_2 = \emptyset$, then we define $\max_{I_2 \in \mathcal{I}_2} S(I_2, p) - S(I_2, p_{I_2}) = 0$). Now, doing a similar analysis as in Case 1, we will see that the regret associated with I_2 is given by $\frac{1}{n}\left(2\sum_{i=k+2}^{h}(p-\ell'_i)+(n-2k)(p-\ell'_{k+1})\right)$, where h is the largest index such that $\ell_h < p$ and $h \ge k+2$. Therefore, $obj_2^{AC}(p) = \max_{I_2 \in \mathcal{I}_2} S(I_2, p) - S(I_2, p_{I_2}) = \frac{1}{n}\left(2\sum_{i=k+2}^{h}(p-L_i)+(n-2k)(p-L_{k+1})\right)$, where h is largest index such that $L_h < p$ and $h \ge k+2$.

Hence, combining both the cases we have that the maximum regret associated with p is $\max(obj_1^{AC}(p), obj_2^{AC}(p))$. □

Using the lemma proved above, one can show that for the minimax optimal solution, p_{opt}, $obj_1^{AC}(p_{opt}) = obj_2^{AC}(p_{opt})$ (this is proved in Claim 17, which is in Appendix B.3). And this observation in turn brings us to our next lemma (which is, again, also stated in Section 3.4) which shows that p_{opt} is always in the interval $[L_{k+1}, R_{k+1}]$.

Lemma 17. *If p_{opt} is the minimax optimal solution associated with the avgCost objective function, then $p_{opt} \in [L_{k+1}, R_{k+1}]$.*

Proof. Let us assume for the sake of contradiction that $p_{opt} < L_{k+1}$ or $p_{opt} > R_{k+1}$. We will consider each of these cases separately and show that for both the cases M_{k+1} has a lesser maximum regret, thus contradicting the fact that p_{opt} is the minimax optimal solution.

Case 1: $p_{opt} < L_{k+1}$. From Lemma 16 we know that $n \cdot obj_1^{AC}(M_{k+1}) = 2\sum_{i=j}^{k}(R_i - M_{k+1}) + (n-2k)(R_{k+1} - M_{k+1})$, where j is smallest index such that $R_j > M_{k+1}$ and $j \leq k$ (if there is no such j, then set $j = k+1$). Next, consider the input $I_1 = (R_1, \ldots, R_{2k+1})$ that belongs to $\mathcal{I}_1(p_{opt})$ and let us calculate the regret associated with p_{opt} for I_1.

$$\begin{aligned}
\text{regret}(p_{opt}, I_1) &= \frac{1}{n}\left(2\sum_{i=j'}^{k}(R_i - p_{opt}) + (n-2k)(R_{k+1} - p_{opt})\right) \\
&\qquad \qquad \big(j' \text{ is the smallest index such that } R'_j > p_{opt} \text{ and } j' \leq k\big) \\
&\geq \frac{1}{n}\left(2\sum_{i=j}^{k}(R_i - p_{opt}) + (n-2k)(R_{k+1} - p_{opt})\right) \quad \big(j' \leq j \text{ since } R_j > M_{k+1} > p_{opt}\big) \\
&> \frac{1}{n}\left(2\sum_{i=j}^{k}(R_i - L_{k+1}) + (n-2k)(R_{k+1} - p_{opt})\right) \qquad \big(\text{as } p_{opt} < L_{k+1}\big) \\
&> \frac{1}{n}\left(2\sum_{i=j}^{k}(R_i - (2M_{k+1} - R_{k+1})) + (n-2k)(R_{k+1} - (2M_{k+1} - R_{k+1}))\right) \\
&\qquad \qquad \big(\text{as } L_{k+1} = 2M_{k+1} - R_{k+1}\big) \\
&= obj_1^{AC}(M_{k+1}) + \frac{1}{n}(n - 2j + 2)(R_{k+1} - M_{k+1}).
\end{aligned}$$

Now, since p_{opt} is the optimal solution and $\text{maxRegret}(p_{opt}, \mathcal{I}) = obj_1^{AC}(p_{opt})$ (from Claim 17), we have that $obj_1^{AC}(p_{opt}) \geq \text{regret}(p_{opt}, I_1)$, which in turn implies that from above we have

$$obj_1^{AC}(p_{opt}) > obj_1^{AC}(M_{k+1}) + \frac{1}{n}(n - 2j + 2)(R_{k+1} - M_{k+1}). \tag{B.5}$$

Next, let us consider $obj_2^{AC}(p_{opt})$. Note that, given Claim 16 and since $p_{opt} < L_{k+1}$, $\mathcal{I}_2(p_{opt}) = \emptyset$. Therefore, $obj_2^{AC}(p_{opt}) = 0$, and so from Claim 17 we know that since p_{opt} is an optimal solution, $obj_1^{AC}(p_{opt}) = obj_2^{AC}(p_{opt}) = 0$. However, $j \leq k+1$, and hence from (B.5) we have that $obj_1^{AC}(p_{opt}) > 0$, thus in turn contradicting the fact that p_{opt} is the minimax optimal solution.

Case 2: $p_{opt} > R_{k+1}$. We can do a similar analysis as in Case 1 to again show that this cannot be the case.

Hence, from both the cases above, we have that $p_{opt} \in [L_{k+1}, R_{k+1}]$. □

Input: For each agent i, their input interval $[a_i, b_i]$
Output: Minimax optimal solution for the given instance

$p_{opt} \leftarrow 0$, $mR \leftarrow \infty$
Let $\{L_i\}_{i\in[n]}$ be the sorted order the points in $\{a_i\}_{i\in[n]}$ and $\{R_i\}_{i\in[n]}$ be the sorted order the points in $\{b_i\}_{i\in[n]}$
Let $C = \{R_i, L_j | i \leq k+1, j \geq k+1, L_{k+1} < R_i, L_j < R_{k+1}\}$ and $H = \{h_1, \ldots, h_{|C|}\}$ be the sorted order of the points in C.
for each $h_i \in H$, where $i \in \{1, \ldots, |C|\}$ **do**
    $x_i \leftarrow \#(R_j)$, where $R_j \geq h_i$ and $j \leq k$
    $y_i \leftarrow \#(L_j)$, where $L_j \leq h_i$ and $j \geq k+2$
    $S_i^1 \leftarrow \sum(R_j)$, where $R_j \geq h_i$ and $j \leq k$
    $S_i^2 \leftarrow \sum(L_j)$,where $L_j \leq h_i$ and $j \geq k+2$
end for
for each $[h_i, h_{i+1}]$, where $i \in \{1, \ldots, |C|\}$ **do**
    if $obj_1^{AC}(h_i) == obj_2^{AC}(h_i)$ and $obj_1^{AC}(h_i) < mR$ **then** ▷ Is h_i a possible solution?
        $p_{opt} \leftarrow h_i, mR \leftarrow obj_1^{AC}(h_i)$
    end if
    if $obj_1^{AC}(h_{i+1}) == obj_2^{AC}(h_{i+1})$ and $obj_1^{AC}(h_{i+1}) < mR$ **then** ▷ Is h_{i+1} a possible solution?
        $p_{opt} \leftarrow h_{i+1}, mR \leftarrow obj_1^{AC}(h_{i+1})$
    end if
    $p_i \leftarrow \frac{(n-2k)M_{k+1}+S_{i+1}^1+S_i^2}{x_{i+1}+y_i+(n-2k)}$ ▷ If $p_{opt} \in (h_i, h_{i+1})$, then use the fact that $obj_1^{AC}(p_{opt}) = obj_2^{AC}(p_{opt})$
    if $p_i \in (h_i, h_{i+1})$ and $obj_1^{AC}(p_i) < mR$ **then** ▷ Is p_i feasible and is it a possible solution?
        $p_{opt} \leftarrow p_i, mR \leftarrow obj_1^{AC}(p_i)$
    end if
end for
return p_{opt}

Algorithm 9: Algorithm to compute the minimax optimal solution.

