Problems in Computational Social Choice on Restricted Domains

by

Chinmay Sonar

A thesis submitted in the partial fulfillment
for the degree of

Master of Technology

in Computer Science
of

Indian Institute of Technology Gandhinagar

June 2019

Declaration

I, hereby declare that the investigation presented in the thesis has been carried out by me. The work is original and the work has not been submitted earlier as a whole or in part for a degree/diploma at this or any other Institution or University.

Chinmay Sonar

Neeldhara Misra
Assistant Professor
Department of Computer Science
Indian Institute of Technology Gandhinagar

Abstract

Problems in Computational Social Choice on Restricted Domains

by

Chinmay Sonar

Social Choice is the formalized study of the aggregation of individual preferences towards collective decision making. The task of checking the feasibility of the preference aggregation schemes in the real world context comes under Computational Social Choice (COMSOC). In this thesis, we consider the problems from the three pillars of COMSOC - Voting, Fair Division, and Matching under Preferences.

Several appealing aggregation schemes turn out to be computationally hard. The work from this thesis largely focuses on studying problems in these areas in the context of domain restrictions. We consider popular notions of domain restrictions such as Single-Peaked (SP), Single-Crossing (SC), and Euclidean preference domains. The motivation for studying structured domains is to explore if certain properties that are elusive for in the general case can be achieved by considering appropriate and practically relevant domain restrictions. For instance, some voting rules that are hard to compute in general become tractable on restricted domains. Another illustration is that some domains admit mechanisms that are not vulnerable to strategic behavior. We often find that simply projecting mechanisms for general domains to structured domains does not give us enough leverage, so the quest is usually to design new mechanisms that take advantage of the specific structure of the domain under consideration.

B **Voting:** Our work in voting mainly focuses on the Chamberlin Courant (CC) voting rule. Several fundamental problems such as winner determination has been shown to be hard on for CC rule. We focus on the restricted and nearly restricted domains for this problem. Our work provides a fine-grained analysis of the boundary between tractability and intractability using parametric analysis. We also consider the effect of small perturbations in election instance formally captured by the concept of the robustness radius and provide an XP algorithms along with the matching hardness results.

B **Stable Matching:** For many *hard* matching problems, we show that the hardness persists even the domain restrictions. We also provide an interesting addition to a class of instances which admit a unique stable matching (marriage for the bipartite case). We introduce a new variant of the matching problem called *'Matching Critical Sets'* and pin down the complexity of the same. Finally, we work on the special cases of *(a,b)-supermatch*. We also analyze the performance of the *Gale-Shapley* mechanism on instances with restricted domains through simulations.

B **Fair Division:** We study fair allocation of indivisible items arranged on a *path* under the constraint that only connected subsets of items may be allocated to the agents. We study the problem for both *Envyfreeness* (EF1) and *Equitability* (EQ1). We show that achieving EF1 or EQ1 in conjunction with well-studied measures of *e ciency* (such as Pareto optimality, non-wastefulness, maximum egalitarian or utilitarian welfare) is computationally hard even for *binary* valuations. On the algorithmic side, we show that for *any* fixed ordering of agents, an Eq1 allocation with the highest egalitarian welfare among all consistent allocations can be efficiently computed. For *structured* binary valuations, we obtain polynomial-time algorithms for non-wasteful and pareto optimal EF1 and EQ1 allocations.

. . . to my mother

Acknowledgements

I am indebted to Neeldhara Misra for being an amicable and sanguine advisor. It was her belief in me, and her unwavering support that made me realize my potential. I thank her even more for always being a friend for all our discussions over countless cups of coffee. I will forever admire her contagious optimism and compassion, and will never cease to be amazed by her modesty. I am grateful to her for being perpetually patient with me; talking with her always brought content for me. The work in this thesis would not have been possible without her extraordinary help and guidance. In the last four years, I have learned so many things that every aspect of my academic life and several aspects of my social life are influenced by her thoughts and actions.

I thank Rohit Vaish for introducing me to the interesting area of Fair Division. More so, I thank him for helping me to understand the aim of graduate program, emphasizing on the essential skills I should try to pick during my Ph.D.

I was fortunate to spend five beautiful years during the initial stages of IITGN and the growing CS department. The department gave me a chance to work personally with several professors when I was an undergraduate student. Specifically, I thank Anirban Dasgupta for guiding me not only with coursework and research but also for providing a rational comparison of career paths and eventually helping me choose the most suitable place for my Ph.D. I will always admire and be inspired by his profoundness and simplicity. Furthermore, I thank Manoj Gupta for being available for many discussions and being forthcoming and friendly. Thanks to Nipun Batra, who puts many efforts in his courses and for helping me during graduate applications; his energy and enthusiasm is worthy of being envious.

Prof. Nithin George introduced me to the Dual-degree Program and enthusiastically helped me during the enrolment. The master's program paved my way with eclectic experiences and helped me develop essential habits for being a good graduate student. I am so very thankful to him for all his assistance.

I am thankful to the industrious library staff at IITGN. Thanks for tolerating numerous requests for access to the articles, and providing them as soon as possible.

I thank IITGN for providing funding to travel for the conference, Algorithmic Decision Theory. Through Clemson University and IITGN, I got my first international research experience in the summer of 2017, which eventually but inevitably motivated me to opt for a research-oriented graduate program.

I want to thank all of my friends who were with me through my journey at IIT. Special thanks to Amogh Parab and P. R. Vaidyanathan for sharing a memorable final year journey. Thanks to Kshiteej Sheth, Vinod Ramakrishnan, Deepak Dhariwal, Pranav Kumar, Siddharth Sheshadri, and Akash Pallath for initiating, sustaining and concluding several discussions. Also, Vismay Vakharia, Harsh Modi and Pinank Patel arranged a few memorable gatherings last year.

Thanks to my group mates Chamanvir Kaur and Piyush Rathi, it was a pleasure sharing the fun of learning many theorems with them.

At last, I would like to thank my parents and my family who are truly the wind under my wings. It was their encouragement and support that allowed me to pursue research during my journey at IITGN. I thank my father for believing in me unconditionally and supporting all the choices I have made until now. Thanks to Aditi, my elder sister for listening to me for hours and cheering me up every-time I had a long day. I want to thank Kaustubh, Saurabh, Ashish and all my lovely cousins for being there whenever I turned to them. I simply don't have any way to thank my family enough for their invaluable presence my life, my memories, and my whole essence, hence, I would like to be indebted to them for my lifetime.

Contents

Contents ix

List of Figures xiii

List of Algorithms xv

1 Introduction 1

1.1 Social Choice 1

1.1.1 Voting 2

1.1.2 Matching 3

1.1.3 Fair Division 4

1.2 Restricted Domain in Social Choice - Background and motivation 5

1.3 Contributions and structure of the thesis 6

1.3.1 Part I - Voting 7

1.3.2 Part II - Matching 7

1.3.3 Part III - Fair Division 8

2 Preliminaries 11

2.1 Restricted Domain under consideration 12

2.2 Classical Complexity and reductions 13

2.3 Parameterized Complexity 14

I Voting 17

3 Winner determination for Chamberlin-Courant on restricted domains 19

3.1 Introduction [1] 19

3.1.1 Our Contributions and Organisation of the Chapter 20

3.2 Preliminaries 21
3.3 Tractability on Nearly Structured Preferences 23
3.3.1 Overall Approach 23
3.3.2 Problem Setup 23
3.3.3 (`, D)-CC Extension for the Single-Crossing Domain 25
3.3.4 (`, D)-CC Extension for the Single-Peaked Domain 27
3.4 Hardness for Generalized Restrictions on the Domain 28
3.4.1 3-Crossing Profiles 28
3.4.2 3-composite single-peaked domains. 34
3.5 Conclusion and Open Problems 37

4 Robustness Radius for Chamberlin-Courant on Restricted Domains 39

4.1 Introduction [2] 39
4.1.1 Our Contributions and Organisation of the Chapter 40
4.2 Preliminaries 41
4.3 XP algorithm for Rankings and Approval-CC 42
4.4 W[2]-hardness for Approval-CC 46
4.5 Robustness for nearly restricted domains 47
4.5.1 Domains Close to Single-Crossing Domain 47
4.5.2 Domains Close to Single-Peaked Domains 53
4.6 Conclusion and Open Problems 54

II Matching 57

5 Stable Matching on Restricted Domains 59

5.1 Introduction 59
5.1.1 Our contributions and Organization of the Chapter 61
5.2 Preliminaries and Background in Matching 63
5.2.1 Problem Setup 63
5.2.2 Preferences with ties 64
5.2.3 Variants of Stable Matching Problems 65
5.2.4 Background and Gale Shapley Algorithm 65
5.3 Stable Roommates with Short Lists and Few Ties 66

5.4 Sex-Equal Stable Matchings 69
5.5 Egalitarian Stable Matchings 71
5.6 Domain-Restricted Manipulation 74
5.7 Domains with Unique Stable Matching 76
5.7.1 1-D Euclidean Preferences 76
5.7.2 Globally Ranked Pairs 77
5.7.3 Nearby Domains with large number of stable matchings 78
5.8 Matching Critical Set 79
5.9 Special cases of (a,b)-Supermatch 81
5.9.1 Hardness result for nearby problem 85
5.10 Experimental Results on Restricted Domains 88
5.11 Concluding Remarks and Open Problems 89

6 Extension Problems in Stable Matching 91

6.1 Introduction 91
6.2 Preliminaries 91
6.3 Extension to Unique Matching Profile 92
6.3.1 Extension to general preferences 92
6.3.2 Extension to a special preference profile 93
6.3.3 Extension to 1D-Euclidean Profiles 94

III Fair Division 97

7 Connected Fair Division on a Path with Envyfreeness 99

7.1 Introduction 99
7.1.1 Summary of Results 100
7.2 Preliminaries 100
7.3 Results for Efficiency Notions 102
7.4 Existence of EF1 allocations 104
7.4.1 EF1 and Completeness 104
7.4.2 EF1 and Pareto Optimality 105
7.4.3 EF1 and Non-Wastefulness 106
7.5 Hardness Result for Envyfreeness 106

7.5.1 EF1 and Non-wasteful Allocation 106
7.5.2 EF1 and complete allocation on a collection of Paths 107
7.6 Efficient Algorithms for Restricted Preference Domains 109
7.6.1 An Algorithm for Binary Left-Extremal Valuations 109

8 Connected Fair Division on a Path with Equitability 113
8.1 Preliminaries and Known Results 113
8.1.1 Known Results for Equitability 113
8.1.2 Our Contributions and Organization of the Chapter 114
8.2 Existence of EQ1 allocations 115
8.2.1 EQ1 and Pareto Optimality 115
8.2.2 A few remarks 116
8.3 Efficient Algorithms for Equitable Allocations 117
8.3.1 Algorithm for complete-EQ1 Allocations 117
8.4 Hardness Results 119
8.4.1 Existence of EQ1 and Non-Wasteful 119
8.4.2 Complexity of finding EQ1+complete allocation with maximum utility 123
8.4.3 Existence of Pareto Optimal Allocation 125
8.5 Efficient Algorithms for Restricted Preference Domains 132
8.5.1 EQ1+NW allocations 132
8.5.2 EQ1+PO allocations 133
8.6 Conclusion and Open Problems 134

Bibliography 135

List of Figures

2.1 Single-Peaked Domain 12

2.2 Single-Crossing Domain 13

3.1 Chamberlin Courant Winner Determination 20

3.2 Example of construction of election instance from LSAT 30

5.1 Preferences for egalitarian matching instance 72

5.2 Sub-optimal Manipulation when restricted to inconspicuous manipulation 74

5.3 Sub-optimal Manipulation with domain restriction 75

5.4 1D-Euclidean Preferences 76

5.5 Instance with 2 types of preferences on one side 78

5.6 1D-Euclidean Preferences with ties 79

5.7 Comparison of proposals in Gale-Shapley for general vs restricted domain (SP) 88

7.1 The path G constructed in the proof of Theorem 23. 103

7.2 Sequence of goods on path for Theorem 25 106

8.1 Non-existence of EQ1+ PO allocation even for interval valuations 115

8.2 Non-existence of EQ1+ PO allocation even for *k-interval* valuations . . . 116

8.3 Non-existence of EQ1+ PO allocation even for extremal valuations . . . 116

8.4 Phase I: Critical utility 119

8.5 The instance used in the proof of part (a) of Theorem 34. The path graph is such that the goods in the top row are to the left of those in the middle row, which are to the left of those in the bottom row. 126

8.6 The instance used in proof of part (b) of Theorem 34. 130

List of Algorithms

1 Algorithm for verifying $(n, 0)$-supermatch 83

2 Algorithm for binary left-extremal valuations 110

Chapter 1

Introduction

1.1 Social Choice

Social Choice deals with the aspects of collective decision making for a variety of natural issues in our society. A canonical example of collective decision making is *elections*. For millennia, right from the beginning of the Roman Empire, the way of selecting local authorities by accounting opinions/ preferences of concerned people through elections (commonly known as Elections in the Roman Republic) has been prevalent. Apart from elections, consider the issues such as finding a *match* between two entities for performing a specific task given several participating entities in the market and their willingness to get paired with others or finding a reasonable way to divide the leftover pie among the siblings. We refer the reader to (BCE+16, Chapter 1) for a detailed account of history and developments in social choice.

While designing *procedures* for issues such as *voting, matching* or *fair division*, we expect any *rational* procedure to have some sensible properties, and to avoid some undesired traits. For example, consider the election system in some countries in Africa and South-East Asia. It might not be a popular vote around the world that using these systems is a good practice. In particular, it is fair to assume that an election procedure should not be dictatorial. Similarly, giving out complete leftover pie or not dividing it at all (an empty allocation) are valid ways of allocations unless the aim is to find *fair* or *e cient* division of the cake. Hence, *evaluating* social choice procedures against some standard *axioms* or demanding them to have certain *properties* is typical.

Procedures to solve social choice problems are commonly known as mechanisms. When the market size is large, the time for computing outcome of the decision mechanisms becomes vital. In such cases, it is desirable to find efficient mechanisms for social choice problems at hand for mechanisms to be pragmatic. Often, the mechanisms which satisfy appealing properties turns out to be computationally inefficient. The field of Computational Social Choice (COMSOC) deals with the computational complexity of mechanisms. Here, the quest is for finding efficient mechanisms whenever exists, or showing limitations of mechanisms by proving appropriate complexity lower bounds.

In this investigation, we consider three pillars of social choice namely *Voting* (Sen87), *Matching* (GI89), and *Fair Division* (Tay04). We start with an introduction to each of these and try to sketch the landscape of related results to place our work in the context.

1.1.1 Voting

Two primary reasons for employing elections in various scenarios are based on need and necessity. Consider for instance an election for electing a president of the country, in this case; it is the *need* of the nation to choose *one person* to represent and handle the course of actions for certain period to maintain smooth functioning. Consider another situation where a committee of field experts has to *choose* a certain number of applicants to award a fellowship due to budget constraints. In this case, it is necessary for committee members to aggregate the opinions, and select a limited number of deserving applicants.

In a common election setting, there is a set of *candidates* competing for a position(s)/ award, and there is a set of *voters* who express there *preferences* (opinions) over the candidates. Voting procedures aim to answer the following question – "how individual preferences can be aggregated to give a social choice or election outcome that reflects the interests of the electorate?" (Bra08). In order to answer this question, the agency conducting an election has to *elicit* the *preferences* of the voters. There are several ways to collect these preferences; the extreme two cases would be asking for *complete strict ordering* (pairwise comparisons between all pairs of candidates) while other would be to only ask for the top preference of each voter. It is rational to expect certain fundamental *properties* from any "reasonable" voting rule. Such properties are known as *axioms* in the voting literature. Arrow performed the first meticulous axiomatic study (Arr51) and demonstrated an *impossibility theorem* which states that there does not exist any reasonable preference aggregation scheme which simultaneously satisfies *all* fairness criteria at once. Arrow developed a mathematical model to evaluate all voting rules. It is possible to bypass the *impossibility theorem* by relaxing the set of properties satisfied by the rule. Another way out is to consider voting rules on *restricted domains* where it is possible to obtain voting rules which simultaneously satisfies *all* fairness criterion.

In last fifty years, several *voting rules* (preference aggregation schemes) each satisfying specific set of properties are proposed and are well studied (BF02), (BCE+16). As described earlier for the ways to elicit the preferences, generally the voting under consideration drives the elicitation as some rules only require the information about the top preferences (*plurality*) whilst others need complete rankings (Borda voting rule) (Bor81). The problem with the plurality is the ignorance for voter's preferences after the top preference, to tackle this a well-studied class of voting rules is known as *positional scoring rules*. Here, the voter's submit complete rankings, and each candidate receives a fixed score if it appears at a certain position in a vote. An issue with such rules is the high *elicitation* cost since it is conceivable that obtaining such exhaustive information is not cheap. This motivates one direction of problems in voting, namely *preference elicitation* where the aim is to minimize the amount of elicitation required to compute an outcome of the election. Another natural direction of problems deals with the complexity of *preference aggregation or winner determination*. Here, the task is to compute an outcome of the election in terms of the winner. At last, an important set of voting problems are the *control* problems which deals with issues such as manipulation and bribery in voting. We refer the reader to (BCE+16) for known results and open problems in these directions.

Recall the example of shortlisting candidates for a particular fellowship. Here, the objective is to find a group of a certain number of most deserving candidates rather than one unique winner. Similarly, consider the problem of shortlisting a small list of movies to include on an airplane such that passengers can attain maximum satisfaction. Both of these scenarios and many such others aims to select *commi ee* rather than a single winner, this motivates the need for *Multiwinner rules*. One way to design multiwinner voting mechanisms is to naturally generalize single-winner rules to their multiwinner versions e.g. Plurality, t-Approval, Borda, STV, SNTV, etc. whilst the other approach would be to design completely new rules such as *Chamberlin-Courant* (CC83) and Monroe (Mon95). Properties of *Multiwinner rules* are discussed thoroughly in (EFSS17). The paper (EFSS17) also provides one way of classifying these rules based on the way of committee selection.

Another important trait recognized for Multiwinner rules is *proportional representation*. We go back to the example of Movie shortlisting, the aim here is to have a diverse set of movies which would be appealing for passengers with *every shade of preferences* rather than selecting a collection of overall "best" set of movies (say movies with highest IMDB ratings). Note that this need not be the case of with fellowship shortlisting where one can choose best applicants. Black expressed proportional representation as:

"A scheme of proportional representation attempts to secure an assembly whose membership will, so far as possible, be proportionate to the volume of the different shades of political opinion held throughout the country; the microcosm is to be a true reflexion of the macrocosm" ((Bla58), (BSU13)).

The *Chamberlin-Courant rule* (CC83) and *Monroe rule* (Mon95) are well studied proportional representation rules. In our work, we focus on *Chamberlin-Courant (CC)* rule which comes under *positional scoring rules* (EFSS17). In *CC-rule*, each voter is represented by a top candidate amongst the committee members according to her preferences. For both *CC* ((PRZ08), (LB11)) and *Monroe* the *Winner Determination* problem is NP-hard. We present our results on certain problems on special preference domains with *CC-rule* in Chapter 3 and Chapter 4.

1.1.2 Matching

Traditionally, a stable marriage problem (a two-sided matching problem) is defined by a set of men and women; each of which submits their preference order over agents from opposite sex to a centralized administrator. The task here is to find pairs such that no man-woman pair should be inclined to elope. Set of pairs with this property is known as *stable matching*. A positive answer to the fundamental question of the existence of such matchings was presented in the seminal Gale-Shapley paper (GS62) through an efficient *deferred acceptance* algorithm. The algorithm has several appealing combinatorial properties (Rot08). The problem was generalized to many-one matchings in two-sided markets with important applications in *hospital-resident program* (Rot84) in US, and *school admissions* in Singapore (TST01). Apart from these several other generalizations were introduced (IM08) including the problem of finding Stable Matchings in non-bipartite setting, which is commonly known as *Stable Roommates* problem.

The guarantees of existence and efficient computation from *Deferred Acceptance* algorithm holds only when agents submit complete preference orders i.e. the pairwise preferences for all pairs of agents (note that for such instances, even the *Stable Roommates* problem can be solved efficiently (Irv85)). As the market size increases, it is not feasible for all agents to submit preferences over all other agents since a large fraction of those might be completely irrelevant for it. Also, in some case, two options may be equally liked/disliked by an agent. In such cases, agent should be allowed to submit *incomplete* preferences (i.e. declaring some agents as *unacceptable*), and preferences with *indi erence/ties*. For the stable matching instances with incomplete preferences and ties, the existence of stable matching is not always guaranteed, and the complexity finding these matchings changes sharply making problems a lot harder. In Chapter 5, we consider several optimization problems (ILG87) for such stable instances with special preferences (MS10), (BFK+17).

Another line of research in matching focuses on the ubiquitous phenomenon of the strategic behavior of agents. As a complement of Gibbard-Satterthwaite theorem in voting, Roth (Rot82) showed that there does exist any stable matching mechanism for which stating *true preferences* is a *dominant* strategy for *all* agents. There are two popular models of *manipulation* – *truncation* and *permutation*. Of the two, *truncation* is shown to be a more powerful model in terms of *gain* (RR99; CS14) through manipulation.

Recall a stable matching instance with ties and incomplete preferences. For such instances, there might exist several ways of transforming these instance into an instance with complete preferences for all agents (note that this can be achieved by breaking ties in different ways and by trying all possible order of completions for incomplete preferences). Hence, given a stable matching instance with ties or incomplete preferences, one can ask questions such as, does there an extension of the given instance which admits unique stable matching? Or an extension which admits a stable matching which matches certain agents? In the recent paper by Menon and Larson (ML18), they consider another natural question of finding a fixed matching which admits a minimum number of blocking pairs over all possible extensions to complete preference orders. In section 5.8 and Chapter 6 we provide preliminary results on some of these questions.

We refer the reader to the Introduction Section 5.1 from Chapter 5 for thorough details on known results for these problems.

1.1.3 Fair Division

Consider a classic example of a division of birthday cake (say) at Alice's birthday party. Amongst her friends, Bob, Charlie and Daniel decide to stay back after party to help Alice finish-up the delicious chocolate truffle cake. All four agents seek for different parts of cake: Alice wants a part with a piece of chocolate, Bob likes the part with Cherry, Charlie wants a piece on the border, and Daniel likes all parts equally. A task here is to divide the cake among these four agents by prioritizing their choices in such a way that each agent prefers its share to anyone else's (such allocations are known as *envyfree* allocations). Another example is *fair* allocation is distribution courses among students according to their preferences. We point out an intrinsic difference between the two examples: cake is a *divisible* resource whilst the courses is an *indivisible* resource. We refer the reader to following surveys (BT96), (Bar05), (Tho11) for thorough introduction of model and several other applications.

For the case of *divisible* resources, Brams and Taylor (BT00) showed an existence of *Envyfree (EF)* allocations. When we move to *indivisible* resources, the result does not hold; consider the case of two agents and one good with both agents wish to have the good. Hence, for indivisible goods Budish (Bud11) introduced a relax the condition of *EF* to *EF1* which is *envyfreeness up to one good*. This means that every agent prefers its allocation to an allocation obtained by removing *some good* from the allocation of other envious agents. *EF1* allocations always exist, and can be obtained by the famous envy-graph algorithm by Lipton et al. (LMMS04) or by a simple *Round-Robin procedure* (CKM⁺16).

Another fairness notion considered in literature is *Equitability (EQ)*. An equitable allocation implies the same happiness (utility) for each agent according its valuation at the end of the protocol. For two agents, (BT00) gave an *adjusted winner procedure* which returns Envyfree and Equitable allocation. The existence of *Equitable* allocation for divisible goods was given by (Alo87). Again, with similar reasoning for the case of indivisible goods, *EQ* allocations are not guaranteed to exist. We analogously define *EQ1* allocations. Existence of such allocations was shown by (Chè17). In Chapter 8 we show existence of *EQ1* allocations in Theorem 31, and give a simple procedure to compute such allocations.

In Chapter 7 we consider connected *EF1* allocation on a path for *indivisible* goods. Next, in Chapter 8, we consider *EQ1* allocations for same setting.

1.2 Restricted Domain in Social Choice - Background and motivation

In the previous section 1.1, we alluded to the *special/restricted* domains several times. In practice, it is atypical to have completely arbitrary preferences of participating agents. More often, these preferences are driven by an underlying philosophy. For example, consider the case of political elections from West. Usually, there are two major parties, with one having the left-wing ideology and the other with the right-wing ideology. Consider an axis with the leftmost side being extreme left-wing candidate, the rightmost side being extreme right-wing candidate, and in with neutral candidates around the center. In general, it is observed that each voter's ideology lie at some point on this axis, and hence, she prefers the candidate close to that point the most ("peak"). For the rest of the candidates the preference decreases as we move away from the "peak" from both sides. Such preferences are known as 'single-peaked' preferences (Bla48). Ballester et al. (BH11) gave a nice *characterization* of these domains.

Another example domain restriction is 'single-crossing' domains. Here, instead of candidates, we order voters such that for each pair of candidates there exist a single point at which voters switch from preferring one over the other. Single-crossing domain was defined by Roberts (Rob77) in his seminal paper of re-distributive income taxation. The characterization for single-crossing domains, and reference to several applications is given in (BCW13). In 1D-Euclidean domain, we arrange both candidates and voters on the axis, and the preferences of voters are derived from their euclidean distance from respective candidates (Coo50). Consider a scenario of the division of customers among the outlets of McDonald's franchise in a city. Given all amenities same across all outlets, the customers will prefer the one closest to their homes. For this case, we can place both the outlets (candidates) and customers (voters) on a line according to their locations, and derive the preferences of customers over the outlets.

In India, the admissions to the Indian Institute of Technology (IITs) is made using the ranks of the students in Joint Entrance Examination (JEE). Several popular ranking agencies have consistently ranked IIT Bombay, IIT Delhi, and IIT Kharagpur as the top three institutes in the country for the past few years. Consider the preferences of top hundred students in JEE. A large fraction of these students will place these three in institutes as their top choices, and it is unlikely to find one of these three institutes at the bottom of someone's preference list. In this case, the difference between the highest and lowest rank for these three institutes in the preferences of students is *bounded*. Such *candidate range bounded* preferences occur in many other scenarios in the market. Betzler et al. (BU09) motivated the study of many such preferences and gave efficient algorithms with these range parameters for *Kemeny elections*.

A prime motivation to study restricted domains is that these domains admit several desirable properties that are elusive on the general domains. One such property is efficient algorithms in certain cases for which the problem is *hard* on general domains. A few examples of this are: Single crossing domains are non-manipulable (Mou80) and admit transitivity of majority rule (Ina69). For the single-peaked and single-crossing domains, problem of finding *Chamberlin-Courant winner* is tractable (BSU13), (SYFE15). In stable matching, both 1D-Euclidean domains and narcissistic single-peaked domains admit unique stable matching which gives non-manipulability, and these the there exist a sublinear time mechanism to find a stable matching (BIT86).

There have been many attempts to generalize the described domains. Demange (Dem12) generalized single-crossing to intermediate property on median graph. The paper by Barberà and Moreno (BM11) generalized both single-peaked and single-crossing property to obtain *top-monotonicity* which retains the existence of Condorcet winner. In (CGS12) authors define single-peakedness for a bounded sized cluster of candidates known as "single-peaked width" (we note that analogous notion is defined for single-crossing profiles). Erdélyi et al. (ELP13) study computational aspects of nearly single-peaked electorates. Peters (Pet17) consider recognition of *d-dimensional* Euclidean domains.

In this thesis, for Chapter 3 and Chapter 4 we explore certain problems with Chamberlin-Courant rule on *nearly restricted domains* by using known and newly defined notions of generalized domains. In Chapter 5 and Chapter 6 we investigate *hard* matching problems on known restricted domains.

1.3 Contributions and structure of the thesis

In this section, we provide a brief overview of our contributions in *voting*, *matching*, and *fair division*.

1.3.1 Part I - Voting

In *voting* we fix the voting rule to *Chamberlin-Courant* (CC83) for our work. Given an election instance the *Winner Determination* problem was known to be polynomially solvable on *single-peaked* (BSU13) and *single-crossing* (SYFE15) domains.

In Chapter 3, we focus on the Winner Determination problem for the Chamberlin-Courant rule on "nearly" restricted domains. We consider two natural generalizations of restricted domains. In our first generalization, we consider the profiles which are k - voters / k - candidates away from single-peaked or single-crossing domains i.e. by deleting at most k - voters/candidates one can obtain the profile which belongs to the restricted domains. Finding this 'deletion' set is known to be efficient (EL14). With this generalization, we show FPT algorithms for the case of candidate deletion and XP algorithms for voter deletion parameterized by the size of the deletion set for both single-peaked and single-crossing domains.

In another attempt at generalizing the restricted domains, we define r-composite single-peaked (definition 2) and r-crossing profiles (definition 4). For 3-composite single-peaked domains and 3-crossing domains we show that it is NP-hard to compute the *CC-Winner*. For the detailed account of results we refer the reader to Section 3.1.

Our work in Chapter 4 is based on the concept of *Robustness Radius* for a voting rule. Recently, (BFK+17) defined the concept of *Robustness Radius*. The problem of computing the robustness radius for *Chamberlin-Courant* rule was shown to be W[1]-hard with respect to the size of the committee. We complement this result by providing an XP algorithm with the same parameter. Next, we extend the result for approval setting to show W[1]-hard, and complement with the XP algorithm with respect to the size of the committee. Finally, we turn to the generalized restricted domains (note that for restricted domains, the problem admits a trivial XP algorithm with parameter robustness radius). We show hardness results for generalizations of both single-peaked and single-crossing domains. We refer the reader to Section 4.1 for further details.

Acknowledgements. Chapter 3 is based on joint work with Neeldhara Misra and P. R. Vaidyanathan and has appeared in *ADT 2017* (MSV17). The work in Chapter 4 is largely based on joint work with Neeldhara Misra, and has appeared in *SOFSEM 2019* (MS19).

1.3.2 Part II - Matching

Our main focus of Chapter 5 is to investigate popular problems in *matching* on restricted domains. We consider several *hard* matching problems, and the issue of strategic behaviour (manipulation) on commonly known restricted domains in stable matching literature, and other well studied domains in the computational social choice. Next, we provide a short summary of our results in matching, we refer the reader to Subsection 5.1.1.

Bredereck et al. (BCFN19) posed a question of finding a stable matching for Stable Roommates with Ties and Incomplete preference parameterized by the "degree of incompleteness". We show the problem to be W[1]-hard even on single-peaked and single-crossing domains. Next, for Sex-Equal stable matching and Egalitarian stable matching, we show that the hardness persists even on the restricted domains.

For the problem of manipulation, we show an interesting contrast to the results in (VG17) which states that the optimal manipulation can be obtained 'inconspicuously' (by making minimal changes to the original preference order), by showing limitations of manipulation when preferences are limited to *restricted* domains. We follow this with a discussion on preference profiles with unique stable marriages. We show that *1D-Euclidean* preferences admit unique stable matching. We complement this by showing an exponential number of stable matchings for "almost" restricted domains.

Next, we define a problem of finding a stable matching for a *'critical set'*. Given a critical set of agents, problem asks for a stable matching which matches *all* agents from this set. We provide further motivation in Section 5.1 and details on results in Section 5.8. We conclude the Chapter 5 with an elegant algorithm for finding *(n,0)-supermatch* which is a special case of *(a,b)-supermatch* defined in (GSOS17).

In Chapter 6, we take a detour to consider the problems related to the *extension* of stable matching instances with ties and incomplete preferences. We focus on extensions which admits a unique stable matching. For the general case, we show hardness for finding the existence of such *extension*. For a special case when the agents are matched to their top preference in the extended profile, we show an efficient algorithm to solve such instance, and we conclude the Chapter with showing hardness for finding extension to *1D-Euclidean* domain.

Acknowledgements. The work from both Chapter 5 and Chapter 6 is done in collaboration with Neeldhara Misra.

1.3.3 Part III - Fair Division

In this work, we focus on *connected fair division* of indivisible goods. We consider the case when goods are arranged on the path for the case of binary valuations. Notice that in this case, a connected allocation essentially means an interval of goods on the path. Our results can be primarily classified in two classes based on two fairness notions – Envyfreeness and Equitability.

In Chapter 7, we study the notion of envyfreeness. Along with the general binary valuations on the path, we consider restricted valuation functions as a special case. For the efficiency, we study *Pareto Optimality* and *Non-wastefulnsess* (each agent should only be allocated the goods she approves) along with envyfreeness. We start with the investigation on the existence of these allocations and later move to the computation. We note that the existence and computation of EF1+PO allocations for general allocation have been resolved in (IP19). We resolve the question of existence for all the cases and find the complexity of almost all the cases except when valuations form a contiguous *interval*.