Equipped with the lemmas proved above, we are now ready to compute the minimax optimal solution.

Proposition 42. *If p_{opt} is the minimax optimal solution associated with the avgCost objective function, then Algorithm 1 computes p_{opt} in $O(n \log n)$ time.*

Proof. We know from Lemma 17 that $p_{opt} \in [L_{k+1}, R_{k+1}]$. So, all we are doing in Algorithm 1 is to consider all the points in $C = \{R_i, L_j | i \leq k+1, j \geq k+1, R_i > L_{k+1}, L_j < R_{k+1}\}$ in sorted order (which is the set H in the algorithm) and check if for every interval $[h_i, h_{i+1}]$, whether $p_{opt} = h_i$, $p_{opt} = h_{i+1}$, or $p_{opt} \in (h_i, h_{i+1})$ (see lines 10–21). In the last case, since there are no points in C that are between h_i and h_{i+1}, we can use the fact that for an optimal point p_{opt}, $obj_1^{AC}(p_{opt}) = obj_2^{AC}(p_{opt})$ (this is proved in Claim 17, which is in Appendix B.3) to obtain a value of p_i (line 17). In line 18 we just check if this point actually lies in (h_i, h_{i+1}) and also see if it is better than the best solution we currently have.

Through the cases we consider in Algorithm 1 we are trying out all possible values the minimax optimal solution can take and hence the correctness follows. Also, it is easy to see that this can be done in $O(n \log n)$ time. □

B.3 Additional claims

Claim 15. For all $i \in [n]$, $L_i \leq R_i$. Additionally, $R_i - L_i \leq \delta$.

Proof. Note that the statement is true if we show that for every L_i there are at most $(i-1)$ values in $\{R_k\}_{k\in[n]}$ such that they have a value less than L_i (this is enough as this would imply that $R_i \geq L_i$). To see why that is true, consider the numbers $L_i, \ldots, L_n$ which are left endpoints of the reports of some agents. We know that each of these have a right endpoint associated with them (i.e., a b', where the input is of the form $[L_i, b']$) that are greater than them. Now, these constitute $(n-i+1)$ numbers and since there are only n in total, there can at most $i-1$ of them that have value less than L_i.

To see the second part of the claim, consider the smallest j such that $R_j - L_j > \delta$. Now, if b is the right endpoint associated with L_j, then the fact that $R_j - L_j > \delta$ implies that $b < R_j$ (because otherwise $R_j - L_j \leq b - L_j \leq \delta$). Additionally, this in turn also implies that for some L_k, where $k \in \{1, \ldots, j-1\}$, there is a right endpoint b' associated with L_k such that $b' \geq R_j$. However, since j is the smallest index value such that $R_j - L_j > \delta$, we now have a contradiction as $\delta < R_j - L_j < R_j - L_k \leq b' - L_k \leq \delta$. □

Claim 16. For any input $I = (\ell_1, \ldots, \ell_n)$, $\ell_i \in [L_i, R_i]$.

Proof. Recall that, as mentioned under notations in Appendix B.2, the ℓs are in sorted order and I is valid input if and only if there is a entry associated with every agent in it (i.e., for every agent i, $\exists j\colon \ell_j \in [a_i, b_i]$). Now, let us assume for the sake of contradiction that $\ell_i < L_i$. This implies that for I to be a valid input, there has to be $(n-i+1)$ values in it that are greater than ℓ_i—one from each agent i who reported an interval $[a_i, b_i]$ such that $a_i \geq L_i$. However, this is not possible since there can only be at most $(n-i)$ values greater than ℓ_i as ℓ_i is the ith element in a sorted list.

Similarly, one can argue analogously for the case when $\ell_i > R_i$. Therefore, $\ell_i \in [L_i, R_i]$. $\square$

Claim 17. If p_{opt} is a minimax optimal solution, then $obj_1^{AC}(p_{opt}) = obj_2^{AC}(p_{opt})$.

Proof. For the sake of contradiction, let us assume without loss of generality that $obj_1^{AC}(p_{opt}) < obj_2^{AC}(p_{opt})$. Also, for the point p_{opt}, let j be the smallest index such that $R_j > p_{opt}$ and $j \leq k$ (if no such j, then set $j = k+1$) and let h be the largest index such that $L_h < p_{opt}$ and $h \geq k+2$ (if no such h, then set $h = k+1$). Note that if n is even, then both j and h cannot be $k+1$, for if so then $obj_1^{AC}(p_{opt})$ would be equal to $obj_2^{AC}(p_{opt})$. Now, consider the point $p = p_{opt} - \epsilon$, where $\epsilon = \frac{n(obj_2^{AC}(p_{opt}) - obj_1^{AC}(p_{opt}))}{\max((n-2j+2),(2h-n))}$, and let us compute $obj_1^{AC}(p)$ and $obj_2^{AC}(p)$.

$$\begin{aligned}
obj_1^{AC}(p) &= \frac{1}{n}\left(2\sum_{i=j'}^{k}(R_i - p) + (n-2k)(R_{k+1} - p)\right) \\
&\qquad \big(j' \text{ the smallest index such that } R_{j'} > p \text{ and } j' \leq k \text{ (if no such } j'\text{, then set } j' = k+1)\big) \\
&= \frac{1}{n}\left(2\sum_{i=j}^{k}(R_i - (p_{opt} - \epsilon)) + (n-2k)(R_{k+1} - (p_{opt} - \epsilon))\right) \qquad \big(\text{note that } j' = j^{1}\big) \\
&= obj_1^{AC}(p_{opt}) + \frac{1}{n}\epsilon(n-2j+2) \\
&\leq \frac{obj_1^{AC}(p_{opt}) + obj_2^{AC}(p_{opt})}{2} \\
&< obj_2^{AC}(p_{opt}).
\end{aligned}$$

Computing $obj_2^{AC}(p)$ similarly we see that $obj_2^{AC}(p) < obj_2^{AC}(p_{opt})$. Now, since $obj_1^{AC}(p) < obj_2^{AC}(p_{opt})$ and $obj_2^{AC}(p) < obj_2^{AC}(p_{opt})$, this implies that p_{opt} cannot be the minimax optimal solution as its maximum regret is larger than that of p's. $\square$

Claim 18. In the $\frac{\delta}{2}$-equispaced-median mechanism, let $[a, b]$ be an interval of length at most δ and let x and y be the points that are closest (with ties being broken in favour of points in $[a, b]$ in both cases) to a and b, respectively. Then, the interval $[x, y]$ can have at most 3 points that are also in A.

[1] because if not then p_{opt} is not optimal since we can move to $R_{j'} \neq R_j$ and it can be verified that this point has regret less than that of p_{opt}.