Next, we consider the notion of Equitability for our work in Chapter 8. We consider the same set of restricted domains and efficiency notions, as described above. Unlike envyfreeness, to the best of our knowledge, we are first to consider equitability for connected allocations of discrete goods on a path. We resolve all the questions regarding the existence of such allocations. On the computation part, we provide a general polytime algorithm computing EQ1+complete allocations for any setting. We also provide several elegant efficient algorithms for finding EQ1 allocations for restricted domains.

Except for the case of arbitrary interval valuations, we provide hardness results for all the remaining cases.

Acknowledgements. Both Chapter 7 and Chapter 8 are based on joint work with Rohit Vaish, Neeldhara Misra and P. R. Vaidyanathan.

Chapter 2

Preliminaries

In this chapter, we provide the definitions to the concepts common across all the three following parts of the thesis. We will define the types of preferences followed by the domain restriction we consider in our study. We end the chapter with the terminology in classical and parameterized complexity theory.

For a given natural number $k \in \mathbb{N}$, we denote the set $\{1, 2, \ldots, k\}$ by $[k]$. A generic instance with preferences consists of a two-tuple $\langle V, P \rangle$ (the tuple might contain extra information according to the problem at hand). Here, $V = \{1, 2, \ldots, n\}$ is the set of participating *agents*. A binary preference relation $\succ$ indicates the preferential distinction between two agents. The relation $\succ$ is asymmetric and transitive. For agents a, b, i; $a \succ_i b$ denotes that agent i strictly prefers agent a over b. Similarly, the binary transitive relation $\sim$ indicates the indifference relation i.e., $a \sim_i b$ denotes indifference between agents a and b for agent i. The preferences are said to be complete-strict orderings if each agent provides a strict ranking (i.e. preferences without indifference) over all other agents. We define P to be the preference profile, which is a collection of preferences of all participating agents in the instance.

Another type of preferences we consider in this work are approval ballots. In this case, each agent $v \in V$, approves a subset $A_v \subseteq V \setminus v$ and dis-approves all other agents in the instance. In general, these preferences are represented as the binary vectors of appropriate dimension (depending upon the number of agents in the instance).

2.1 Restricted Domain under consideration

In this section, we provide the definitions of the restricted domains we consider. We refer the reader to Section 1.2. Let $E := (C, V)$ be an election with set of candidates C and set of voters V.

Definition 1. *Single-Peaked Preferences: A preference order $\succ$ over V is called* single-peaked *with respect to order $\succ'$ of V if, for every pair of agents $x, y \in V$, we have $x \succ y$ whenever we have either $c \succ' x \succ' y$ or $y \succ' x \succ' c$, where $c \in V$ is an agent at the first position of $\succ$. A profile P is called single-peaked with respect to $\succ'$ if $\forall i, \succ_i \in P$ is single-peaked with respect to $\succ'$.*

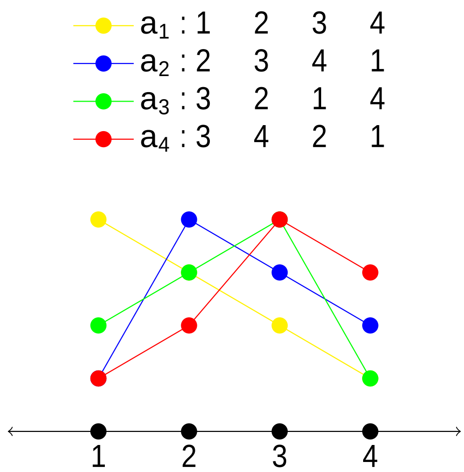


Figure 2.1: Single-Peaked Domain

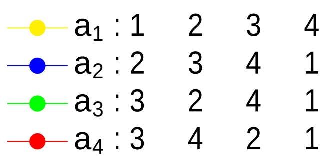
We generalize definition of a k-composite single-peaked profile, which is a natural generalization of the single-peaked notion above.

Definition 2. *k-composite single-peaked: A profile is k-composite single-peaked if there is an ordering of the candidates $\succ$ and a partition of the candidate set into at most k parts such that each part induces a single-peaked profile on $\succ$ restricted to that part.*

We note, importantly, that this is different from the more well-studied notion of multi-peaked profiles, where we have the additional constraint that the k parts have to additionally form intervals on a fixed global ordering.

A similar notion called k-additional axis where the votes(rather than the candidates) are divided into k buckets and each bucket is single-peaked, has been studied in (ELP13).

Definition 3. *Single-Crossing Preferences: A profile P of ℓ preferences over set of agents V is called a* single-crossing *profile if there exists a permutation σ of $[\ell]$ such that, for every pair of distinct agents $x, y \in V$, whenever we have $x \succ_{\sigma(i)} y$ and $x \succ_{\sigma(j)} y$ for two integers i, j with $1 \leq \sigma(i) \leq \sigma(j) \leq n$, we have $x \succ_{\sigma(k)} y$ for every $\sigma(i) \leq k \leq \sigma(j)$.*



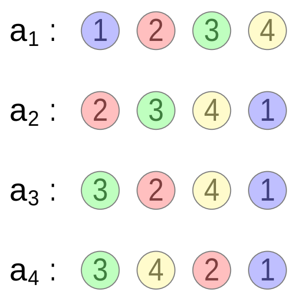


Figure 2.2: Single-Crossing Profile (Any pair of colors crosses at most once)

We now define the notion of single-crossing domains to r-single-crossing domains in the following natural way (c.f. (MSV17)): for every pair of candidates (a, b), instead of demanding one index where the preferences "switch" from one way to the other, we allow for r such switches.

Definition 4. *r-single-crossing: a profile is r-single crossing if for every pair of candidates a and b, there exist r indices $j_0[(a,b)], j_1[(a,b)], \dots j_r[(a,b)], j_{r+1}[(a,b)]$ with $j_0[(a,b)] = 1$ and $j_{r+1}[(a,b)] = n+1$, such that for all $1 \leq i \leq r+1$, all voters v_j with $j_i[(a,b)] \leq j < j_{i+1}[(a,b)]$ are unanimous in their preferences over a and b.*

Definition 5. *1D Euclidean Preferences: A profile P is said to be 1-D Euclidean, if there exist a mapping $X : V \to \mathbb{R}$ which maps every agent on a real line such that for any agent i, $x \succ_i y$ if and only if $|d(i,x)| < |d(i,y)|$.*

Hence, an agent prefers the agent closer to it to the farther agent.

Definition 6. *Narcissistic profile: We say that the given profile is narcissistic if every agent appears in the top position (i.e. most preferred) in the preference order of at least one agent.*

Apart from these we consider some special preference classes for dichotomous preferences. These classes are inspired from the work of Elkind et al. (EL15). Some of these are variants of single-peaked and single-crossing properties for dichotomous preferences. Primarily, we use these classes in our work in Connected Fair Division part III, hence, we defer the definitions to Section 7.2.

2.2 Classical Complexity and reductions

The complexity of a computational problem measured by the resources required to solve the problem. According to these resources, there are several *classes* of computational complexity. In the context of this thesis, we consider *time* as the primary resource for our classification of problems (the other resource of prime interest is the space required to solve the problem). For the detailed introduction to the theory of computation complexity, we refer the reader to (Sip) and (AB09).

The computational problems are often defined as a set of instances (strings) L with certain properties, and the problems are posed as a decision version. The solution to such a decision problem is *YES* if the given instance belongs to the set and is *NO* if the instance does not belong to the set. A *time complexity* for the problem is defined as the number of computational steps required to solve the problem in the worst case as a function of the size of the input instance.

Two well-studied complexity classes (with respect to time complexity) are P and NP. A problem is said to be in P if there exists an algorithm which decides the problem in *polynomial time*. Such problems are said to be efficiently solvable. NP is a class of problems which admits efficiently verifiable certificates in the positive case. In other words, there exist a polynomially verifiable "witness" for the *YES* instances of the problem.

Definition 7. *A decision problem* L *is said to be* NP*-complete if:*

1. $L \in \mathsf{NP}$
2. L *is* NP*-hard*

The first condition implies the existence of efficient certificate for L. The second condition implies that L is at least as hard as any other problem in NP i.e. there exist a many one reduction from any other $L' \in \mathsf{NP}$ to L. We now give the definition of the many-one reductions as suggested by Karp (Kar72).

Definition 8. *Given two* L *and* L'*, if there exist a polynomial function* g*, such that*

$$x \in L \iff g(x) \in L'$$

then we say that there exist a polynomial time many-one reduction from L *to* L'*, and we denote it as* $L \leq_p^m L'$*.*

To show a particular problem to be NP-complete, one has to show a an efficient certificate for the problem, and show a polynomial time reduction from known NP-complete problem to the problem at hand (this will establish condition 2 for NP-completeness).

An efficient (polynomial time) algorithm for NP-complete problem seems unlikely since by the definition that would imply efficient algorithm to any other problem in NP. Throughout this thesis we will assume that $\mathsf{P} \neq \mathsf{NP}$.

2.3 Parameterized Complexity

A parameterized problem is denoted by a pair (Q, k) N. The first component Q is a classical language, and the number k is called the parameter.

A central concept in parameterized complexity is the notion of *xed parameter tractability* (FPT). We call a problem (FPT) if there exists an algorithm that decides it in time $O(f(k)n^{O(1)})$ on instances of size n. Notice that the first component in the running time can be *any* function of *only* the parameter under consideration.

On the other hand, a problem is said to belong to the class XP if there exists an algorithm that decides it in time $n^{O(f(k))}$ on instances of size n.

For further fine grained analysis, the problems outside FPT are divided into the complexity classes known as *W-hierarchy*. The relation between FPT, *W-hierarchy*, and XP is as follows:

$$\mathsf{FPT} \quad \mathsf{W[1]} \quad \mathsf{W[2]} \qquad\qquad [\mathsf{W}\text{-SAT}] \quad \mathsf{W[P]} \quad \mathsf{XP}$$

We refer the reader to (CFK+15) for a comprehensive introduction to parameterized algorithms.

Part I

Voting

Chapter 3

Winner determination for Chamberlin-Courant on restricted domains

3.1 Introduction [1]

We refer the reader to subsection 1.1.1 for a gentle introduction to voting. A traditional election setting consists of voters expressing their preferences over alternatives, where preferences can be modelled in several ways (approval ballots, ternary ballots, top-truncated lists, total orders, and so forth). Usually, given such a scenario, we would like to identify a winning alternative. In many applications, however, we need to identify not one, but a fixed *set of alternatives* that best represent the interests of the voters. Such a problem arises in a variety of scenarios like committee selection, parliamentary elections, movie recommendation systems, and so forth.

There are several ways of measuring how well a committee fares against a set of votes. When votes are approval ballots, for instance, the maximum or the sum of Hamming distances is often used as a measure of quality. We consider the setting of votes given as complete rankings, and focus on the well-studied Chamberlin-Courant rule (CC83), which achieves proportional representation. The way this voting rule works is the following. We begin by fixing a notion of a "dissatisfaction function" $\alpha : \mathbb{N} \rightarrow \mathbb{N}$, which simply specifies, by $\alpha(i)$, how unhappy a voter is when she is represented by a candidate who is ranked at the i^{th} position on her list. Given a committee with k candidates, a voter is represented by the candidate that she ranks the highest among candidates from X. If $\Phi(v)$ denotes the candidate that is representing voter v, the optimal committee under the Chamberlin-Courant voting rule seeks to minimize either the sum or the maximum value of $\alpha(\text{pos}_v(\Phi(v)))$, taken over all voters v (where $\text{pos}_v(c)$ denotes the ranking of the candidate c in the vote v).

The Chamberlin-Courant rule (and the closely related Monroe voting rule, which we do not consider in the present work) has several desirable properties. It has been argued (SYFE15) that rules that achieve proportional representation are particularly well-suited for electing committees that need to make unanimous decisions, and in particular,

[1]Some portions of this chapter are taken verbatim from (MSV17)

that takes minority candidates into account. However, it turns out that finding an optimal committee under this rule is NP-hard (PRZ08), (LB11), and it is therefore unlikely to admit an efficient algorithm.

On the other hand, there have been promising developments showing that the optimal Chamberlin-Courant committees can be computed efficiently on structured profiles which are commonly encountered in practical scenarios. Two such restrictions that have been particularly successful are the single-peaked (BSU13) and single-crossing domains (SYFE15). In a parallel development, (EL14) showed various efficient algorithms for detecting profiles that are close to being structured (in this case, the notion of closeness is that these profiles exhibit the structure of the domain on all but a small number of candidates or voters). More generally, the notion of closeness to a domain is well-studied and has been defined in various ways (FHH14).

We combine these scenarios to address the following question: how well do the efficient algorithms on the restricted domains extend to profiles that are of the latter type, that is, they exhibit the properties of the domain on all but a small number of candidates or voters? We now turn to our findings in the context of this question and closely related issues.

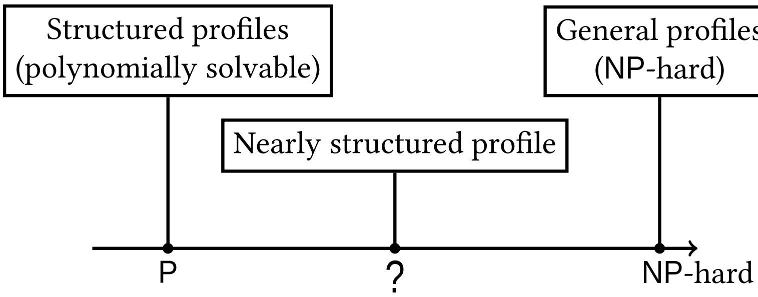


Figure 3.1: Chamberlin Courant Winner Determination

3.1.1 Our Contributions and Organisation of the Chapter

A natural framework for addressing the problem of how well algorithms on structured domains scale up to nearly-structured ones is parameterized complexity (CFK+15). To begin with, we show efficient algorithms on profiles that k candidates or k voters away from the single-peaked and single-crossing domains. In particular, for profiles that are k candidates away from being single-peaked or single-crossing, we show algorithms whose running time is FPT in k. For profiles that are k voters away from being single-peaked or single-crossing, our algorithms are XP in k. These algorithms are obtained by a careful extension of the known algorithms (BSU13; SYFE15) on the structured profiles. This provides a natural application for the work by Elkind and Lackner in (EL14), who study the problem of finding deletion sets to single-peaked and single-crossing profiles.

In contrast to these results, for a different, but equally natural way of generalizing these domain, we show severe intractability results. In particular, we show that the problem is NP-hard on profiles that can be "decomposed" into a constant number of single-peaked profiles. Also, if the number of crossings per pair of candidates in a profile is permitted to be at most three (instead of one), the problem continues to be NP-hard. This stands in contrast with other attempts at generalizing these domains (such as single-peaked or single-crossing width (SYFE15; CGS12)), as it rules out the possibility of fixed-parameter (or even XP) algorithms when parameterized by the number of peaks, or the maximum number of crossings per candidate pair. We show the hardness for both Harsanyi ((Har75) ℓ_1) and Rawlsian (ℓ_∞) misrepresentation aggregation functions.

	SP CC	SP MM[2] CC	SC CC	SC MM CC
Struct[3]	$O(nm^2)$	$O(nm)$	$O(n^2mk)$	$O(n^2mk)$
VDel	$2^{Rk}O(nm^2)$	$2^{Rk}O(nm)$	$2^{Rk}O(n^2mk)$	$2^{Rk}O(n^2mk)$
CDel	$2^kO(nm^2)$	$2^kO(nm)$	$2^kO(n^2mk)$	$2^kO(n^2mk)$

Table 3.1: Summary of algorithms for CC-Winner Determination

Related Work. Our work builds primarily on two lines of work from before. We appeal to the known algorithms that determine the optimal Chamberlin-Courant committees on single-peaked profiles (BSU13) and single-crossing profiles (SYFE15). These results have been extended to other multiwinner voting rules, which we do not consider in the present work. Also, efficient algorithms have been shown on more general preference restrictions such as single-peakedness on trees, or single-crossing width.

3.2 Preliminaries

In this section, we introduce some of the notation and definitions that we will use. For a more detailed introduction to notions relating to restricted domains and voting rules, we refer the reader to the appropriate chapters in (BCE+16), and for a comprehensive introduction to parameterized algorithms, we refer the reader to (CFK+15).

For a positive integer ℓ, we denote the set $\{1, \ldots, \ell\}$ by $[\ell]$. We first define some general notions relating to voting rules. Let $V = \{v_i : i \in [n]\}$ be a set of n *voters* and $C = \{c_j : j \in [m]\}$ be a set of m *candidates*. If not mentioned otherwise, we denote the set of candidates, the set of voters, the number of candidates, and the number of voters by C, V, m, and n respectively.

Every voter v_i has a *preference* $\succ_i$ which is a complete order over the set C of candidates. We say voter v_i prefers a candidate $x \in C$ over another candidate $y \in C$ if $x \succ_i y$. We denote the set of all preferences over C by $\mathcal{L}(C)$. The n-tuple $(\succ_i)_{i \in [n]} \in \mathcal{L}(C)^n$ of the preferences of all the voters is called a *profile*. Note that a profile, in general, is a multiset of linear orders. For a subset $M \subseteq [n]$, we call $(\succ_i)_{i \in M}$ a sub-profile of $(\succ_i)_{i \in [n]}$. For a subset of candidates $D \subseteq C$, we use $\mathcal{P}|_D$ to denote the projection of the profile on the candidates in D alone. A *domain* is a set of profiles.

The rest of this section is organized as follows. We define the Chamberlin-Courant voting rule. We then define the problems that we will study subsequently.

Chamberlin-Courant. The Chamberlin–Courant voting rule is based on the notion of a *dissatisfaction function* or a *misrepresentation function*. This function specifies, for each $i \in [m]$, a voter's dissatisfaction from being represented by candidate she ranks in position i.

Definition 9. *For an* m*-candidate election, a* dissatisfaction function *is given by a non-decreasing function* $\alpha : [m] \to \mathbb{N}$ *with* $\alpha(1) = 0$.

A popular dissatisfaction function is Borda, given by $\alpha^m_B(i) = \alpha_B(i) = i - 1$. We now turn to the notion of an assignment function. Let k be a positive integer. A k-*CC-assignment function* for an election $E = (C, V)$ is a mapping $\Phi : V \to C$ such that $\|\Phi(V)\| \leq k$. For a given assignment function Φ, we say that voter $v \in V$ is *represented* by candidate $\Phi(v)$ in the chosen committee. There are several ways to measure the quality of an assignment function Φ with respect to a dissatisfaction function α; we use the following two:

1. $\ell_1(\Phi) = \sum_{i=1,\ldots,n} \alpha(pos_{v_i}(\Phi(v_i)))$, and
2. $\ell_\infty(\Phi) = \max_{i=1,\ldots,n} \alpha(pos_{v_i}(\Phi(v_i)))$.

We are now ready to define the Chamberlin-Courant voting rule, which is the primary focus of this paper.

Definition 10. *For every family of dissatisfaction functions* $\alpha = (\alpha^m)^\infty_{m=1}$*, and every* $\ell \in \{\ell_1, \ell_\infty\}$*, the* α-ℓ-CC voting rule *is a mapping that takes an election* $E = (C, V)$ *and a positive integer* k *with* $k \leq \|C\|$ *as its input, and returns a* k*-CC-assignment function* Φ *for* E *that minimizes* $\ell(\Phi)$ *(if there are several optimal assignments, the rule is free to return any of them).*

Chamberlin and Courant (CC83) originally proposed the utilitarian variants of their rules with a focus on the Borda dissatisfaction function. The egalitarian variant was considered by, for instance, Betzler et al. (BSU13).

We refer the reader to Section 2.1 and Section 2.3 to recall the definitions of single-peaked, single-crossing domains and their generalization (composite single peaked and r-crossing profiles), and concepts in parameterized complexity respectively.

Nearly Structured Domains. Let $D = \{SP, SC\}$ be a fixed domain, where SP refers to single-peaked domains, and SC denotes single-crossing domains. We say that a profile P over candidates C has a candidate (voter) modulator of size k to D if there exists a subset of at most k candidates (voters) such that the restriction of the profile to all but the chosen candidates (voters) belongs to the domain D. Whenever a profile admits a k-sized candidate modulator to D, we say that it is k-*close to* D *via candidates*. The notion of being k-*close to* D *via voters* is analogously defined.

The work of (ELP13), (BCW16) shows that it is polynomial-time to find the smallest candidate (voter) modulator to the domain of single-peaked (single-crossing) profiles respectively. The work of (EL14) addressed the NP-hard variants and showed 2-approximation and 6-approximation algorithms for finding the smallest voter and candidate modulator to the domains of single-peaked and single-crossing profiles, respectively. Therefore, in all our problem formulations, we assume that we are given an instance of an election with a modulator to either domain as a part of the input — since it is tractable to find such modulators in all cases.

Problem Definition. We now define the main problem that we address in this work, which we denote by $\ell, \mathcal{D}$-CC Via , where ℓ is an aggregation function, $\mathcal{D}$ is a domain and is either candidates or voters, referring to the type of the modulator we are given as a part of the input.

> $\ell, \mathcal{D}$-CC Via
> **Input:** An election $E = (C, V)$, a committee size b, a target misrepresentation score R, a misrepresentation function , and a k-sized modulator X to the domain $\mathcal{D}$.
> **Parameter:** k
> **Question:** Is there a committee of size b whose ℓ-misrepresentation score under the function is at most R?

3.3 Tractability on Nearly Structured Preferences

The goal of this section is to establish the following theorem.

Theorem 1. *For all $\ell \in \{\ell_1, \ell_\infty\}$ and for all $\mathcal{D} \in \{SP, SC\}$, the $(\ell, \mathcal{D})$-CC Via Candidates problem is in* FPT *and the $(\ell, \mathcal{D})$-CC Via Voters problem is in* XP.

3.3.1 Overall Approach

We describe now informally our overall approach for solving the $(\ell, \mathcal{D})$-CC Via problem. First, we brute force through all possible "behaviors" of the solution on the modulator. Next, instead of solving the "vanilla" Chamberlin-Courant optimization problem on the part of the profile that is structured (according to the domain $\mathcal{D}$), we adapt our solution to account for the guessed behavior on the modulator. For ease of presentation, we define an intermediate auxiliary problem, which is an extension version of the original problem, described below.

3.3.2 Problem Setup

In the extension problem corresponding to $(\ell, \mathcal{D})$, we are given, as usual, an election $E = (C, V)$, a committee size b, a target misrepresentation score R and a misrepresentation function . In addition, we are also given a subset of candidates X with size at most k and a partition of X into G and B. The promise is that the election induced by the votes V when restricted to the candidates $C \setminus X$ is structured according to the domain $\mathcal{D}$. The goal is to find an optimal Chamberlin-Courant committee among the ones that contain all candidates in G and contain none of the candidates in B. The formal definition is as follows. In the following, we say that a committee respects a partition $(D \uplus G \uplus B)$ of the candidate set C if it contains all of G and none of B.

> $(\ell, \mathcal{D})$-CC Extension
> **Input:** An election $E = (C, V)$, a partition of the candidates into $(D \uplus G \uplus B)$, a committee size b, a target misrepresentation score R, a misrepresentation function ; such that the election induced by (D, V) belongs to the domain $\mathcal{D}$.
> **Question:** Is there a committee of size b that respects $(D \uplus G \uplus B)$ and whose ℓ-misrepresentation score under the function is at most R?

Before describing how to solve the $(\ell, \mathcal{D})$-CC Extension problem, we first establish that it is indeed useful for solving the $(\ell, \mathcal{D})$-CC Via problem. Let $\mathcal{D}$ be a fixed domain from $\{$Single-Peaked, Single-Crossing$\}$. First, consider the $(\ell, \mathcal{D})$-CC Via problem where we are given a k-sized candidate modulator as input, or that is fixed to be candidates. Let $(E = (C, V), b, R, \ , X)$, denoted by I, be an instance of $(\ell, \mathcal{D})$-CC Via . Recall that X is a candidate modulator to the domain $\mathcal{D}$, in other words, the election induced by $(C \setminus X, V)$ has the structure of $\mathcal{D}$. Our algorithm proceeds as follows. For a subset of candidates $Y \subseteq X$, let:

$$J_Y := (E = (C, V); (C \setminus X, Y, X \setminus Y), b, R, \ ).$$

If J_Y is a Yes-instance of $(\ell, \mathcal{D})$-CC Extension for some $Y \subseteq X$, then our algorithm returns Yes and aborts. If, on the other hand, for every subset $Y \subseteq X$ of candidates it turns out that J_Y is a No-instance of $(\ell, \mathcal{D})$-CC Extension, then we return No. It is easy to see that whenever the algorithm returns Yes, assuming the correctness of the $(\ell, \mathcal{D})$-CC Extension procedure used, there exists a committee that has the desired misrepresentation score.

To argue the correctness of the algorithm, we show that if I is a Yes-instance then the algorithm does indeed produce a committee that can achieve the desired misrepresentation score. To this end, let $C^\star$ be a committee whose ℓ-misrepresentation score under the function is at most R. Let $Y^\star$ denote $C^\star \cap X$. Then note that $C^\star$ is a committee that respects the partition $D := C \setminus X$, $G := Y^\star$, and $B := X \setminus Y^\star$. Further, note that since X is a candidate modulator to $\mathcal{D}$, the election induced by (D, V) belongs to the domain $\mathcal{D}$. Clearly, the instance $(E = (C, V); (D, G, B), b, R, \ )$ is a well-formed input to the $(\ell, \mathcal{D})$-CC Extension problem, and $C^\star$ is a valid solution to it. Assuming again the correctness of the $(\ell, \mathcal{D})$-CC Extension procedure used, we are done. Observe that the running time of our algorithm here is $2^k q(n, m)$, where $q(n, m)$ is the time required by the $(\ell, \mathcal{D})$-CC Extension procedure on an instance of size $n + m$.

We now turn to the $(\ell, \mathcal{D})$-CC Via problem where we are given a k-sized voter modulator as input, or that is fixed to be voters. Here a direct brute-force approach as in the previous case does not suggest itself, because of which we suffer a greater overhead in our running time. For simplicity, we first describe our algorithm for the egalitarian variant, that is, we fix $\ell = \ell_\infty$. We later describe the changes we need to make when we deal with the utilitarian variant.

Let $(E = (C, V), b, R, \ , X)$, denoted by I, be an instance of (ℓ, D)-CC VIA . Recall that X is a voter modulator to the domain D, in other words, the election induced by $(C, V \setminus X)$ has the structure of D. For every voter, we guess the candidate who represents that voter in an arbitrary but fixed, and valid, Chamberlin-Courant committee. For such a guess , let Y denote the set of at most k candidates who have been chosen to represent the voters in the modulator. More specifically, a voter $v \in X$, let (v) denote the candidate that we have guessed as the representative for the voter v, and let $d(v)$ denote the set of candidates ranked higher than (v) by the voter v. Note that Y is simply $\cup_{v \in X} (v)$.

We first run the following easy sanity check: if, for $u, v \in X$, $u \neq v$, we have that $(v) \in d(u)$, then we reject the guess Y. Otherwise, define $B := \cup_{v \in X} d(v)$ and $G := Y$, and let $D := C \setminus (G \cup B)$. Observe that B and G are disjoint because of the sanity check. Further, let:

$$J := (E = (C, V \setminus X); (D , G , B), b, R,).$$

It is easily checked that J is a well-formed instance for (ℓ, D)-CC EXTENSION. As before, we return YES if and only if there exists a guess for which J is a YES instance of (ℓ, D)-CC EXTENSION. To see the correctness of this approach, let $C^\star$ be a committee whose ℓ-misrepresentation score under the function is at most R. For each voter $v \in X$, let $^\star(v)$ denote the top-ranking candidate from $C^\star$ in the vote of v. Let $Y^\star$ be given by $\cup_{v \in X} {}^\star(v)$, and let $B^\star$ be the set of all candidates ranked higher than $^\star(v)$ in the votes v from X. Observe that $C^\star$ does not contain any candidates from $B^\star$ by the definition of *.

Now, as before, define: $G := Y^\star$, $B := B^\star$, and $D := C \setminus (G \cup B)$. Clearly, the instance $(E = (C, V \setminus X); (D, G, B), k, R, \ )$ is a well-formed input to the (ℓ, D)-CC EXTENSION problem, and $C^\star$ is a valid solution to it. Assuming again the correctness of the (ℓ, D)-CC EXTENSION procedure used, we are done. Observe that the running time of our algorithm here is $n^k q(n, m)$, where $q(n, m)$ is the time required by the (ℓ, D)-CC EXTENSION procedure on an instance of size $n + m$. For the utilitarian version of the problem (where $\ell = \ell_1$), the procedure is identical, except that we use R' instead of R in the definition J , where R' is $R - R_{X,}$, and $R_{X,}$ is the sum of the misrepresentation score of the candidate (v) with respect to the voter v, and the sum is over $v \in X$. It is easily verified that the other details work out in the same fashion.

The rest of this section is section is devoted to showing that the (ℓ, D)-CC EXTENSION problem can be solved in polynomial time by adapting suitably the known algorithms for the Chamberlin-Courant problem on the relevant domain D. These adaptations are sometimes subtle and in particular for the single-peaked case, we have to treat the utilitarian and the egalitarian variants separately (corresponding to $\ell = \ell_1$ and $\ell = \ell_\infty$ respectively).

3.3.3 (ℓ, D)-CC EXTENSION for the Single-Crossing Domain

In this section we demonstrate a polynomial time algorithm for the (ℓ, D)-CC EXTENSION problem for the case when D = SC. This builds closely on the algorithm shown by (SYFE15). First, we show a structural property which is an easy adaptation of Lemma

5 in (SYFE15). The statement corresponding to single-crossing profiles states that there is an optimal committee for which an optimal assignment assigns candidates in contiguous blocks over the single-crossing order. For the (ℓ, D)-CC Extension problem, this continues to be the case for candidates c from D except that some voters in the contiguous block may be assigned to candidates in G instead of being assigned to c. We now state this formally. In the statement below, an optimal b-CC assignment is considered only among committees that respect the semantics of (D, G, B) in the given instance I of (ℓ, D)-CC Extension.

Lemma 1. *Let $I = (E = (C, V); (D, G, B), b, R, \rho)$ be an instance of (ℓ,SC)-CC Extension. Suppose $V = (v_1, \ldots, v_n)$ is the single-crossing order of the votes and $C = (c_1, \ldots, c_m)$ is an ordering of the candidates according to v_i. Then for every $b \in [m]$, every dissatisfaction function ρ for m candidates, and for every $\ell \in \{\ell_1, \ell_\infty\}$, there is an optimal b-CC assignment Φ for E under ρ-ℓ-CC such that for each candidate $c_i \in D$, if $\Phi^{-1}(c_i) \neq \emptyset$, then there are two integers e_i and f_i, with $e_i < f_i$, such that for every vote v in the set of voters $V' = \{v_{e_i}, v_{e_i+1}, \ldots, v_{f_i}\}$, $\Phi(v) \in \{c_i\} \cup G$. Moreover, for each $i < j$ such that $\Phi^{-1}(c_i) \neq \emptyset$ and $\Phi^{-1}(c_j) \neq \emptyset$, it holds that $e_i < f_i$.*

We next provide the intuition for the proof of the Lemma above, and refer the reader to the proof of (SYFE15, Lemma 5) as the proof is along similar lines. Consider the case of pristine SC election instance (which is election $E' = (D, V)$ in our case). Observe that if there are voters u, v, w appearing in that order in the single-crossing ordering, and for two candidates $c_1, c_2 \in D$, if u and w were to be assigned to c_1 and v were to be assigned to c_2, then this would imply that $c_1 \succ_u c_2$ and $c_1 \succ_w c_2$, while $c_2 \succ_v c_1$, violating the single-crossing structure of the election restricted to D. Hence, for SC elections the 'contiguous block' property holds for the assignment function. In the above instance I since the only other assignments allowed are to candidates in G (as these candidates already belong to the committee), the claim follows.

We now have the following natural consequence.

Lemma 2. *(ℓ,SC)-CC Extension admits a polynomial time algorithm, both for when $\ell = \ell_1$ and when $\ell = \ell_\infty$.*

Proof. (Sketch) Lemma 1 suggests a natural dynamic programming over the SC ordering of voters and ordering of candidates according to v_1 (the first voter in single-crossing ordering). For the Single-Crossing profiles our algorithm is essentially a modified version of the dynamic programming routine which was originally developed in (SYFE15). Here, for $i \in \{0\} \cup [n]$, $j \in [m - |G| - |B|]$ and $t \in b - |G|$, we define $A[i, j, t]$ as the best possible misrepresentation score that can be achieved by a committee of size $t + |G|$ that respects the semantics of (G, B, D) formed using a subset of first j candidates considering the first i votes, where the candidates of D are ordered according to the ranking of the first voter in the single-crossing ordering and the voters are ordered according to the single-crossing ordering. The recurrence for single-crossing orders works by "guessing" the first voter v to be represented by the candidate c_j, and the optimal representation of the preceding voters is found recursively. In our setting, this approach continues to work, except that instead of simply adding up the misrepresentation score of c_j for all voters in the interval starting from v and ending at v_i, we check (for every vote in this interval) if there is a candidate from G who is ranked above c_j, and appropriately adjust the calculation of the misrepresentation score for such voters. The time complexity of above algorithm turns out to be $O(mn^2k)$ (as calculating the misrepresentation score for each voter can take $O(n)$ time). For the detailed description and the correctness of the approach we refer the reader to (SYFE15, Theorem 6). □

3.3.4 (ℓ, D)-CC Extension for the Single-Peaked Domain

For the single-peaked domain, as alluded to earlier, we need to consider the utilitarian and egalitarian variants separately. We first consider $\ell = \ell_1$.