Proof. Suppose this were not the case and there existed two other points $x' < y'$ such that $x', y' \in A$ and $x', y' \in (x, y)$. Now, for this to happen x has to be less than a and y has to be greater than b, for if otherwise then using the fact that x and y are the points that are the closest (with ties broken in favour of points in $[a, b]$) to a and b, respectively, one can see that we can have only at most 3 points in $[x, y]$. However, this would imply that

$$\begin{aligned}
2(b-a) &= (b-y'+y'-x'+x'-a)+(b-a)\\
&> (y-b)+(y'-x')+(a-x)+(b-a)\\
&\quad \big(b-y' > y-b \text{ and } x'-a > a-x\text{, because } x, y \text{ are closest to } a, b \text{ and using the tie-breaking rule}\big)\\
&= (y-x)+(y-x')\\
&= 2\delta, \qquad \big(y-x = y-y'+y'-x'+x'-x = \tfrac{3\delta}{2} \text{ as the distance between points in } A \text{ is } \tfrac{\delta}{2}\big)
\end{aligned}$$

which in turn contradicts the fact that $b - a \le \delta$. □

Claim 19. Consider a mechanism $\mathcal{M} = (X, F)$ such that it is very weakly dominant for any agent to report her candidate locations if it is a single point. Let i be an agent with candidate locations $K_i = [a_i, b_i]$ and p be the outcome of the mechanism for some profile of candidate locations (K_i, K_{-i}). Then, if p_a and p_b are the outcomes of $\mathcal{M}$ for the profiles (a_i, K_{-i}) and (b_i, K_{-i}), respectively,

$$\text{maxRegret}_i(p) = \max(|a_i - p| - |a_i - p_a|, |b_i - p| - |b_i - p_b|).$$

Proof. From (3.1) we know that

$$\text{maxRegret}_i(p) = \max_{\ell_i \in K_i} \max_{\sigma_i \in \Delta(\Sigma_i)} C(\ell_i, p) - C(\ell_i, F(\sigma_i(K_i), s_{-i}(K_{-i}))),$$

where $s_{-i}(K_{-i})$ is some set of strategies played by the others.

Now, since it very weakly dominant for any agent to report her true candidate locations if it is a single point in $\mathcal{M}$, we have that

$$\text{maxRegret}_i(p) = \max_{\ell_i \in K_i} C(\ell_i, p) - C(\ell_i, p_\ell)$$

where p_ℓ is the outcome of $\mathcal{M}$ for the profile $(\ell, s_{-i}(K_{-i}))$.

Given this, let us assume that the claim is false. This implies that there exists a $c \in (a_i, b_i)$ such that $C(c, p) - C(c, p_c) > C(a_i, p) - C(a_i, p_a)$ and $C(c, p) - C(c, p_c) > C(b_i, p) - C(b_i, p_b)$. However, we will show that this is impossible in both the cases below.

Case 1: $c \leq p$**.** In this case, we have,

$$\begin{aligned}
C(c,p) - C(c,p_c) &= |c-p| - |c-p_c| \\
&= (p-a_i) + (a_i - c) - |c-p_c| \\
&= (p-a_i) - |a_i - p_a| + (a_i - c) + |a_i - p_a| - |c - p_c| \\
&= C(a_i,p) - C(a_i,p_a) + (a_i - c) + |a_i - p_a| - |c-p_c| \\
&\leq C(a_i,p) - C(a_i,p_a) + (a_i - c) + |a_i - p_c| - |c-p_c| \\
&\qquad \big(\text{using very weak dominance for single reports}\big) \\
&\leq C(a_i,p) - C(a_i,p_a).
\end{aligned}$$

Case 2: $c > p$**.** We can handle this analogously as in Case 1 and show that $C(c,p) - C(c,p_c) \leq C(b_i,p) - C(b_i,p_b)$. □

Claim 20. Let p_{opt} be the minimax optimal solution associated with $\mathcal{I} = [a_1, b_1] \times \ldots \times [a_n, b_n]$ for the avgCost objective. Then, if p is a point such that $|p - p_{opt}| = d$ and $k = \lfloor \frac{n}{2} \rfloor$, then

$$\frac{n-2k}{n} \cdot d \leq \text{maxRegret}(p, \mathcal{I}) - \text{OMV}_{AC}(\mathcal{I}) \leq d.$$

Proof. If $\{L_i\}_{i\in[n]}$ and $\{R_i\}_{i\in[n]}$ denote the sorted order of the sets $\{a_i\}_{i\in[n]}$ and $\{b_i\}_{i\in[n]}$, respectively, then from Lemma 16 we know that for any point p the maximum regret associated with p can written as $\max(obj_1^{AC}(p), obj_2^{AC}(p))$, where

- $obj_1^{AC}(p) = \frac{1}{n}\left(2\sum_{i=j}^{k}(R_i - p) + (n-2k)(R_{k+1} - p)\right)$, where j is the smallest index such that $R_j > p$ and $j \leq k$ (if no such j, then set $j = k+1$)
- $obj_2^{AC}(p) = \frac{1}{n}\left(2\sum_{i=k+2}^{h}(p - L_i) + (n-2k)(p - L_{k+1})\right)$, where h is the largest index such that $L_h < p$ and $h \geq k+2$ (if no such h, then set $h = k+1$).

Additionally, if p_{opt} is the minimax optimal solution, we know from Claim 17 that $\text{OMV}_{AC}(\mathcal{I}) = obj_1^{AC}(p_{opt}) = obj_2^{AC}(p_{opt})$.

Now, let us assume without loss of generality that $p < p_{opt}$. Then, it clear that from above that $obj_1^{AC}(p) > obj_1^{AC}(p_{opt})$ and that $obj_2^{AC}(p) < obj_2^{AC}(p_{opt})$. So considering $obj_1^{AC}(p)$, we have

$$obj_1^{AC}(p) = \frac{1}{n}\left(2\sum_{i=j}^{k}(R_i - p) + (n-2k)(R_{k+1} - p)\right)$$

$$
\begin{aligned}
&= \frac{1}{n}\left(2\sum_{i=j}^{j'-1}(R_i - p) + 2\sum_{i=j'}^{k}(R_i - p) + (n-2k)(R_{k+1} - p_{opt} + p_{opt} - p)\right) \\
&\hspace{20em} \text{(where } j' \geq j \text{ is the smallest index such that } R_{j'} > p_{opt}\text{)} \\
&\leq \frac{1}{n}\left(2\sum_{i=j}^{j'-1}(p_{opt} - p) + n \cdot obj_1^{AC}(p_{opt}) + 2\sum_{i=j'}^{k}(p_{opt} - p) + (n-2k)(p_{opt} - p)\right) \\
&\hspace{5em} \left(\text{since } R_i \leq p_{opt} \text{ for } i \leq j'-1 \text{ and } n \cdot obj_1^{AC}(p_{opt}) = 2\textstyle\sum_{i=j'}^{k}(R_i - p) + (n-2k)(R_{k+1} - p_{opt})\right) \\
&\leq \frac{1}{n}\left(2kd + (n-2k)d + n \cdot obj_1^{AC}(p_{opt})\right) \qquad (p_{opt} - p \leq d \text{ and } j \geq 1) \\
&= d + \mathrm{OMV}_{AC}(\mathcal{I}).
\end{aligned}
$$

To see the upper bound, it is easy to see from above that one can set $j = k+1$ and hence we would have that $obj_1^{AC}(p) - \mathrm{OMV}_{AC}(\mathcal{I}) \geq \frac{(n-2k)}{n}(p_{opt} - p) = \frac{(n-2k)}{n} \cdot d$. $\square$

B.4 Minimax optimal solution for maxCost

Given the candidate locations $[a_i, b_i]$ for all i, where, for some $\delta \in [0, B]$, $b_i - a_i \leq \delta$, here we are concerned with computing the minimax optimal solution p_{opt} such that $p_{opt} = \arg\min_{p \in [0,B]} \max_{I \in \mathcal{I}} \left(S(I, p) - \min_{p' \in [0,B]} S(I, p')\right)$, where $\mathcal{I} = [a_1, b_1] \times \ldots \times [a_n, b_n]$ and S is the maximum cost function. Note that from the discussion in Section 3.5 we know that if $I = (\ell_1, \ldots, \ell_n)$ (as stated previously, we can assume without loss of generality that the l_is are in sorted order) is a valid input in $\mathcal{I}$, then $\min_{p' \in [0,B]} S(I, p') = S(I, p_I)$, where $p_I = \frac{\ell_1 + \ell_n}{2}$. Therefore, we can rewrite the definition of p_{opt} as $p_{opt} = \arg\min_{p \in [0,B]} \max_{I \in \mathcal{I}} S(I, p) - S(I, p_I)$.

Next, we prove the following lemma.