In the following discussion the terms *first* and *last* are with respect to the societal order, which we denote by A. A candidate c_i is said to be *smaller* than another candidate c_j if the candidate c_i appears before c_j in the societal order A, and a candidate is said to be *larger* if it appears after the other candidate. Betzler et al. (BSU13) proposed separate algorithms for the utilitarian and egalitarian variants. To solve (ℓ, D)-CC Extension in this setting, we extend the dynamic programming algorithm proposed by Betzler et al for the utilitarian setting.

Lemma 3. *$(\ell_1$,SP)-CC Extension admits a polynomial time algorithm.*

Proof. Recall that we are given an instance $(E = (C, V); (G, B, D), b, r, R, \ell)$ of $(\ell_1$,SP)-CC Extension. If $b = |G|$, then there is nothing to do. If $b > |G|$, we assume without loss of generality that there is at least one voter whose top candidate does not belong to G, otherwise we may simply return Yes since the committee G is already good enough for any reasonable R[4]. The main semantics of the DP table employed previously is the following. For $i \in [m]$ and $j \in 1, \ldots, \min(i, k)$, we define $z(i, j)$ to be the total misrepresentation for a set of j winners from $\{c_1, \ldots, c_i\}$ including c_i. The final answer is given by $\min_{i \in \{k,\ldots,m\}} z(i, k)$.

Let d denote $|D|$ and $c_1 \succ c_2 \succ \cdots \succ c_d$ be the single-peaked order. As before, for $i \in [m]$ and $j \in 1, \ldots, \min(i, k)$, we define a modified DP table as follows: let $z(i, j)$ be the total misrepresentation for a set of j winners from $\{c_1, \ldots, c_i\}$ including $\{c_i\} \cup G$. Note that now the final answer is given by $\min_{i \in \{b',\ldots,m\}} z(i, b')$, where $b' = |G| - b$. Observe that our solution respects the partition (G, B, D), since the semantics of z is such that we pick all the candidates from G and do not include any candidate from B. Towards describing the recurrence, we establish some notation. First, let $g(v)$ denote the highest-ranked candidate from G in the preference ordering of the voter v. Also, define:

$$g(p, i) := \sum_{v \in V} \max\{0, \min\{r(v, c_p) - r(v, c_i), r(v, g(v)) - r(v, c_i)\}\}$$

Intuitively, $g(p, i)$ gives the potential gain of assigning candidate i to the voter v, assuming that the voter v was previously assigned to either the candidate c_p or $g(v)$. Both $g(i)$ and $g(p, i)$ can be precomputed in time $O(nm^2)$ by performing one pass over the votes and two passes over the candidates. We are now ready to describe the main recurrence:

[4] If $R < (1) \cdot n$, for instance, then it is already impossible to achieve for any committee.

$$z[i,j] = \min_{j-1 \leqslant p \leqslant i-1} z[p, j-1] - g(p,i) ,$$

with the base case:

$$z[i,1] = \min(r(v,c_i), r(v, g(v))).$$

Our argument for correctness only focuses on the part that needs to be adapted appropriately from the proof of (BSU13). Let C be a committee that witnesses the value of $z[i,j]$. Let p be the largest index smaller than i (in the societal ordering) which is such that $c_p \in C$ and let $g(v)$ be c_q. If for a voter v it holds that $r(v,c_i) < r(v,c_p)$ and $r(v,c_i) < r(v,c_q)$, then note that $r(v,c_i) < r(v,c_t)$ for all $t < p$. Then the contribution of such a voter v to the misrepresentation of $z[p, i-1]$ is $\min(r(v,c_p), r(v,c_q))$. This implies that the improvement in the misrepresentation score of this voter obtained by reassigning the voter to the candidate c_i is precisely given by $g(p,i)$. For all other voters, an assignment to c_i does not improve their misrepresentation, so the algorithm does nothing in these situations. The correctness follows from the fact that the algorithm tries all possible values of p, and the inductively assumed correctness of $z[p, j-1]$. The time complexity of the core algorithm is $O(m^2)$, as both i and j can take at most m values, coupled with the time to precompute $d(p,i)$ and $g(i)$, the total time complexity is $O(nm^2)$. □

We now turn to the egalitarian version of the rule, that is, $\ell = \ell_1$. Here again, the solution involves a straightforward adaptation of the approach of (BSU13) to account for the constraints imposed by the semantics of (G, B, D) in the extension problem.

Lemma 4. *(ℓ_1,SP)-CC Extension admits a polynomial time algorithm.*

Proof. (Sketch.) Let q be the largest integer for which $(q) \leqslant R$. We first remove voters who have a candidate from G in their top q positions. Let V' denote the remaining set of voters. For a voter $v \in V'$, let $T_q(v)$ denote the top q candidates in v's ranking. Consider the set $M(v) := T_q(v) \setminus B$. Note that any valid committee must contain a candidate from $M(v)$ for all $v \in V'$. However, observe that the set $M(v) \subseteq D$, and therefore forms a continuous interval on the societal ordering of candidates in D. Therefore our problem reduces to finding a clique cover of size at most $b - |G|$ on the interval graph that is naturally defined by the votes in V', which can be found in time $O(nm)$. □

3.4 Hardness for Generalized Restrictions on the Domain

3.4.1 3-Crossing Profiles

To show the hardness of computing an optimal ℓ_1-CC committee on 3-crossing domains, we reduce from the following variant of SAT, which is called LSAT. In an LSAT instance, each clause has at most three literals, and further the literals of the formula can be sorted such that every clause corresponds to at most three consecutive literals in the sorted list, and each clause shares at most one of its literals with another clause, in which case this literal is extreme in both clauses. The hardness of LSAT was shown in (ABC⁺15). For ease of description, we will assume in the following reduction that every clause has exactly three literals, although it is easy to see that the reduction can be extended to account for smaller clauses as well. We refer the reader to section 2.1 to recall the definitions of single-crossing domain and its generalization.

Theorem 2. *Computing an optimal ℓ_1-CC committee with respect to the Borda misrepresentation score is* NP*-hard even when the domain is three-crossing domain.*

Proof. We will first describe our construction.

Construction. Let φ be an instance of LSAT with variables $x_1, \ldots, x_n$ and clauses $C_1, \ldots C_m$. Without loss of generality, let us assume that the ordering of the clauses in the LSAT instance is also given by $C_1, \ldots, C_m$. Towards constructing the election instance, we introduce one candidate for every literal in φ. Let p_i and q_i denote the candidates corresponding to the variable x_i. We also introduce $(n+1)$ dummy candidates for each variable (which is a total of $n(n+1)$ dummy candidates). Let $d[i,j]$ denote the j^{th} dummy candidate corresponding to the variable x_i. We use C to denote the $2n$ candidates corresponding to the literals, and D to denote the set of dummy candidates. The set of candidates for the constructed election instance is $C \cup D$.

Towards describing the votes, let us fix an ordering $\succ$ on the candidates as follows. The first $2n$ candidates are from C arranged according to the LSAT ordering. The last $n(n+1)$ candidates are from D and are arranged in an arbitrary but fixed order. For a subset of candidates X, the notation $\overline{X}$ refers to an ordering of X according to $\succ$. We would now like to setup the votes in such a way that a winning committee corresponds to a valid satisfying assignment. For $1 \leqslant i \leqslant m-1$, let G_i denote literals in the set $C_i \cap C_{i+1}$, while we let G_m denote the literals in C_m. We are now ready to describe the votes. For every $1 \leqslant i \leqslant m$, we introduce the vote v_i, which has the literals of the clause C_i in the top three positions, and the remaining candidates are ranked as follows:

$$v_i := \overline{G_i} \succ \overline{G_{i+1}} \succ \cdots \succ \overline{G_m} \succ \overline{G_{i-1}} \succ \cdots \succ \overline{G_1} \succ \overline{D}$$

It is useful to note that the vote v_{i+1} can be thought of as a ranking obtained from the vote v_i by "pushing back" the tuple $\overline{G_i}$ to just behind $\overline{G_m}$. Therefore, the ordering among the G_i's in v_m is reverse of their ordering in v_1. Observe that if a literal occurs in $C_i \setminus C_{i+1}$, then it appears among the top three positions of both v_i and v_{i+1}.

We now turn to the second part of our profile, which consists of votes corresponding to the variables. Here, for a subset of candidates X, we will use $\overline{\overline{X}}$ to refer to an ordering of X according to v_m. Now, for every variable x_i, we introduce the following $(n+1)$ votes, with $1 \leqslant j \leqslant (n+1)$.

$$v_{i,j} := d[i,j] \succ p_i \succ q_i \succ \overline{\overline{(C \setminus \{p_i, q_i\})}} \succ \overline{D \setminus \{d[i,j]\}}$$

This completes a description of the profile. We fix the Borda misrepresentation target score at two and the committee size is set to n.

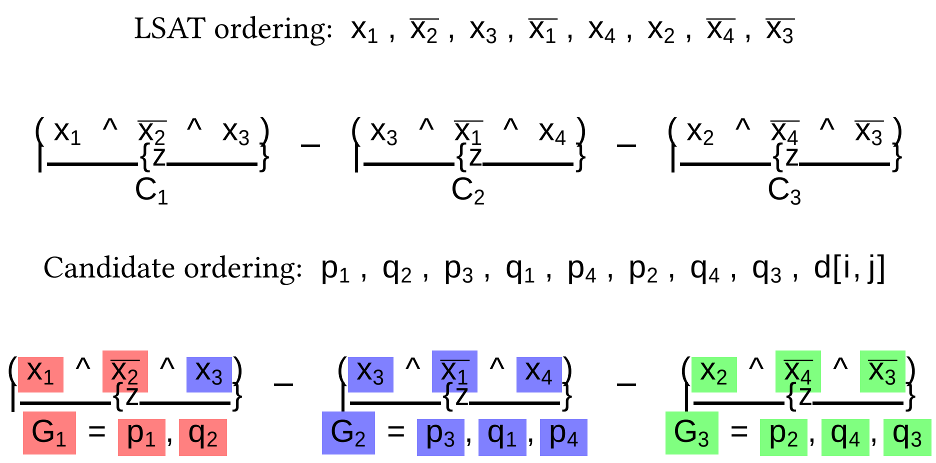


Figure 3.2: Example of construction of election instance from LSAT

Next we show the proof of equivalence.

Forward direction. We simply pick the literals corresponding to a satisfying assignment. If a satisfying assignment does not set a variable, then we pick either p_i or q_i. This clearly satisfies every vote v_i based on a clause (since otherwise, the assignment would not be a satisfying one), and trivially satisfies the votes $v_{i,j}$ based on variables since we pick exactly one of p_i or q_i for each $1 \leq i \leq n$.

Reverse direction. Let W be a committee whose score is at most two. Observe that W must choose at least one of p_i or q_i, for all $1 \leq i \leq n$. Indeed, if not, then such a committee is forced to pick every $d[i,j]$, $1 \leq j \leq n+1$, which is a violation of the committee size. Since the committee has at most n candidates, it follows by a standard pigeon-hole argument that $|W \cap \{p_i, q_i\}| \leq 1$ for all $1 \leq i \leq n$. Therefore, the committee corresponds naturally to an unambiguous assignment of the variables. It is easily checked that this satisfies every clause, because an unsatisfied clause c would correspond to a voter $v(c)$ whose Borda misrepresentation score would exceed two. This completes the proof.

We now turn to the analysis of the constructed profile.

Analysis of the profile. Let us use W_1 to denote the votes $v_1, v_2, \ldots, v_m$ and W_2 to denote $V \setminus W_1$. Further, we use P to denote the candidates corresponding to the positive literals of ϕ, that is, $P = \{p_1, \ldots, p_n\}$, Q to denote the candidates corresponding to the negative literals, and D, as before, to denote the set of dummy candidates. Consider any pair of candidates $a, b \in C$, such that $a \neq b$ then, the following cases arise.

1. $a \in P, b \in P$. Let $a = p_i$ and let $b = p_j$. Without loss of generality, let $i < j$. We further consider the following two cases:

   (a) Suppose p_i appears before p_j in σ, that is, $p_i \succ p_j$ with respect to the LSAT ordering σ. Let t_i be the unique index such that $p_i \in G_{t_i}$, and similarly let t_j be the unique index such that $p_j \in G_{t_j}$. Note that $t_j > t_i$ if $p_i \succ p_j$ with respect to σ. If $t_i = t_j$ then it is easy to verify that $p_i \succ p_j$ in all the votes $v_1, \ldots, v_m$. On the other hand, suppose, $t_j > t_i$ then observe that $p_i \succ p_j$ in all votes $v_1, v_2, \ldots, v_{t_i}$ and $p_j \succ p_i$ in all votes $v_{t_i+1}, \ldots v_m$. This follows from the construction of the votes and the fact that the preferences between

p_i and p_j have flipped in v_m relative to v_1. Indeed, p_i "crosses" p_j when v_{t_i+1} is obtained from v_{t_i} by moving $G_{t_{i+1}}$ to before G_{t_i}. Therefore in all situations, p_i and p_j cross at most once amongst the votes in W_1.

To analyze the worst case crossings, suppose that $p_j \quad p_i$ in v_m. Note that the ordering between p_i and p_j is the same as their ordering in v_m in all the following votes:

$$v_{1,1}, \ldots, v_{1,n+1}, \ldots, v_{i-1,1}, \ldots, v_{i-1,n+1}.$$

Observe that $p_i \quad p_j$ among all the votes from set f$v_{i,1}, \ldots, v_{i,n+1}$g, since p_i is in the second position and q_i is in the third position for these votes. Further, in all the remaining votes, it is easily checked that $p_j \quad p_i$. Indeed, in the votes $v_{j,1} \ldots, v_{j,n+1}$, p_j is already in the second position from the top, and a dummy candidate occupies the top position, which agrees with $p_j \quad p_i$. For any other vote after $v_{i,n+1}$, the relative ordering between p_i and p_j is identical to their ordering in v_m. This gives overall three crossings between any p_i, p_j.

The other situation, when $p_i \quad p_j$ in v_m admits a similar analysis, although with two crossings in total instead of three. Here, however, the point of crossing is at the chunk of votes corresponding to v_j rather than v_i from set W_2.

(b) For the case when $q_j \quad p_i$ in , the similar analysis shows at most three crossings.

2. $a\ 2\ Q, b\ 2\ Q$. The proof is exactly similar to case 1.

3. $a\ 2\ P, b\ 2\ Q$ with $a = p_i$ and $b = q_j$.

(a) Suppose $p_i \quad q_j$ with respect to . Let t_p be the unique index such that $p_i\ 2\ G_{t_p}$, and t_q be the unique index such that $q_j\ 2\ G_{t_q}$. Note that $t_p > t_q$ for $p_i \quad q_j$. Following the argument in Case 1, where $t_j > t_i$ implied p_i and p_j crossing at most once in W_1, we can say that p_i and q_j also cross at most once in W_1 since $t_p > t_q$. Now looking at W_2 , for the case when $i = j$ that is $a = p_i$ and $b = q_i$. Suppose that $q_j \quad p_i$ in v_m. Note that the ordering between p_i and q_j is the same as their ordering in v_m in all the following votes:

$$v_{1,1}, \ldots, v_{1,n+1}, \ldots, v_{i-1,1}, \ldots, v_{i-1,n+1}.$$

Observe that in $p_i \quad q_j$ in all the votes $v_{i,1}, \ldots, v_{i,n+1}$, since p_i is at the second position and a dummy candidate occupies the top position. Further, in all the remaining votes, it is easily checked that $q_j \quad p_i$. Indeed, in the votes $v_{j,1}, \ldots, v_{j,n+1}$, q_j is already in the third position from the top, p_j is at the second position from the top and a dummy candidate occupies the top position, which agrees with $q_j \quad p_i$ if $i \neq j$. For any other vote, the relative ordering between p_i and q_j is identical to their ordering in v_m. Hence, p_i and q_j cross twice in W_2, with the crossing points on either side of the chunk of votes corresponding to v_i. In the case where $i = j$, i.e. $a = p_i$ and $b = q_i$, $q_i \quad p_i$ in all the votes $v_{1,1}, \ldots, v_{i-1,n+1}$ following the ordering in v_m. $p_i \quad q_i$ in all the votes $v_{i,1}, \ldots, v_{i,n+1}$ by virtue of the construction, and $q_i \quad p_i$ in all the remaining votes following the ordering in v_m. This shows that p_i and q_j cross exactly twice in W_2, the crossing point being the chunk of votes corresponding to v_i. Hence, overall p_i and q_j have at most three crossings.