Lemma 43. *Given a point p, the maximum regret associated with p for the maximum cost objective can written as $\max(obj_1^{MC}(p), obj_2^{MC}(p))$, where*

- $obj_1^{MC}(p) = \frac{R_1 + R_n}{2} - p$
- $obj_2^{MC}(p) = p - \frac{L_1 + L_n}{2}$.

Proof. Consider the maximum regret associated with locating the facility at p, which is given by $\max_{I \in \mathcal{I}} S(I, p) - S(I, p_I)$, and the sets $\mathcal{I}'_1, \mathcal{I}'_2$, where $\mathcal{I}'_1 \subseteq \mathcal{I}$ is the set of all inputs such that for every $I'_1 = (\ell_1, \ldots, \ell_n)$ that belongs to $\mathcal{I}'_1$, $\frac{\ell_1 + \ell_n}{2} \geq p$ and $\mathcal{I}'_2 \subseteq \mathcal{I}$ is the set of all inputs such that for every $I'_2 = \langle \ell'_1, \ldots, \ell'_n \rangle$ that belongs to $\mathcal{I}'_2$, $\frac{\ell'_1+\ell'_n}{2} < p$. Since $\mathcal{I} = \mathcal{I}'_1 \cup \mathcal{I}'_2$, we can rewrite maximum regret associated with locating the facility at p

$$\max\{obj_1^{MC}(p), obj_2^{MC}(p)\},$$

where $obj_1^{MC}(p) = \max_{I'_1 \in \mathcal{I}'_1} S(I'_1, p) - S(I'_1, p_{I'_1})$ and $obj_2^{MC}(p) = \max_{I'_2 \in \mathcal{I}'_2} S(I'_2, p) - S(I'_2, p_{I'_2})$. So, now, let us consider each of these terms separately in the cases below.

Case 1: $obj_1^{MC}(p)$. Let us consider an arbitrary input $I'_1 = \langle \ell_1, \ldots, \ell_n \rangle$ that belongs to $\mathcal{I}'_1$ (if $\mathcal{I}'_1 = \emptyset$, then we define $\max_{I'_1 \in \mathcal{I}'_1} S(I'_1, p) - S(I'_1, p_{I_1}) = 0$). Now, the regret associated with I'_1 is given by

$$\begin{aligned}
\text{regret}(p, I'_1) &= S(I'_1, p) - S(I'_1, p_{I'_1}) \\
&= \max\{|\ell_1 - p|, |\ell_n - p|\} - \left|\ell_n - \frac{\ell_1 + \ell_n}{2}\right| \\
&= (\ell_n - p) - \left(\ell_n - \frac{\ell_1 + \ell_n}{2}\right) && \left(\text{as } p \le \tfrac{\ell_1+\ell_n}{2}\right) \\
&= \frac{\ell_1 + \ell_n}{2} - p.
\end{aligned}$$

Therefore, making use of the fact that $\ell_i \in [L_i, R_i]$ (this is proved in Claim 16, which is in Appendix B.3), we have that

$$\begin{aligned}
obj_1^{MC}(p) &= \max_{I'_1 \in \mathcal{I}'_1} S(I'_1, p) - S(I'_1, p_{I'_1}) \\
&= \max_{I'_1 \in \mathcal{I}'_1} \left(\frac{\ell_1 + \ell_n}{2} - p\right) \\
&= \frac{R_1 + R_n}{2} - p.
\end{aligned} \tag{B.6}$$

Case 2: $obj_2^{MC}(p)$**.** Like in the previous case, consider an arbitrary input $I'_2 = \langle \ell'_1, \ldots, \ell'_n \rangle$ that belongs to $\mathcal{I}'_2$ (if $\mathcal{I}'_2 = \emptyset$, then we define $\max_{I'_2 \in \mathcal{I}'_2} S(I'_2, p) - S(I'_2, p_{I'_2}) = 0$). Now, doing a similar analysis as in Case 1, we will see that the regret associated with I'_2 is given by $p - \frac{\ell'_1+\ell'_n}{2}$. Hence, again, making use of the fact that $\ell_i \in [L_i, R_i]$ (see Claim 16), we have that

$$obj_2^{MC}(p) = \max_{I'_2 \in \mathcal{I}'_2} S(I'_2, p) - S(I'_2, p_{I'_2})$$

$$= \max_{I'_2 \in \mathcal{I}'_2} \left(p - \frac{\ell'_1 + \ell'_n}{2} \right)$$

$$= p - \frac{L_1 + L_n}{2}. \tag{B.7}$$

Combining (B.6) and (B.7), we have our lemma. □

Equipped with the lemma proved above, we can now find the minimax optimal solution for the maximum cost objective.

Proposition 44. *If p_{opt} is the minimax optimal solution associated with the maximum cost objective, then $p_{opt} = \frac{L_1+R_1+L_n+R_n}{4}$.*

Proof. Note that the statement immediately follows if we prove that for a minimax optimal solution, p_{opt}, $obj_1^{MC}(p_{opt}) = obj_2^{MC}(p_{opt})$. And the latter proposition is easy to see, for if $obj_1^{MC}(p_{opt}) \neq obj_2^{MC}(p_{opt})$, then using Lemma 43 we can see that the maximum regret associated with p_{opt} is greater than $\frac{R_1+R_n-L_1-L_n}{4}$, whereas the maximum regret associated with the point $\frac{L_1+R_1+L_n+R_n}{4}$ is exactly $\frac{R_1+R_n-L_1-L_n}{4}$. □

Appendix C

Omitted material from Chapter 4

C.1 Additional claims

Claim 21. For $i \in [n]$, let $a_i \in \mathcal{N}$ and let v_i be the valuation function of a_i, where v_i is a unit-sum valuation function and for all $h \in A_i$ such that $rank(a_i, h) = 1$, $v_i(h) < \frac{1}{n^{1/3}}$. Then, for all $j \in [k]$, where $k = \lfloor \sqrt[3]{n}/2 \rfloor$, and $h \in A_i$ such that $rank(a_i, h) = j$, $v_i(h) \geq \frac{1}{2n}$

Proof. Let $t_1 = \frac{1}{n^{1/3}}$. For any $j \in [k]$ and $h \in A_i$ such that $rank(a_i, h) = j$, we will show that $v_i(h) \geq \frac{1-(j-1)t_1}{n-(j-1)}$. To see this, consider the smallest j^* such that for $h^* \in A_i$ with $rank(a_i, h^*) = j^*$ and $v_i(h^*) < \frac{1-(j^*-1)t_1}{n-(j^*-1)}$. Note that $j^* \geq 2$, since v_i is a unit-sum valuation function and hence $v_i(h) \geq \frac{1}{n}$ for all $h \in A_i$ such that $rank(a_i, h) = 1$.

Next, note that, for any $\ell \in \{2, \ldots, n\}$ and $U_\ell = \{h \in A_i \mid rank(a_i, h) \leq \ell - 1\}$, if there exists an $h \in A_i$ such that $rank(a_i, h) = \ell$, then it follows from the definition of rank that $|U_\ell| = \ell - 1$. Therefore, using this below, we have that,

$$\begin{aligned}
\sum_{h \in A_i} v_i(h) &= v_i(h^*) + \sum_{h \in U_{j^*}} v_i(h) + \sum_{A_i \setminus (U_{j^*} \cup \{h^*\})} v_i(h) \\
&\leq v_i(h^*) + (j^* - 1) \cdot t_1 + (n - j^*) \cdot v_i(h^*) \\
&< (j^* - 1) \cdot t_1 + (n - (j^* - 1)) \cdot \frac{1 - (j^* - 1)t_1}{n - (j^* - 1)} \\
&= 1.
\end{aligned}$$

However, this in turn contradicts the fact that v_i is a unit-sum valuation function.

Therefore, since $t_1 \leq \frac{1}{2k}$, for all $j \in [k]$ and $h \in A_i$ such that $rank(a_i, h) = j$, we have that

$$v_i(h) \geq \frac{1-(j-1)t_1}{n-(j-1)} \geq \frac{1-k \cdot t_1}{n-(j-1)} \geq \frac{1-k \cdot \frac{1}{2k}}{n-(j-1)} \geq \frac{1}{2n}.$$

□

Claim 22. Given an instance $\mathcal{I} = (\mathcal{N}, \mathcal{H}, \mathcal{P} = (P_1, \ldots, P_n))$, a matching μ is a

i) rank-maximal matching w.r.t. $\mathcal{I}$ if and only if it is a priority-$\boldsymbol{p}$ matching, where $\boldsymbol{p} = (p_1, \ldots, p_n)$ and, for $j \in [n]$, $p_j = n^{2(n-j+1)}$.

ii) max-cardinality rank-maximal matching w.r.t. $\mathcal{I}$ if and only if it is a priority-$\boldsymbol{p}$ matching, where $\boldsymbol{p} = (p_1, \ldots, p_n)$ and, for $j \in [n]$, $p_j = n^{2n} + n^{2(n-j)}$.

iii) fair matching w.r.t. $\mathcal{I}$ if and only if it is a priority-$\boldsymbol{p}$ matching, where $\boldsymbol{p} = (p_1, \ldots, p_n)$ and, for $j \in [n]$, $p_j = 4n^{2n} - 2n^{(j-1)}$.

Proof. **i).** ($\Rightarrow$) Suppose μ is a priority-$\boldsymbol{p}$ matching (which in turn implies that it is the max-weight matching in $\mathbb{G}_{\mathcal{I}}$ with weights $\{w_e\}_{e \in \mathcal{E}}$, where, for an edge $e = (a_i, h_j)$ and $r = rank(a_i, h_j)$, $w_e = p_r$), but is not rank-maximal w.r.t. $\mathcal{I}$. Let μ' be a rank-maximal matching w.r.t. $\mathcal{I}$, and the let the signatures of μ and μ' be $(s_1, \ldots, s_n)$ and $(s'_1, \ldots, s'_n)$, respectively. Since μ is not rank-maximal, consider the smallest $j \in [n]$ such that $s_i = s'_i$ for all $i < j$ and $s_j < s'_j$. Next, note that both μ and μ' are matchings in $\mathbb{G}_{\mathcal{I}}$, and that the weight of a j-th ranked edge is $n^{2(n-j+1)}$. This in turn implies that weight of μ' is greater than weight of μ, since $n^{2(n-j+1)} > n \cdot n^{2(n-j)}$, or in other words that taking a j-th ranked edge in $\mathbb{G}_{\mathcal{I}}$ with weights $\{w_e\}_{e \in \mathcal{E}}$ is more beneficial than taking any number of edges of lower rank. However, this contradicts the fact that μ is the max-weight matching in $\mathbb{G}_{\mathcal{I}}$.

($\Leftarrow$) Suppose μ is a rank-maximal matching, but is not a priority-$\boldsymbol{p}$ matching. This implies there is another matching μ' such that μ' has a higher weight than μ in $\mathbb{G}_{\mathcal{I}}$ with weights $\{w_e\}_{e \in \mathcal{E}}$, where, for an edge $e = (a_i, h_j)$ and $r = rank(a_i, h_j)$, $w_e = p_r$. Also note that μ' cannot be rank-maximal, since all rank-maximal matchings will have the same weight in $\mathbb{G}_{\mathcal{I}}$ with weights $\{w_e\}_{e \in \mathcal{E}}$. Given this, it is easy to see that we have a contradiction given our choice of p_js and the fact that μ is rank-maximal.

ii). ($\Rightarrow$) Suppose μ is a priority-$\boldsymbol{p}$ matching. First we will argue that μ is a max-cardinality matching in $\mathbb{G}_{\mathcal{I}}$. To see this, suppose not. This implies there is another matching μ' such that $|\mu'| > |\mu|$. Next, note that if $W[\mu]$ denotes the weight of μ in $\mathbb{G}_{\mathcal{I}}$ with weights $\{w_e\}_{e \in \mathcal{E}}$, then $W[\mu] \leq |\mu| \cdot p_1$. Also, note that we have, $W[\mu'] \geq |\mu'| \cdot n^{2n} > |\mu| \cdot p_1 \geq W[\mu]$, where the second inequality follows from the fact that $|\mu'| - |\mu| \geq 1$ and $|\mu| \leq n$. However, this in turn contradicts the fact that μ is a priority-$\boldsymbol{p}$ matching.

Now that we have the established that μ is a max-cardinality matching, we need to show that it is rank-maximal (among max-cardinality matchings). And the proof of this can be obtained by proceeding as in the corresponding case in i) above.

($\Leftarrow$) The proof of this can be obtained by proceeding as in the corresponding case in i) above.

iii). This case can be handled as in ii) above, with the only difference being that the signature in this case is the $(n+1)$-tuple $(\sum_{i=1}^{n} s_i, -s_n, -s_{n-1}, \ldots, -s_1)$. $\square$

Claim 23. Given an instance $\mathcal{I} = (\mathcal{N}, \mathcal{H}, \mathcal{P})$ and priorities $\boldsymbol{p} = (p_1, \ldots, p_n)$, where $\boldsymbol{p} \in \mathbb{P}$, let μ denote a Pareto optimal matching or a priority-$\boldsymbol{p}$ matching w.r.t. $\mathcal{I}$. If B_i denotes the set of agents matched to a rank-i edge in μ, k is a positive integer that is at most $\lfloor n/2 \rfloor$, and $B = \cup_{i=1}^{k} B_i$, then $|B| \geq \min\{k, |\mu|\}$.

Proof. Let $B' = \cup_{i=k+1}^{n} B_i$ be the set of agents who are matched to an object of rank at least $k+1$ in μ. In order to prove our claim, let us first consider the case when $k \geq |\mu|$. For this case we will show that $|B'| = 0$, thus implying that $|B| = \mu$. To see this, suppose $|B'| \geq 1$. Without loss of generality, let $a_1 \in |B'|$ and let $O_1^k = \{h \in A_1 \mid rank(a_1, h) \leq k\}$. Since μ is a Pareto optimal or priority-$\boldsymbol{p}$ matching, all the objects in O_1^k are matched to some agent in $\mathcal{N} \setminus \{a_1\}$, because if there exists an unmatched, say, $h_1 \in O_1^k$, then $\mu \setminus \{a_1, \mu(a_1)\} \cup \{a_1, h_1\}$ Pareto dominates μ and has the same size as μ. However, this in turn implies $|\mu| \geq (|O_1^k| + 1) = k + 1$, a contradiction.

Next, let us consider the case when $k < |\mu|$. To prove our claim, let us assume for the sake of contradiction that $|B| < k$, which in turn, along with the fact that $k < |\mu|$, implies that $|B'| \geq 1$. Let $Y = |B'|$ and w.l.o.g. let us assume that $\{a_1, \ldots a_Y\}$ are the agents in B'. Note that, for $i \in [Y]$, a_i finds at least $k+1$ objects acceptable since they are matched to an object of rank at least $k+1$.

Now, consider any $i \in [Y]$ and let $O_i^k = \{h \in A_i \mid rank(a_i, h) \leq k\}$. As argued above for the case when $k \geq |\mu|$, since μ is a Pareto optimal matching or a priority-$\boldsymbol{p}$ matching, all the objects in O_i^k are matched to some agent. Additionally, also note that, when combined with the previous observation, there is at least one object, say, $h_i^s \in O_i^k$ such that it is matched to an agent, say, $a_p \in B'$ such that $rank(a_p, h_i^s) \geq k+1$ (because otherwise, $|B| \geq k$, a contradiction); we will refer to h_i^s as a_i's special object. Given all the observations above, we will now argue that if μ is such that $|B| < k$, then there exists a matching μ' such that μ' Pareto dominates μ and has the same size as μ.