For the other situation, when $p_i \quad q_j$ in v_m, p_i and q_j do not cross in W_2. In all the votes $v_{1,1}, \ldots, v_{i-1,n+1}$; $p_i \quad q_j$ following their ordering in v_m. In the votes $v_{i,1}, \ldots, v_{i,n+1}$ also $p_i \quad q_j$ since p_i is at the second position from top and a dummy candidate occupies the top position. In all the votes $v_{i+1,1}, \ldots, v_{j-1,n+1}$, again $p_i \quad q_j$ following the ordering in v_m. In the votes $v_{j,1}, \ldots, v_{j,n+1}$, $q_j \quad p_i$ which constitutes one crossing at the start of this chunk of votes, and another crossing at the end; as, in all the remaining votes $p_i \quad q_j$ following the ordering in v_m. Hence, for the case when $p_i \quad q_j$ in v_m there are only two crossings instead of three.

(b) Suppose $q_j \quad p_i$ with respect to . For the case when $i \neq j$, with a similar analysis it is possible to show that there are two and three crossings according to their ordering in v_m.

4. $a \in D, b \in P$. Let $a = d[i,j]$ and let $b = p_l$, here $1 \leq i, l \leq n$ and $1 \leq j \leq n+1$. We further consider following two cases:

   (a) $i \neq l$: Notice that, $p_l \quad d[i,j]$ in . From the construction of votes in W_1, for all the votes $v_1, \ldots, v_m$ we have $p_l \quad d[i,j]$. This is because, all the dummy candidates $d[i,j]'s$ appears in the fixed are at the end of voting preferences for every voter in W_1. Hence p_l and $d[i,j]$ does not cross each other in W_1. Note that, the ordering between p_l and $d[i,j]$ is same as their ordering in v_m in all the following votes:

$$v_{1,1}, \ldots, v_{i-1,n+1}, v_{i,1}, \ldots, v_{i,j-1}$$

For vote $v_{i,j}$; $d[i,j] \quad p_l$ since $d[i,j]$ is in the topmost position for that vote. Further, in all the remaining votes, it can be easily checked that $p_l \quad d[i,j]$ as after vote $v_{i,j}$, $d[i,j]$ moves all the way back to its position in v_m. Hence, we can see that, there are two crossings between $d[i,j]$ and p_l and the crossings occur for the votes $v_{i,j}$ and $v_{i,j+1}$. Hence, in total we have 2 crossings between $d[i,j]$ and p_l.

Now consider the other case when $i = l$, here, the crossings will occur at same votes $v_{i,j}$ and $v_{i,j+1}$. The only difference is, for vote $v_{i,j}$, both $d[i,j]$ and p_l will be in top three positions of preference order, with $d[i,j]$ being on the topmost position. Again, the total number of crossings between $d[i,j]$ and p_l remains two.

5. $a \in D, b \in Q$. Let $a = d[i,j]$ and let $b = q_l$, here $1 \leq i, l \leq n$ and $1 \leq j \leq n+1$. The analysis for this part is exactly same as analysis done in part 4, with the only difference that, we have replaced p_l with q_l, hence by relabelling p_l with q_l for the above analysis, the analysis for this case can be obtained. Again, the total number of crossings for this part is same as part 4 (i.e. the total number of crossings between $d[i,j]$ and q_l is two).

6. $a \in D, b \in D$. Let $a = d[i, j]$ and $b = d[\ ,\ ]$. Without loss of generality, let $d[i, j]\ \ d[\ ,\ ]$ in .

   Note that, from our construction of candidate set we have $n(n + 1)$ total dummy candidates and total of $n(n + 1)$ votes in W_2. Hence we have a unique dummy candidate to appear at topmost position in preference order of each vote in W_2.

   Observe that for all the votes in W_1 we do not have any crossing between $d[i, j]$ and $d[\ ,\ ]$, since all the dummy candidates are in the fixed order for these votes and this fixed order is identical to the ordering of dummy candidates in .

   For the given case when $d[i, j]\ \ d[\ ,\ ]$ in , we will incur crossing between these two candidates for votes $v_{\ ,}$ and $v_{\ , +1}$ in W_2 (for $v_{[\ ,\ ]}$ we have $d[\ ,\ ]$ $d[i, j]$). It is easy to see that, for all votes before and after $v_{\ ,}$ we will have $d[i, j]$ $d[\ ,\ ]$. Hence the total number of crossings between $d[i, j]$ and $d[\ ,\ ]$ is two.

   Now consider the other situation when, $d[\ ,\ ]\ \ d[i, j]$ in . The analysis for this case is exactly same as the above analysis. The only difference is that, for this case we will incur crossing between $d[\ ,\ ]$ and $d[i, j]$ for votes $v_{i,j}$ and $v_{i,j+1}$. Therefore we have total two crossings for this case.

This completes our discussion on crossings between any pair of candidates for the constructed profile. Note that there can be at most three crossings between any pair. Hence, the profile in 3-crossing. This completes the proof of Theorem 2. □

We next move onto the utilitarian version of misrepresentation function (ℓ_1).

Theorem 3. *Computing an optimal ℓ_1-CC committee with respect to the Borda misrepresentation score is* NP*-hard even when the domain is three-crossing domain.*

Proof. To show the hardness for computing optimal committee for ℓ_1-CC, we will again show the reduction from LSAT instance.

Construction. Our construction is similar to that in Theorem 2. We construct the set C candidates in the same way, and borrow D to denote the set of dummy candidates (we will introduce these soon). We also borrow the notations and $\overline{X}$ with the same semantics.

We fix the dissatisfaction limit to $\ \ = 2\ \ $ number of voters in election instance $\ =$ $2\ \ (m + n\ \ (n + 1))$. We would now like to setup the votes in such a way that a winning committee corresponds to a valid satisfying assignment. Let denote the set of *unique* dummy candidates. The set D is the union over all such . For $1 \leqslant i \leqslant m - 1$, let G_i denote literals in the set $C_i \cap C_{i+1}$, while we let G_m denote the literals in C_m. We are now ready to describe the votes. For every $1 \leqslant i \leqslant m$, we introduce the vote v_i, which has the literals of the clause C_i in the top three positions, followed by many unique dummy candidates, and the remaining candidates are ranked as follows:

$$v_i := \overline{G_i} \quad \overline{G_{i+1}} \quad \quad \overline{G_m} \quad \overline{G_{i-1}} \quad \quad \overline{G_1} \quad \overline{D}$$

It is useful to note that the vote v_{i+1} can be thought of as a ranking obtained from the vote v_i by "pushing back" the tuple $\overline{G_i}$ to just behind $\overline{G_m}$ pull up $\overline{G_{i+1}}$ to the top.

Therefore, the ordering among the G_i's in v_m is reverse of their ordering in v_1. Observe that if a literal occurs in $C_i \cap C_{i+1}$, then it appears among the top three positions of both v_i and v_{i+1}.

We now turn to the second part of our profile, which consists of votes corresponding to the variables. Here, for a subset of candidates X, we will use $\overline{\overline{X}}$ to refer to an ordering of X according to v_m. Now, for every variable x_i, we introduce the following $(n+1)$ votes, with $1 \leq j \leq (n+1)$.

$$v_{i,j} := d[i,j] \quad p_i \quad q_i \qquad \overline{\overline{(C \setminus \{p_i, q_i\})}} \qquad \overline{D \setminus \{d[i,j] \cup \quad \}}$$

This completes the description of our profile. We will now show the equivalence.

Forward direction. In the forward direction, we simply pick the committee corresponding to a satisfying assignment for LSAT. Clearly, the misrepresentation score for this committee is at most (since every voter has at least one of his top-three candidates in the winning committee). Hence, we showed that if there exists a satisfying assignment for LSAT, then there exists a winning committee of misrepresentation at most .

Reverse direction. We need to show that if the optimal committee has misrepresentation score at most , then we can find a satisfying assignment for LSAT instance (we will show that this assignment corresponds to the optimal committee). Let W be a committee whose score is at most . Observe that W must choose at least one of p_i or q_i, for all $1 \leq i \leq n$. Indeed, if not, then such a committee is forced to pick every dummy candidate for, $1 \leq j \leq n+1$ (since any dummy candidate appears exactly once in top positions) which is a violation of the committee size. With the constraint of committee size n, it is easy to see that any committee will correspond to an unambiguous assignment of literals in LSAT instance. This committee satisfies all the clauses, indeed if not, then the committee will incur strictly more than dissatisfaction from the corresponding vote itself.

Analysis for crossings. It is easy to see that since any dummy candidate comes in top positions only for one vote, all the dummy candidates have at most 2-crossings with any other candidate. The $2n$ candidates corresponding to the literals cross once in the clause votes block and twice in the variable votes block as described in proof of Theorem 2. This gives a total of three crossings.

This completes the proof Theorem 3. □

3.4.2 3-composite single-peaked domains.

In this section, we show the hardness of computing an optimal ℓ_1-CC committee on 3-composite single-peaked domains with respect to the Borda misrepresentation score. The reduction is again from LSAT, and the construction is similar to the one used in the proof of Theorem 2 in that we again have candidates corresponding to literals and votes representing clauses. A committee corresponds to a satisfying assignment precisely when its misrepresentation score is at most two. The main difference from before is that, in this case, we order the candidates on a societal axis (single-peaked axis) and all the votes are single-peaked with respect to this axis.

Theorem 4. *Computing an optimal ℓ_1-CC committee with respect to the Borda misrepresentation score is* NP*-hard even when the domain is a three-composite single-peaked domain.*

Proof. We refer the reader to section 2.1 to recall the definition of 3-composite single-peaked domain.

Construction. Let ϕ be an instance of LSAT with variables $x_1, \ldots, x_n$ and clauses $C_1, \ldots C_m$. Towards constructing the election instance, we introduce one candidate for every literal in ϕ. Let p_1 and q_i denote the candidates corresponding to the variable x_i. We also introduce $(n+1)$ dummy candidates for each variable (which is a total of $n(n+1)$ dummy candidates). Let $d[i,j]$ denote the j^{th} dummy candidate corresponding to the variable x_i. We use C to denote the $2n$ candidates corresponding to the literals, and D to denote the set of dummy candidates. P and Q denote the candidates corresponding to the positive and the negated literals respectively.

Let us fix the ordering $\succ$ on the candidates as follows. The first $2n$ candidates are from C arranged according to the LSAT ordering. The last $n(n+1)$ candidates are from D and are arranged in an arbitrary but fixed order. Let $\succ'$ be the reverse of $\succ$. For a subset of candidates X, the notation $\overline{X}$ refers to an ordering of X according to $\succ$. For a subset of candidates $X \subseteq C$, who occupy adjacent positions in the LSAT ordering projected over C, the notation $\overrightarrow{C \setminus X}$ refers to an ordering according to $\succ$ of the candidates from $C \setminus X$ who appear after X in the LSAT ordering and similarly $\overleftarrow{C \setminus X}$ refers to an ordering according to $\succ'$ of the candidates from $C \setminus X$ who appear before X in the LSAT ordering. This notation easily yields an ordering which is single-peaked – $\overline{X} \succ \overrightarrow{C \setminus X} \succ \overleftarrow{C \setminus X}$. To represent this succinctly, we introduce the notation $\overleftrightarrow{C \setminus X}$ which refers to an ordering according to $\succ$ of the candidates from $C \setminus X$ who appear after X in the LSAT ordering followed by an ordering according to $\succ'$ of the candidates from $C \setminus X$ who appear before X in the LSAT ordering.

We would now like to setup the votes in such a way that a winning committee corresponds to a valid satisfying assignment. We introduce one vote for every clause as follows. Suppose the clause c consists of the literals (ℓ_1, ℓ_2, ℓ_3), and let the candidates corresponding to these literals be t_1, t_2, t_3 respectively. If $\ell_1 < \ell_2 < \ell_3$ in the LSAT ordering, then we introduce the following vote:

$$v(c) := t_2 \succ t_1 \succ t_3 \succ \overleftrightarrow{(C \setminus \{t_1, t_2, t_3\})} \succ \overline{D}$$

For every variable x_i, we also introduce the following $(n+1)$ votes, with $1 \leqslant j \leqslant (n+1)$:

$$v(x_i, j) := d[i,j] \succ p_i \succ q_i \succ \overleftrightarrow{(P \setminus \{p_i\})} \succ \overleftrightarrow{(Q \setminus \{q_i\})} \succ \overleftrightarrow{D \setminus \{d[i,j]\}}$$

This completes a description of the profile. We fix the Borda misrepresentation target score at two and the committee size is set to n.

Analysis of the profile. We claim that the constructed profile is three-composite single-peaked with respect to the partition (P, Q, D). First we look at $v(c)$ – the votes based on the clauses. $v(c)$ when projected on D is trivially single-peaked; when projected on

C it is single-peaked since three candidates (t_1, t_2, t_3) appearing contiguously in are placed at placed at the top with middle candidate (t_2); and hence when projected on P, Q C will remain single-peaked . Now we look at $v(x_i, j)$ – the votes based on variables. For each of these sets, we put exactly one candidate in top three positions which determines the peak for the set, and all other candidates are placed subsequently to maintain single-peakedness with respect to these peaks. Hence, the profile is single-peaked when projected over P, Q and D. We now prove the equivalence of these two instances.

For the proof of equivalence, we refer the reader to the arguments for forward and reverse directions in Theorem 2. The arguments work out in exact same way for this case.

□

We next show the hardness for the utilitarian aggregation function.

Theorem 5. *Computing an optimal ℓ_1-CC committee with respect to the Borda misrepresentation score is* NP*-hard even when the domain is a three-composite single-peaked domain.*

Proof. We combine the ideas from Theorem 3 and Theorem 4 to get the hardness for this case.

Construction. Let be an instance of LSAT with n variables and m clauses. We borrow the sets C, D with same semantics as in Theorem 4. Note that $C = P \cup Q$ where P and Q denote the candidates corresponding to the positive and negative literals respectively. We also use the notations $\overleftarrow{C \setminus X}$, $C \setminus X$ and $\overrightarrow{C \setminus X}$ defined earlier for set of candidates C, X.

We borrow the notations and from Theorem 3 denoting dissatisfaction limit and set of *unique* dummy candidates respectively. Note that D denotes the set of all dummy candidates introduced in the instance. Let us fix to be the ordering of candidates as follows. The candidates from C follow the LSAT ordering and the dummy candidates are arranged in an arbitrary but fixed ordering at the end in . Let $_P$, $_Q$ and $_D$ denote the ordering according to restricted over set of candidates P, Q and D respectively.

We would now like to setup the votes in such a way that a winning committee corresponds to a valid satisfying assignment. We introduce one vote for every clause as follows. Suppose the clause c consists of the literals (ℓ_1, ℓ_2, ℓ_3), and let the candidates corresponding to these literals be t_1, t_2, t_3 respectively. If $\ell_1 < \ell_2 < \ell_3$ in the LSAT ordering, then we introduce the following vote:

$$v(c) := t_2 \quad t_1 \quad t_3 \qquad \overrightarrow{(C \setminus \{t_1, t_2, t_3\})} \quad \overrightarrow{D \setminus}$$

For every variable x_i, we also introduce the following $(n+1)$ votes, with $1 \leq j \leq (n+1)$:

$$v(x_i, j) := d[i, j] \quad p_i \quad q_i \qquad \overrightarrow{(P \setminus \{p_i\})} \quad \overrightarrow{(Q \setminus \{q_i\})} \quad \overrightarrow{D \setminus \{ \cup d[i, j]\}}$$

This completes a description of the profile. We fix the Borda misrepresentation target score at two and the committee size is set to n.

The proof of correctness works exactly same as in case of Theorem 3. For the analysis of profile, we refer the reader to the analysis in Theorem 4.

This completes the proof for Theorem 5.