To see this, consider the graph $\mathbb{G}' = (\{a_1, \cdots, a_Y\}, \mathcal{E}')$, where there is a directed edge from a_i to a_j if $\mu(a_j) = h_i^s$ (i.e., if a_j is matched to a_i's special object). Since, for every $i \in [Y]$, a_i has a special object h_i^s and h_i^s is matched to one of the agents in $\{a_1, \cdots, a_Y\} \setminus \{a_i\}$, note that this graph has to have a cycle. So, now, consider this cycle and implement the trade indicated by this cycle—meaning, if (a_i, a_j) is an edge in this cycle, then allocate $\mu(a_j)$ to a_i. Note that the resulting matching, say, μ' has the same size as μ and also Pareto dominates μ (since every agent a_i in this cycle gets an object that is in O_i^k), which in turn contradicts the fact that μ is a Pareto optimal or priority-$\boldsymbol{p}$ matching. □

Claim 24. Given an instance $\mathcal{I} = (\mathcal{N}, \mathcal{H}, \mathcal{P})$, let μ_1 be any arbitrary matching in $\mathbb{G}_{\mathcal{I}}$, and μ_2 denote the matching in $\mathbb{G}_{\mathcal{I}}$ that matches as many agents as possible with an edge of rank at most k, where $k = \lfloor \sqrt[3]{n}/2 \rfloor$. If L_i and X_i' denote the set of agents matched to a rank-i edge in μ_1 and μ_2, respectively, and $X' = \cup_{i=1}^{k} X_i'$, then $|X'| \geq \min\left\{k, \left(\sum_{i=k+1}^{n} |L_i|\right)\right\}$.

Proof. Let $Y = \sum_{i=k+1}^{n} |L_i|$, i.e., the number of agents who are matched to an object of rank at least $k+1$ in μ_1. Since all the agents in $\cup_{i=k+1}^{n} L_i$ find at least $k+1$ objects acceptable, this implies that there is a matching of size $z = \min\{k, Y\}$ in $\mathbb{G}_{\mathcal{I}}$ where all the agents are matched to a rank-i edge, where $i \in [k]$. Now, since μ_2 is the matching in $\mathbb{G}_{\mathcal{I}}$ that matches as many agents as possible with an edge of rank at most k, $X' \geq z$. □

C.2 Missing proofs from Section 4.3.2

C.2.1 Proof of Theorem 31

Theorem 31. *Given an instance $\mathcal{I} = (\mathcal{N}, \mathcal{H}, \mathcal{P} = (P_1, \ldots, P_n))$ and a vector of priorities $\boldsymbol{p} = (p_1, \ldots, p_n)$, where $\boldsymbol{p} \in \mathbb{P}$, Algorithm 5 asks one non-adaptive query per (agent, object) and returns a priority-$\boldsymbol{p}$ matching that achieves an $O(n^{2/3})$-approximation to the optimal welfare among all priority-$\boldsymbol{p}$ matchings for the case when agents have unit-sum valuations.*

To prove this, we use the same notations and terminologies that were introduced for the proof of Theorem 29, except that now these are defined with respect to priority-$\boldsymbol{p}$ matchings. Next, we prove the following lemma, which is almost identical to Lemma 30.

Lemma 45. *Let μ be the matching that is computed in line 12 in Algorithm 5. Then, $SW(\mu_H^*) \leq n^{2/3} \cdot SW(\mu)$.*

Proof. This can be proved by proceeding exactly as in the proof of Lemma 30. □

Proof of Theorem 31. First, it is easy to see that for the given priority vector $\boldsymbol{p} \in \mathbb{P}$ the matching returned by Algorithm 5 is a priority-$\boldsymbol{p}$ matching. Next, let S be the set of agents who answered "Yes" w.r.t. t_1 (i.e., all these agent have a value of at least t_1 for (one of) their first choice object(s)). Also, let B_i be the number of agents who are matched to (one of) their i-th choice object(s) in μ. Note that since μ is a priority-$\boldsymbol{p}$ matching, we know that $|B_i| = |H_i| + |L_i|$ for all $i \in [n]$ (since all priority-$\boldsymbol{p}$ matchings have the same signature). Additionally, we also know that $\text{SW}(\mu) \geq |B_1| \cdot \frac{1}{n}$, since the agents have unit-sum valuations.

Now, if $|S| \geq 1$, then we have that,

$$\begin{aligned}
\frac{\text{SW}(\mu^*)}{\text{SW}(\mu)} &= \frac{\text{SW}(\mu^*_H) + \text{SW}(\mu^*_L)}{\text{SW}(\mu)} \\
&= \frac{\text{SW}(\mu^*_H) + \text{SW}(\mu^*_{L_1}) + \sum_{i=2}^{n} \text{SW}(\mu^*_{L_i})}{\text{SW}(\mu)} \\
&\leq \frac{n^{2/3} \cdot \text{SW}(\mu) + |B_1| \cdot t_1 + n \cdot t_2}{\text{SW}(\mu)} \\
&\leq \frac{n^{2/3} \cdot \text{SW}(\mu)}{\text{SW}(\mu)} + \frac{|B_1| \cdot t_1}{|B_1| \cdot \frac{1}{n}} + \frac{n \cdot t_2}{t_1} \\
&\leq n^{2/3} + n^{2/3} + n^{2/3}, \qquad \text{(C.1)}
\end{aligned}$$

where the first inequality by using Lemma 45 and the fact that $L_1 \leq B_1$, second inequality follows from the fact that $\text{SW}(\mu) \geq t_1$, since $|S| \geq 1$. and the final inequality follows from our choice of t_1 and t_2.

On the other hand, if $|S| = 0$, then, first, let $k = \lfloor \sqrt[3]{n}/2 \rfloor$. Next, note that every agent values their first choice at a value less than t_1. This in turn implies that, since their valuations are unit-sum, for $j \in [k]$, their value for a rank-j object is at least $\frac{1}{2n}$ (see Claim 21 in Appendix C.1 for a proof). Additionally, let $B = \cup_{i=1}^{k} B_i$ and $B' = \cup_{i=k+1}^{n} B_i$; from Claim 23 we know that $|B| \geq \min\{k, |\mu|\}$. Given this, we have,

$$\begin{aligned}
\frac{\text{SW}(\mu^*)}{\text{SW}(\mu)} &= \frac{\text{SW}(\mu^*_H) + \text{SW}(\mu^*_L)}{\text{SW}(\mu)} \qquad \text{(C.2)} \\
&= \frac{\text{SW}(\mu^*_H) + \sum_{i=1}^{k} \text{SW}(\mu^*_{L_i}) + \sum_{i=k+1}^{n} \text{SW}(\mu^*_{L_i})}{\text{SW}(\mu)} \\
&\leq \frac{\text{SW}(\mu^*_H) + \left(\sum_{i=1}^{k} |B_i|\right) \cdot t_1 + \left(\sum_{i=k+1}^{n} |B_i|\right) \cdot t_{k+1}}{\text{SW}(\mu)}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{n^{2/3} \cdot \mathrm{SW}(\mu) + |B| \cdot t_1 + |B'| \cdot t_{k+1}}{\mathrm{SW}(\mu)} \\
&\leq \frac{n^{2/3} \cdot \mathrm{SW}(\mu)}{\mathrm{SW}(\mu)} + \frac{|B| \cdot t_1}{|B|/(2n)} + \frac{|B'| \cdot t_{k+1}}{|B|/(2n)} \\
&\leq n^{2/3} + 2n^{2/3} + 8n^{2/3}, \qquad \text{(C.3)}
\end{aligned}
$$

where the first inequality follows from using Lemma 45 and the fact that $L_i \leq B_i$ for all $i \in [n]$, the second inequality follows from Claim 5, and the final inequality follows since $|B| \geq \min\{k, |\mu|\}$ and $t_{k+1} \leq 2/n$.

Finally, combining (C.1) and (C.2) gives us our theorem. □

C.3 Additional discussions

C.3.1 Power of ordinal algorithms

In this section we look at the power of ordinal algorithms—meaning we want to understand the worst-case loss in welfare experienced by an ordinal algorithm (i.e., an algorithm which only uses the ordinal information given by the agents). We argue that when agents have unit-sum valuations functions, any deterministic algorithm $\mathcal{A}$ is such that $\mathcal{L}(\mathcal{A}) \in \Omega(n^2)$. The proof of this is similar to the proof of result by Amanatidis et al. where they show that the distortion of any deterministic ordinal algorithm is $\Omega(n^2)$ [Ama+21, Thm. 1]. Also, just like in Amanatidis et al. [Ama+21, Thm. 1], this bound is asymptotically tight.

Theorem 26. *Let X denote one of the properties in the set {Pareto-optimal, rank-maximal, max-cardinality rank-maximal, and fair}. Let $\mathcal{A}$ be a deterministic ordinal algorithm that always produces a matching that satisfies property X. If there are n agents with unit-sum valuation functions, then $\mathcal{L}(\mathcal{A}) \in \Omega(n^2)$. Moreover, this bound is asymptotically tight.*

Proof (sketch). The proof here follows almost directly from the proof of Theorem 1 in the paper by Amanatidis et al. [Ama+21] where they show that the distortion of any deterministic ordinal algorithm is $\Omega(n^2)$.

The main observation to note is that for the instance they construct if there are j pairs of agents and if an agent in the i-th pair gets the top-choice object a, then any matching $\mathcal{M}$ that matches the other agent in the i-th pair to their second choice b_i, one of the agents in pair $\ell \in [j] \setminus \{i\}$ to their second choice b_ℓ, and the remaining agents to their highest possible choice (i.e., for pair $\ell \in [j] \setminus \{i\}$, allocate object $c_{\ell-1}$ to the agent who has not received their second choice) is Pareto-optimal/rank-maximal/max-cardinality rank-maximal/fair.

Given this, one can construct valuation functions as described in their proof and see that an ordinal algorithm cannot distinguish between matchings that have welfare of at least $n/4$ and ones which have a welfare of at most $1/n$, thus resulting in a lower bound of $\Omega(n^2)$.

Finally, to see that the bound is tight, first note that when agents have unit-sum valuation functions, as long as there is at least one agent who is matched to their top-choice object, $\mathcal{L}(\mathcal{A}) \in O(n^2)$, since the maximum social welfare achievable is n, and the agent who gets their top-choice has a value of at least $1/n$ for their top-choice object. Now, in order to achieve this, one can run any Pareto-optimal/rank-maximal/max-cardinality rank-maximal/fair algorithm and the matching returned by any such algorithm has at least one agent matched to a rank-1 object (see Claim 23 in Appendix C.1 for a proof). $\square$

One can derive a similar lower bound for unit-range valuations and show that it is $\Omega(n)$. This can be done by making similar observations as in the proof of Theorem 26. The only part that needs to be modified is in the way the valuation functions are defined; all we need to do here is to define the value of each top-choice object to be 1 and the least preferred object to be 0. Moreover, this is again asymptotically tight because of the exact same reason mentioned above for unit-sum valuations. The only difference is that when agent have unit-range valuations, the value of the top-choice object is 1, thus resulting in an $O(n)$ bound.

Theorem 46. *Let X denote one of the properties in the set {Pareto-optimal, rank-maximal, max-cardinality rank-maximal, and fair}. Let $\mathcal{A}$ be a deterministic ordinal algorithm that always produces a matching with property X. If there are n agents with unit-range valuation functions, then $\mathcal{L}(\mathcal{A}) \in \Omega(n)$. Moreover, this bound is asymptotically tight.*

C.3.2 The unit-range case

In this section we discuss the case when agents have unit-range valuations. Note that the adaptive algorithm presented in Section 4.3.1 works for both the unit-sum and unit-range case. Therefore, here we look at the non-adaptive case, in particular focussing on the case when an algorithm is allowed to ask at most one query per (agent, object) pair.

Input: an instance $\mathcal{I} = (\mathcal{N}, \mathcal{H}, \mathcal{P} = (P_1, \ldots, P_n))$ and priorities $\boldsymbol{p} = (p_1, \ldots, p_n)$
Output: returns a Pareto optimal matching when $p_i = 0$ for all $i \in [n]$, priority-$\boldsymbol{p}$ matching otherwise

$\mathbb{G}_{\mathcal{I}} = (\mathcal{N} \cup \mathcal{H}, \mathcal{E}) \leftarrow$ graph induced by $\mathcal{I}$
$t_1 \leftarrow 1$
$t_i \leftarrow \frac{1}{\sqrt{n}}$, for all $i \in \{2, \ldots, n\}$
for $e = (a_i, h_r) \in \mathcal{N} \times \mathcal{H}$ **do**
    $r \leftarrow \text{rank}(a_i, h_j)$
    if $\mathcal{Q}(a_i, h_j, t_r)$ **then**
        $w_e \leftarrow p_r + t_r$
    else
        $w_e \leftarrow p_r$
    end if
end for
$\mu \leftarrow$ max-weight matching in $\mathbb{G}_{\mathcal{I}}$, where weights are $(w_e)_{e \in \mathcal{E}}$
if $p_i = 0$ for all $i \in [n]$ **then**
    $\mu \leftarrow$ with μ as the initial endowment, run modified top-trading cycles (TTC) algorithm by Saban and Sethuraman [SS13, Algorithm 1, Rule 2] and return the resulting matching.
end if
return μ

Algorithm 10: An $O(\sqrt{n})$-approximation algorithm for finding the optimal social welfare among Pareto optimal or priority-p matchings for the case when the agents have unit-range valuations.

C.3.2.1 Improving welfare when asking one query per (agent, object) pair

Here we show an algorithm that achieves an $O(\sqrt{n})$-approximation to the optimal welfare among all Pareto optimal or priority-$\boldsymbol{p}$ matchings when it is allowed to ask at most one query per (agent, object) pair. In the next section, we will show that this is asymptotically optimal. At a high-level, the algorithm is very similar to Algorithm 5, with the main difference being that instead of t_is defined there, where we used $t_1 = \frac{1}{n^{1/3}}$ and $t_i = \frac{1}{\min\{i, n^{1/3}\} \cdot n^{2/3}}$ for all $i \in \{2, \ldots, n\}$, here the values are more uniform and we use $t_1 = 1$ and $t_i = \frac{1}{\sqrt{n}}$ for all $i \in \{2, \ldots, n\}$. Note that the query $\mathcal{Q}(a_i, h_j, t_1)$, where h_j is such that $rank(a_i, h_j) = 1$, is not really useful since all the agents will answer "Yes" to this because of the fact that they have unit-range valuations. Nevertheless, we still use it in Algorithm 10 to make it clear that it is very similar to Algorithm 5.

Theorem 47. *Given an instance $\mathcal{I} = (\mathcal{N}, \mathcal{H}, \mathcal{P} = (P_1, \ldots, P_n))$, and a vector of priorities*

$\boldsymbol{p} = (p_1, \ldots, p_n)$, where $\boldsymbol{p} \in \mathbb{P} \cup \{(0, \ldots, 0)\}$, Algorithm 10 asks one non-adaptive query per (agent, object) pair and for the case when agents have unit-range valuations returns a

i) Pareto optimal matching that achieves an $O(\sqrt{n})$-approximation to the optimal welfare among all Pareto optimal matchings when $p_i = 0$ for all $i \in [n]$.

ii) priority-$\boldsymbol{p}$ matching that achieves an $O(\sqrt{n})$-approximation to the optimal welfare among all priority-$\boldsymbol{p}$ matchings when $\boldsymbol{p} \in \mathbb{P}$.