□

3.5 Conclusion and Open Problems

We have made some progress in demonstrating that the Chamberlin-Courant voting rule can be computed efficiently on nearly-structured domains, and there are some notions of being "almost structured" for which the rule remains hard. Several specific problems remain open. The most pertinent issue is whether the problem admits a FPT algorithm when parameterized by the size of a voter modulator to either single-peaked or single-crossing profiles.

Another open problem is computations of *CC-Winner* for 2-*crossing* profiles. Resolving this would complete the investigation for generalized single-crossing domains.

Chapter 4

Robustness Radius for Chamberlin-Courant on Restricted Domains

4.1 Introduction [1]

A *voting rule* is a function that maps a collection of preferences over a fixed set of alternatives to a set of winning options, where each option could be one or more alternatives — corresponding, respectively, to the scenarios of single-winner and committee elections. We refer the reader to subsection 1.1.1 for a gentle introduction to voting. A voting rule is *vulnerable to change* if small perturbations in the input profile can cause for its outcome to vary wildly. There have been several notions in the contemporary computational social choice literature that captures the degree of vulnerability of various voting rules.

A recent exercise in this direction was carried out in (BFK+17), where the notion of *robustness radius* was introduced as the minimum number of swaps that was required between consecutive alternatives to change the outcome of a multiwinner voting rule. We note here that we are implicitly assuming that preferences are modeled as linear orders over the alternatives, although the notion of swaps can be defined naturally for the situation where the votes are given by approval ballots. In the work of (BFK+17), several voting rules are considered, and efficient algorithms were proposed for Robustness Radius for many of these rules. On the other hand, for some voting rules, the problem turned out to be hard: even when the question was to decide if there is *one* swap that influences the outcome. This is the motivation for the present work: we focus on the Chamberlin-Courant voting rule (c.f. Section 2 on Preliminaries for the definition), for which Robustness Radius turns out to be intractable, and look for exact algorithms on general profiles and ask if the problem becomes easier to tackle on structured preferences.

[1] Some portions of this chapter are taken verbatim from (MS19)

4.1.1 Our Contributions and Organisation of the Chapter

Our first contribution is an explicit XP algorithm for the ROBUSTNESS RADIUS problem in the context of the Chamberlin-Courant voting rule. Recall that it is already NP-hard to determine if there exists *one* swap which changes the set of winning committees. Notice that the natural brute-force approach to check if there are at most r swaps which affect the set of winning committees is to simply try all possible ways of executing r swaps and recompute the set of winning committees at every step. This approach, roughly speaking, requires $O((mn)^r \cdot m^k)$ time. We improve this by suggesting an algorithm whose running time can be bounded by $O^{?}(m^k)$. We show this result for both the Chamberlin-Courant voting rule with the Borda misrepresentation function as well as for the approval version of the Chamberlin-Courant voting rule (where we also show that an analogous hardness result also holds).

On the other hand, we initiate an exploration of whether the ROBUSTNESS RADIUS problem remains hard on structured preferences. We provide some insights on this issue by demonstrating that the problem remains NP-hard on "nearly-structured" profiles. In particular, we show that:

1. Determining if the robustness radius of a profile is one for the ℓ_1-CC (respectively, ℓ_1-CC) voting rule, with respect to the Borda misrepresentation score, is NP-hard even when the input profiles are restricted to the six-crossing domain[2] (respectively, the four-crossing domain).

2. Determining if the robustness radius of a profile is one for the ℓ_1-CC voting rule, with respect to the Borda misrepresentation score, is NP-hard even when the domain is a four-composite single-peaked domain.

Related Work. The notion of robustness is also captured by other closely related notions, such as the margin of victory (MoV) (Xia12) and swap bribery (EFS09). In the former, the metric of change is the number of voters who need to be influenced, rather than the total number of swaps. On the other hand, in swap bribery, the goal is not to simply influence a change in the set of committees, but to ensure that a specific committee does or does not win (corresponding to constructive and destructive versions of the problem, respectively). We note that swap bribery has been mostly studied in the context of single-winner voting rules. Observe that any profile that is a non-trivial YES-instance of swap bribery is also a YES-instance of ROBUSTNESS RADIUS with the same budget, but the converse is not necessarily true. Similarly, any profile that is a YES-instance of ROBUSTNESS RADIUS is also a YES-instance of MoV with the same budget, but again the converse need not be true. However, we remark that in the case of the Approval-CC voting rule, the notions of ROBUSTNESS RADIUS and MoV happen to coincide. Robustness has also been studied for single-winner voting rules in earlier work (SYE13).

[2] We refer the reader to the Preliminaries for the definition of ℓ-single-crossing domains and to the Appendix for the definition of ℓ-composite single-peaked domains.

4.2 Preliminaries

In this section, we introduce some key definitions and establish notation. For a comprehensive introduction, we refer the reader to (BCE[+]16; End17).

Notation. For a positive integer ℓ, we denote the set $\{1, \ldots, \ell\}$ by $[\ell]$. We first define some general notions relating to voting rules. Let $V = \{v_i : i \in [n]\}$ be a set of n *voters* and $C = \{c_j : j \in [m]\}$ be a set of m *candidates*. If not mentioned otherwise, we denote the set of candidates, the set of voters, the number of candidates, and the number of voters by C, V, m, and n respectively.

Every voter v_i has a *preference* $\succ_i$ which is typically a complete order over the set C of candidates (rankings) or a subset of approved candidates (approval ballots). An instance of an election consists of the st of candidates C and the preferences of the voters V, usually denoted as $E = (C, V)$. A *multiwinner committee rule* R is a function that, given an election E and a committee size k, outputs a *family* $R(E, k)$ of k-sized subsets of C, such that each of these k-sized committees ties for the victory (i.e. each of these committees is one of the optimal committees according to R).

We now state some definitions in the context of rankings, although we remark that analogous notions exist also in the setting of approval ballots. We say voter v_i prefers a candidate $x \in C$ over another candidate $y \in C$ if $x \succ_i y$. We denote the set of all preferences over C by $\mathcal{L}(C)$. A rank of a candidate y according to the voter v_i is 1 + the number of candidates $x \in C$ such that $x \succ_i y$. The n-tuple $(\succ_i)_{i \in [n]} \in \mathcal{L}(C)^n$ of the preferences of all the voters is called a *profile*. Note that a profile, in general, is a multiset of linear orders. For a subset $M \subseteq [n]$, we call $(\succ_i)_{i \in M}$ a sub-profile of $(\succ_i)_{i \in [n]}$. For a subset of candidates $D \subseteq C$, we use $\mathcal{P}|_D$ to denote the projection of the profile on the candidates in D alone. A *domain* is a set of profiles.

Chamberlin-Courant for Rankings. The Chamberlin–Courant voting rule is based on the notion of a *dissatisfaction function* or a *misrepresentation function* (we use these terms interchangeably). This function specifies, for each $i \in [m]$, a voter's dissatisfaction from being represented by candidate she ranks in position i. A popular dissatisfaction function is Borda, given by $\alpha^m_B(i) = \alpha_B(i) = i - 1$, and this will be our measure of dissatisfaction in the setting of rankings.

We now turn to the notion of an assignment function. Let k be a positive integer. A k-*CC-assignment function* for an election $E = (C, V)$ is a mapping $\Phi : V \to C$ such that $\|\Phi(V)\| = k$, where $\|\Phi(V)\|$ denotes the image of Φ. For a given assignment function Φ, we say that voter $v \in V$ is *represented* by candidate $\Phi(v)$ in the chosen committee. There are several ways to measure the quality of an assignment function Φ with respect to a dissatisfaction function α; we use the following:

1. $\ell_1(\Phi, \alpha) = \sum_{i=1,\ldots,n} \alpha(\mathrm{pos}_{v_i}(\Phi(v_i)))$, and
2. $\ell_\infty(\Phi, \alpha) = \max_{i=1,\ldots,n} \alpha(\mathrm{pos}_{v_i}(\Phi(v_i)))$.

Unless specified otherwise, α will be the Borda dissatisfaction function described above. We are now ready to define the Chamberlin-Courant voting rule.

Definition 11 (Chamberlin-Courant (CC83))**.** *For* $\ell \in \{\ell_1, \ell_\infty\}$*, the* ℓ-CC voting rule *is a mapping that takes an election* $E = (C, V)$ *and a positive integer* k *with* $k \leq |C|$ *as its input, and returns the images of all the* k*-CC-assignment functions* Φ *for* E *that minimizes* $\ell(\Phi, \alpha)$*.*

Chamberlin Courant for Approval Ballots. Recall that an approval vote v_i on the set of candidates C is an arbitrary subset S_v of C such v approves all the candidates in S_v. We define the misrepresentation score for k-sized committee W for an approval voting profile is defined as the number of voters which do not have any of their approved candidates in W (i.e. $W \cap S_v = \emptyset$). Hence the optimal committees under approval Chamberlin Courant are the committees which maximize the number of voters with at least one approved candidate in the winning committee. This notion of Chamberlin-Courant for the setting of approval ballots was proposed by (LS18).

To recall the definitions of single-peaked, single-crossing domains, and their generalizations; we request the reader to revisit Section 2.1.

Robustness Radius. Let R be a multiwinner voting rule. For the given election $E = (C, V)$, a committee size k, and an integer r, in the R-Robustness Radius problem we ask if it is possible to obtain an election E' by making at most r swaps of adjacent candidates within the rankings in E (or by introducing or removing at most r candidates from the approval sets of voters in case of approval ballots) so that $R(E', k) \neq R(E, k)$.

We refer the reader to Section 2.3 for a refresher on Parameterized Complexity.

4.3 XP algorithm for Rankings and Approval-CC

The Robustness Radius problem for the ℓ_1-Chamberlin-Courant voting rule with the Borda dissatisfaction function is known to be in FPT when parameterized by either the number of candidates or the number of voters. For the former, the approach involves formulating the problem as an ILP and then using Lenstra's algorithm. In the case of the latter, the algorithm is based on guessing all possible partitions of the voters based on their anticipated representatives and then employing a dynamic programming approach.

In this section, we give a simple but explicit algorithm for the problem which has a XP running time in k, the committee size. This complements the W[1]-hardness of the problem when parameterized by k (BFK+17). We establish this result for both when the votes are rankings as well as when they are approval ballots. First, we address the case when the votes are rankings.

Theorem 6. *On general profiles comprising of rankings over alternatives, Robustness Radius for the* ℓ_1*-Chamberlin-Courant voting rule with the Borda dissatisfaction function admits a* $O^\star(m^k)$ *algorithm, where* m *is the number of candidates and* k *is the committee size.*

Proof. We first determine the set of all optimal committees of size k in time $O(m^k)$. Suppose there are at least two committees, say A and B, that are both optimal. The manner in which this case can be handled is also addressed in (BFK+17). For the sake of completeness, we reproduce the main point here, but in particular we do not address certain edge cases: for example, a slightly different discussion is called for if there are less than k candidates in total occupying the top positions across the votes. We refer the reader to (BFK+17) for a more detailed explanation.

Now, note that since A and B are distinct committees, there is at least one voter v whose Chamberlin-Courant representative with respect to A and B are distinct candidates: say c_a and c_b, respectively. Assume, without loss of generality, that $c_a \succ_v c_b$. Note that swapping the candidate c_b so that its rank in the vote v decreases by one results in a new profile where:

1. the dissatisfaction score of the committee B is one less than in the original profile, and,

2. the dissatisfaction score of the committee A is at least its score in the original profile (indeed; the dissatisfaction score either stays the same or increases if c_a is adjacent to c_b in the vote v).

Therefore, when there are at least two optimal committees, it is possible to change the set of winning committees with only one swap, making this situation easy to resolve. We now turn to the case when the the input profile admits a unique winning committee A. Our overall approach in this case is the following: we "guess" a committee B that belongs to the set of winning committees after r swaps (note that such a committee must exist if we are dealing with a YES-instance). For a fixed choice of B, we determine, greedily, the minimum number swaps required to make B a winning committee. We now turn to a formal description of the algorithm.

Recall that a profile Q is said to be within r swaps of a profile P if Q can be obtained by at most r swaps of consecutive candidates in P. In the following discussion, we say that a committee B is *nearly winning* if there exists a profile Q, within r swaps of P, where B is a winning committee. We refer to Q as the *witness* for B. Note that the existence of a nearly winning committee $B \neq A$ characterizes the YES-instances. Let $\Delta_{B,A}(P)$ denote the difference between the dissatisfaction scores of the committees B and A with respect to the profile P. We begin by making the following observation.

Proposition 1. *Let P and Q be two profiles such that Q can be obtained by making at most r swaps of consecutive candidates in the profile P. Note that:*

$$\Delta_{B,A}(P) - 2r \leq \Delta_{B,A}(Q) \leq \Delta_{B,A}(P) + 2r.$$

The claim above follows from the fact that if Q is a profile obtained from P by one swap of consecutive candidates in some vote of P, then it is easy to see that $\Delta_{B,A}(P) - 2 \leq \Delta_{B,A}(Q) \leq \Delta_{B,A}(P) + 2$ (in each swap we can at best increase/decease the dissatisfaction of P and decrease/increase dissatisfaction of Q). Note that if B is nearly winning, then $\Delta_{B,A}(Q) \leq 0$, where Q is the witness profile. We now have a case analysis based on $\Delta_{B,A}(P)$.

Case 1. $\Delta_{B,A}(P) > 2r$. In this case, by Proposition 1, we know that in every profile Q within r swaps of P, $\Delta_{B,A}(Q) > 0$, which is to say that B will have a greater Borda dissatisfaction score than A in every profile that is r swaps away from the input profile. Therefore, in this case, we reject the choice of B as a potential nearly winning committee.

Case 2. $\Delta_{B,A}(P) \leq r$. An analogous argument can be used to see that B is in fact nearly winning in this case. Indeed, any r swaps that improve the ranks of the candidates in B will result in a profile Q that is within r swaps of P and where $\Delta_{B,A}(Q) \leq 0$. So, B is either nearly winning with witness profile Q, or A is no longer a winning committee in Q. Therefore, in this situation, we output Yes.

Case 3. $\Delta_{B,A}(P) = r + s, 1 \leq s \leq r$. For a vote v, let $A(v)$ and $B(v)$ denote, respectively, the candidates from A and B with the highest rank in the vote v. Further, let $d_{B,A}(v)$ denote the difference between the ranks of $B(v)$ and $A(v)$. Let $W \subseteq V$ be the subset of votes for which $d_{B,A}(v) > 0$, and let $w_1, w_2, \ldots$ denote an ordering of the votes in W in increasing order of these differences. We now make the following claim.

Proposition 2. *There exists a profile Q that is r swaps away from P where $\Delta_{B,A}(Q) \leq 0$ if, and only if:*

$$t := \sum_{i=1} d_{B,A}(w_i) \leq r. \tag{4.1}$$

Proof. In the forward direction, suppose equation 4.1 holds. Then perform swaps in the votes $w_1, \ldots, w_s$ so that for any $i \in [s]$, the candidate $B(w_i)$ is promoted to the position just above $A(w_i)$. In other words, each swap involves $B(w_i)$ and in the profile obtained after the swaps, $B(w_i) \succ A(w_i)$ for all $i \in [s]$, and the difference in the ranks of these pairs is exactly one. Note that a total of t swaps are performed to obtain this profile. Denote this profile by R and note that $\Delta_{B,A}(R) = r + s - t - s = r - t$ (since the last swap made on each vote w_i reduces the gap between the dissatisfaction scores of the two committees by two). Also, $(r - t)$ is also exactly the number of remaining swaps we can still make, so a witness profile can be obtained using the argument we made in the previous case.

On the other hand, if there exists a profile Q that is r swaps away from P where $\Delta_{B,A}(Q) \leq 0$, then at least s of those swaps must have been of the form

$$A(v) \succ B(v) \rightarrow B(v) \succ A(v),$$

for some vote v, since such swaps are the only ones for which the gap between the dissatisfaction scores of the two committees reduces by two. It is easily checked that if (4.1) does not hold, and in particular, $t > r$, then it is not possible to make s swaps of this form with at most r swaps in total. This concludes the argument. □

To summarize, our algorithm in this case identifies and sorts the votes in W, and returns Yes if condition (4.1) holds, and rejects the choice of B otherwise. Observe that we output No if no choice of B results in a positive outcome in this case analysis. In terms of the running time, we require $O(m^k)$ time in distinguishing whether we have a unique winning committee or not, and if we are in the latter situation, we need $O(m^k)$ time to guess a nearly winning committee. For each choice B of a potential winning committee, we spend time $O(mn \log n)$ in the worst case to determine if B is indeed a nearly winning committee. Therefore, hiding polynomial factors, the overall running time of our algorithm is $O^{\star}(m^k)$ and this concludes the proof.