Proof. Just like we did to prove Theorem 29 let us introduce some notation. Given the priority vector $\boldsymbol{p}$, let μ^* denote the matching that achieves the optimal social welfare among Pareto optimal or priority-$\boldsymbol{p}$ matchings when the agents have unit-range valuations. Let H_i denote the set of agents who are matched to their i-th choice in μ^* and have value at least t_i for their partner in μ^*. Similarly, let L_i denote the set of agents who are matched to their i-th choice in μ^* and have value less than t_i for their partner in μ^*. We define $H = \cup_{i=1}^n H_i$, $L = \cup_{i=1}^n L_i$, and, for some $S \subseteq \mathcal{N}$, $\mu^*_S \subseteq \mu^*$ be the set of edges (a_i, h_j) such that $a_i \in S$ and $(a_i, h_j) \in \mu^*$.

Additionally, let μ' be the matching that is computed in line 12 in Algorithm 10. Note that if μ is the matching returned by Algorithm 10, then $\mu' = \mu$ when $\boldsymbol{p} \in \mathbb{P}$. Also, for the case when $p_i = 0$ for all $i \in [n]$, $\mathrm{SW}(\mu) \geq \mathrm{SW}(\mu')$, since μ is the matching returned by the modified TTC algorithm by Saban and Sethuraman [SS13, Algorithm 1, Rule 2] with initial endowments μ' and we know that the resulting matching is individually rational [SS13, Theorem 2].

Next, it is easy to see that the matching is Pareto optimal when $p_i = 0$ for all $i \in [n]$ and is a priority-$\boldsymbol{p}$ matching when $\boldsymbol{p} \in \mathbb{P}$. (This can be seen by proceeding exactly like in the second paragraph of the proof of Theorem 28.)

Finally, in order to bound the approximation ratio, we will directly bound the ratio of $\frac{\mathrm{SW}(\mu^*)}{\mathrm{SW}(\mu)}$. To do this, first note that we can proceed exactly like in the proof of Lemma 30 to see that $\mathrm{SW}(\mu^*_H) \leq \sqrt{n} \cdot \mathrm{SW}(\mu')$. This is so because the only difference is in the way the t_is are defined for all $i \in [n]$. Second, note that $\mathrm{SW}(\mu^*_L) \leq |\mu^*_L| \cdot \frac{1}{\sqrt{n}} \leq \sqrt{n}$, where the first inequality follows since $|L_1| = 0$ and every agent in L_i for $i \in \{2, \ldots, n\}$ has value at most $t_i = 1/\sqrt{n}$ for their good, and the second inequality follows since $|\mu^*_L| \leq n$. Therefore, we have that,

$$\frac{\mathrm{SW}(\mu^*)}{\mathrm{SW}(\mu)} = \frac{\mathrm{SW}(\mu^*_H) + \mathrm{SW}(\mu^*_L)}{\mathrm{SW}(\mu)} \leq \frac{\sqrt{n} \cdot \mathrm{SW}(\mu') + \sqrt{n}}{\mathrm{SW}(\mu)} \leq 2\sqrt{n}, \tag{C.4}$$

where the last inequality follows since $\mathrm{SW}(\mu) \geq 1$, as there is at least one agent who is matched to a rank-1 edge in μ (since μ is either a Pareto optimal or a priority-$\boldsymbol{p}$ matching and hence we can use Claim 23). $\square$

C.3.2.2 Lower bounds

Here we derive a lower bound that is similar to one for unit-sum valuations and show that any deterministic algorithm $\mathcal{A}$ that produces a Pareto-optimal/rank-maximal/max-cardinality rank-maximal/fair matching and that asks at most one query per (agent, object) pair has $\mathcal{L}(\mathcal{A}) \in \Omega(\sqrt{n})$. Note that this implies that Algorithm 10 is asymptotically optimal.

Theorem 48. *Let X denote one of the properties in the set {Pareto-optimal, rank-maximal, max-cardinality rank-maximal, and fair}. Let $\mathcal{A}$ be a non-adaptive deterministic algorithm that always produces a matching with property X and asks at most one query per (agent, object) pair. If there are n agents with unit-range valuation functions, then $\mathcal{L}(\mathcal{A}) \in \Omega(\sqrt{n})$.*

Proof (sketch). The proof of this is almost identical to the proof of Theorem 32, with the main difference being the way the utility functions $u_0, u_1, \ldots, u_8$ are defined. For this proof, we redefine $u_0, u_1, \ldots, u_8$ the following way, to ensure that they are unit-range valuation functions.

$$\begin{aligned}
&u_0 = (1, c_1^2, 0, \ldots, 0) && \\
&u_1 = (1, c_1, 0, \ldots, 0) && u_2 = (1, 1/2, 0, \ldots, 0) \\
&u_3 = (1, c_1 + \epsilon, 0, \ldots, 0) && u_4 = (1, 1/4 - \epsilon, 0, \ldots, 0) \\
&u_5 = (1, c_1^2, 0 \ldots, 0) && u_6 = (1, c_1 - \epsilon, 0, \ldots, 0) \\
&u_7 = (1, c_1^2, 0, \ldots, 0) && u_8 = (1, c_1/4, 0, \ldots, 0),
\end{aligned}$$

Given these definitions, one can now proceed exactly like in the proof of Theorem 32. Note that the proof will be simpler in this case because for unit-range valuations, the queries of the form $\mathcal{Q}(\cdot, \cdot, T_{i1})$ are not useful, since by definition all unit-range utility functions have value 1 for the most preferred good. $\square$

Appendix D

Omitted material from Chapter 5

D.1 Brief descriptions of various fair and efficient algorithms

Here we provide brief descriptions of the following algorithms that guarantee PO and at least EF1.

i) Adjusted winner protocol [BT96a; BT96b]: The adjusted winner protocol returns an EF1 and PO allocation for two agents and it proceeds the following way.

   - One of the players (say, player 1) is designated as the *winner* and the other player is the *loser*. Initially, all the goods are allocated to the winner.
   - Sort the goods by decreasing value of $v_\ell(g)/v_w(g)$, where $v_\ell(g)$ (respectively, $v_w(g)$) is the value that the loser (respectively, winner) attributes to good g.
   - At each step, among the goods that the winner has, consider the "best" good according to the order in step 2 (i.e., one with the highest value of $v_\ell(g)/v_w(g)$), and allocate it to the loser. Continue this process until the loser does not envy the winner by more than one good.

ii) Leximin solution [PR18]. The leximin solution returns a PMMS and PO allocation for two agents by finding an allocation that maximizes the minimum utility any agent gets, then, subject to that, maximizes the second minimum utility any agent gets, and so on.

iii) Maximum Nash Welfare solution [Car+16]. The maximum Nash welfare solution returns an EF1 and PO allocation for any number of agents by finding an allocation that maximizes the product of utilities of the agents.

iv) Fisher-market based algorithm [BKV18]. The algorithm by Barman, Krishnamurthy, and Vaish [BKV18] returns an EF1 and PO allocation for any number of agents. The algorithm is quite involved and requires several definitions, so we only provide a very brief and high-level description here. We refer the reader to Algorithm 1 in [BKV18] for a complete description.

  - The main idea in this algorithm is to start with a Pareto optimal allocation (in particular, a welfare maximizing allocation) that corresponds to a market equilibrium of a Fisher market. Following this, it performs a series of swaps between the agents ensuring that each intermediate allocation is an equilibrium outcome, and it does this until the allocation satisfies what is referred to as price envy-freeness up to one good (or a suitable approximation of it). The latter property entails that at the current prices spending of any agent i is at least as much as the spending of another agent j up to the removal of the highest priced good from agent j's bundle.