□

We now turn to $O^{\star}(m^k)$ algorithm for Robustness Radius with respect to approval ballots. The general approach is quite analogous to the setting of rankings. However, the notion of swaps is slightly different, and the overall case analysis is, in fact, simpler.

Theorem 7. *On general profiles comprising of approval ballots over alternatives, Robustness Radius for the ℓ_1-Chamberlin-Courant voting rule with the Borda dissatisfaction function admits a $O^{\star}(m^k)$ algorithm, where m is the number of candidates and k is the committee size.*

Proof. (Sketch.) Similar to Theorem 6, we first determine the set of all optimal committees of size k in time $O^{\star}(m^k)$. Let the dissatisfaction score for optimal committee be d_{opt}. Notice that in this case of Approval Chamberlin-Courant, d_{opt} is essentially equal to the number of voters for which no approved candidate appears in the winning committee.

We first consider the unique winner case. Let C be an optimal committee, and C' be a second-best committee (if there are more than one second best committees, we consider any arbitrary committees from those) according to the dissatisfaction scores. Unlike rankings, in this case, there does not exist any approval or disapproval, which can simultaneously increase dissatisfaction for C and decrease dissatisfaction for C'. Hence, the sequence of optimal approvals is to choose any vote v_i which contributes to the dissatisfaction score of C', and we add any arbitrary candidate from C' to approval set of v_i. We repeat this until the dissatisfaction score for both C and C' is equal. Hence, the robustness radius is equal to the difference between dissatisfaction scores for C and C'.

Now we turn to the case of multiple winners. In this case, we check both the optimal number of disapprovals required to remove a committee from the winning set and optimal approvals required to add a committee to the winning set. Note that the latter can be computed as described in the unique winner case above. In order to remove a committee from the winning set, we need to increase the dissatisfaction score for one of the committees. For all winning committees, we find a vote v_i such that the intersection of approved candidates from v_i and the candidates from the committee is minimum. This is the minimum number of disapprovals required to increase the dissatisfaction score of the committee at hand. We compute this minimum number of disapprovals for all winning committees and choose the minimum value. At last, we compare the "costs" for both adding and removing a committee from the winning set and choose the option which demands a lesser number of approvals or disapprovals.

This completes the overall idea of our algorithm. □

4.4 W[2]-hardness for Approval-CC

We now turn to the case of approval ballots. First, we show that the robustness radius problem in this setting remains NP-hard even for determining if the robustness radius is one, as was true for the case when the votes were rankings.

Theorem 8. *Robustness Radius for the Approval Chamberlin-Courant voting rule is* NP-*hard, even when the robustness radius is one and each voter approves at most three candidates. It is also* W[2]-*hard parameterized by the size of the committee when there are no restrictions on the number of candidates approved by a voter, and the robustness radius is one.*

Proof. We reduce from the Hitting Set problem. Note that the NP-hardness in the restricted setting follows from the fact that Hitting Set is already hard for sets of size at most two (recall that this is the Vertex Cover problem), while the W[2]-hardness follows from the fact that Hitting Set is W[2]-hard when parameterized by the size of the hitting set (CFK+15) and our reduction will be parameter-preserving with respect to the parameter of committee size.

Let $(U, F; k)$ be an instance of Hitting Set. Recall that this is a Yes-instance if and only if there exists $S \subseteq U$, with $|S| \leq k$ such that $S \cap X \neq \emptyset$ for any $X \in F$. We construct a profile P over alternatives A as follows. Let:

$$A := \underbrace{\{c_u \mid u \in U\}}_{C} \cup \underbrace{\{d_1, \ldots, d_k\}}_{D}$$

Also, for every $1 \leq i \leq k$, and for every $X \in F$, introduce a vote $v(X, i)$ that approves the candidates corresponding to the elements in X along with d_i. This completes the construction of the instance. We claim that this instance has a robustness radius of one if and only if $(U, F; k)$ is a Yes-instance of Hitting Set.

Forward Direction. Suppose S is a hitting set for (U, F) of size k. Then the set $C_S := \{c_u \mid u \in S\}$ and D are two optimal Approval-CC committees with dissatisfaction scores of zero each. Note that removing the candidate d_1 from any vote of the form $v(X, 1)$ will lead to a profile where the set of winning committees contains C_S but does not contain D. Hence, the robustness radius is indeed one.

Reverse Direction. For the reverse direction, suppose the profile P has robustness radius one. We will now argue the existence of a hitting set of size at most k. Note that D is already an optimal committee with respect to P as it has the best possible Approval-CC dissatisfaction score of zero. Now, suppose P admits another winning committee W distinct from D. Then notice that the Approval-CC dissatisfaction score of W must also be zero, and since there is at least one candidate from D (say d_i) that is not present in W, it is easy to see that the candidates in $C \cap W$ form a hitting set for the instance $(U, F; k)$ — indeed, note that every voter in the sub-profile $\{v(X, i) \mid X \in F\}$ does not approve anyone in $D \cap W$, and therefore must approve someone of in $C \cap W$, making this a hitting set for F.

Therefore, the interesting case is when D is the unique winning committee for P. We claim that any other subset of candidates W of size k has an Approval-CC dissatisfaction score of at least two. This would imply that the robustness radius of P cannot possibly be one, and therefore there is nothing to prove (note that in the case of approval voting, we cannot decrease the difference in the dissatisfaction scores of two committees by making one change in the profile). To this end, observe that $C_W := W \setminus C$ is not a hitting set[3] for F: indeed, if C_W was a hitting set then it is easy to see that W is also an optimal committee with respect to P, contradicting the case that we are in. Let X denote a set that is not hit by C_W. Now, we consider two cases:

W omits two candidates from D In this case, there are at least two candidates in D — say d_i and d_j — who do not belong to W. Then W earns a dissatisfaction score of at least one from each of $v(X, i)$ and $v(X, j)$, which makes its dissatisfaction score at least two, as desired.

W omits exactly one candidate from D In this case, notice that $|C_W| = 1$ and that C_W does not hit at least two sets, say X and Y: else C_W along with an arbitrarily chosen element from X and another chosen from Y, along with an arbitrary choice of $k - 3$ additional candidates would constitute a winning committee in P different from D, again contradicting the case that we are in. Therefore, observe that d_i is the candidate from D that is not present in W, the votes $v(X, i)$ and $v(Y, i)$ contribute one each to the dissatisfaction score of the committee W.

Overall, therefore, if D is the unique winning committee in P, then the robustness radius is greater than one, and there is nothing to prove. This concludes our argument in the reverse direction.

□

4.5 Robustness for nearly restricted domains

In this section, we explore the complexity of Robustness Radius on nearly-structured preferences. We discover that the problem remains NP-hard on slight generalizations of both single-crossing and single-peaked domains even when the robustness radius is one. We start the section with hardness results for nearly single-crossing domains and follow that with hardness on nearly single-peaked domains.

4.5.1 Domains Close to Single-Crossing Domain

We first show the hardness for determining robustness radius for utilitarian (ℓ_1) aggregation function even when the profile is 6-crossing. We note that our overall approach is similar to the one employed in (BFK+17).

[3] Note the slight abuse of terminology here: when referring to C_W as a hitting set, we are referring to the elements of U corresponding to the candidates in C_W. As long as this is clear from the context, we will continue to use this convention.

Theorem 9. *Determining if the robustness radius of a profile is one for the* ℓ_1*-CC voting rule, with respect to the Borda misrepresentation score, is* NP*-hard even when the input profiles are restricted to the six-crossing domain.*

Proof. We reduce from Independent Set on 3-regular graphs. Let $(G := (V, E), t)$ be an instance of Independent Set on 3-regular graphs where V, E denotes the set of vertices and edges of G respectively. The problem asks if there a set $T \subseteq V$ with $|T| > t$ such that for any $u, v \in T$, $(u, v) \notin E$. The hardness for the problem was shown in (FSS10). We construct a profile based on G as follows. Our set of candidates C is given by:

$$C := \underbrace{\{c_u \mid u \in V(G)\}}_{V} \cup \underbrace{\{d_1, \ldots, d_h\}}_{D} \cup \underbrace{\{Z_0, Z_1\}}_{Z} \cup \underbrace{\{x_1, \ldots, x_{t+1}\}}_{X},$$

where h is a parameter that we will specify in due course. We refer to the candidates in X as the *safe candidates* and Z_0 & Z_1 are two special candidates. We will use denote a subset of -many *unique* dummy candidates, where $:= 12nt$. Now we describe the votes. Our voters are divided into three categories as follows:

Special candidate votes: This group consists of $t + 3$ copies of the vote,

$$Z_0$$

These votes ensure that every winning committee must include Z_0. By we denote remaining candidates in an arbitrary but fixed order.

"Safe committee" votes: For each candidate x_i we have $\frac{18t^2}{t+1}$ copies of the vote:

$$v_{x_i} := x_i \quad Z_1$$

Independent Set Votes: For every edge $\{u, v\}$ in the graph, we introduce $2t$ copies of following two votes:

$$u \quad v \quad Z_0$$
$$v \quad u \quad Z_0$$

We denote the block of these $4t$ votes by $V_{u,v}$. The intuition for this is to ensure that if some committee has both the endpoints of some edge then the *overall* misrepresentation will be more than .

The votes described above together constitute our profile P. By fixing an ordering on C and respecting it on the unspecified votes, it is straightforward to verify that all pairs of candidates cross at most six times in this profile. We note that the candidates corresponding to the vertices cross at most *six* times because the construction is based on a *three* regular graph. Define $k = t + 2$ and $r = 1$. The ℓ_1-CC -Robustness Radius instance thus constructed is given by $(C, P, k = t + 2, r = 1)$. This completes the construction of the instance. We now make some observations about the nature of the optimal committees which will help us argue the equivalence subsequently.

Possible winning committees. Let T denote the set of candidates corresponding to t-sized independent set in G (whenever it exists). We refer to the subset of candidates given by $\{Z_0, x_1, x_2, \ldots, x_{t+1}\}$ as the *safe committee* and denote it by S.

Lemma 5. *The constructed profile has a unique winning committee if and only if the graph G has no independent set of size t. The safe committee S has a dissatisfaction score of and is always a winning committee. If (G, t) is a Yes instance, then $\{Z_0, Z_1\} \cup T$ is also an optimal committee, where T denotes an independent set of size t in G. Further, any k-sized committee not of this form will have dissatisfaction strictly greater than $+ 1$.*

Proof. It is easy to see that the dummy candidate will not appear in any optimal committee, since it appears in the top positions for exactly one vote.

Let us compute the dissatisfaction score for the two proposed committees. For the *safe committee*, we get zero dissatisfaction from the special candidate votes and safe committee votes, and for each edge we get the $8t$ dissatisfaction which gives us a total dissatisfaction score of $8t \; \frac{3n}{2} = 12nt =$. For the committee based on the independent set, we get zero dissatisfaction from the special candidate votes, $18t^2$ from the *safe committee* votes (one per vote) and $(\frac{3n}{2} - 3t) \; 8t + 3t \; 2t = 12nt - 18t^2$ from the independent set detector votes. Hence, for both the committees the total dissatisfaction is . It is easy to see that this is the best possible dissatisfaction score that can be achieved by any committee of size k.

Note that any optimal winning committee will have candidate Z_0 otherwise, one has to pick $k + 1$ dummy candidates (to remain optimal), which would exceed the committee size. With Z_0 in optimal committee, if we intend to choose only few of x_i^0s then candidate Z_1 is forced in the committee. With these constraints, now, we only have two possible structures for any optimal committee. We will analyze both of them in next part of the proof.

Consider the possible optimal committees which picks Z_0, Z_1, few endpoints of edges which are covered twice and the partial independent set (we interpret this partial independent set as the set of vertices which only has one endpoint with given edge). The edges for which both the endpoints are in the committee gives zero dissatisfaction, edges for which one endpoint lies in the committee gives $2t$ dissatisfaction, and edges for which both the endpoints are not in the committee gives $8t$ dissatisfaction. Hence, the non-uniformity in dissatisfaction clearly indicates that it is better to cover maximum number of edges by picking one end-point rather than completely losing an edge which causes very high dissatisfaction . So, with the remaining budget for t-candidates, the committee with all candidates from independent set will cover maximum edges (to represent by one endpoint) and will cause strictly less dissatisfaction from any other committee by at least $2t$ points.

We now consider a possible winning committee which contains $\{Z_0, Z_1\}$, partial independent set and x_i^0s for the remaining budget of the committee size. Let's compute the dissatisfaction for this committee. Say we pick p candidates among the x_i^0s and $(k - 2 - p) = (t - p)$ candidates from the independent set. The dissatisfaction is:

$$(t + 1 - p) \; \frac{18t^2}{t+1} + \; \frac{3n}{2} - 3(t - p) \; 8t + (3 \; (t - p) \; 2t)$$

which simplifies to: $\quad + (t - p)\left(\frac{18t^2}{t+1} - 18t\right) + \frac{18t^2}{t+1}$.

For any value of t, it is straightforward to verify that the above expression has value strictly greater than $\;+1$. Hence, committees with this structure will also not be optimal. This completes the proof for the lemma. □

Now, we turn to the equivalence of the two instances.

Forward direction. We need to show that the existence of t-sized independent set in the graph implies the existence of one swap of adjacent candidates which changes the set of winning committees for the new election instance (i.e. the robustness radius is one for the constructed election instance). From Lemma 5, we know that whenever there exist a t-sized independent set T, we have two winning committees. In this election instance consider the swap of Z_1 with a dummy candidate on right in any of the *safe committee votes*. Now the score for $\{Z_0, Z_1\} \cup T$ is $\;+ 1$ and it's not optimal anymore. Hence, we have changed the set of winning committees. This completes the argument for forward direction.

Reverse direction. From Lemma 5, we know that unless independent set exists, any k-candidate committee other than the *safe committee* has dissatisfaction score strictly greater than $\;+ 1$. This entails that there does not exist any swap which can introduce a new committee in winning committee set (since a single swap can change the score of any committee by at most one) or can knock off *safe committee* from the set. Hence, in this case robustness radius equal to one forces the existence of required independent set (since this is the only committee that can change the set of winning committees).

This concludes the proof for Theorem 9. □

To show the hardness of calculating Robustness Radius for ℓ_1-CC for 4-crossing profile, we reduce from LSAT. We next recall the definition of LSAT. In an LSAT instance, each clause has at most three literals, and further the literals of the formula can be sorted such that every clause corresponds to at most three consecutive literals in the sorted list, and each clause shares at most one of its literals with another clause, in which case this literal is extreme in both clauses. The hardness of LSAT was shown in (ABC+15). For ease of description, we will assume in the following reduction that every clause has exactly three literals, although it is easy to see that the reduction can be extended to account for smaller clauses as well.

Theorem 10. *Computing the Robustness Radius for ℓ_1-CC with respect to the Borda misrepresentation score is* NP*-hard even when the domain is four-crossing domain.*

Proof. Let be an instance of LSAT with variables $x_1, \ldots, x_n$ and clauses $C_1, \ldots C_m$. Towards constructing the election instance, we introduce one candidate for every literal in . Let p_i and q_i denote the candidates corresponding to the variable x_i. We also introduce n *safe* candidates $Z = Z_1, Z_2, \ldots, Z_n$ and $(2n + 3)$ dummy candidates for each variable (which is a total of $n(2n + 3)$ dummy candidates). Let $d[i, j]$ denote the j^{th} dummy candidate corresponding to the variable x_i. We use V to denote the $2n$

candidates corresponding to the literals, Z to denote two *safe* candidates and D to denote the set of dummy candidates. Hence, the candidate set C for election is,

$$C = V \cup Z \cup D$$

Let c denote the size of the candidate set C. Let us fix the ordering $\succ$ over the set of candidates as follows. The first n candidates are from *safe* committee followed by $2n$ candidates corresponding to LSAT ordering(ℓ). The last $n(2n+3)$ candidates are from D and are arranged in an arbitrary but fixed order. For a subset of candidates X, the notation $\overline{X}$ refers to an ordering of X according to $\succ$. For $1 \leqslant i \leqslant m-1$, let G_i denote literals in the set $C_i \setminus C_{i+1}$, while we let G_m denote the literals in C_m. For each clause C_i we introduce a vote v_i, which has the safe committee candidate Z_1 on the top position, followed by literals of the clause C_i in next three positions, followed by Z_2, and the remaining candidates in the following order:

$$v_i := Z_1 \succ \overline{G_i} \succ Z_2 \succ \overline{G_{i+1}} \succ \cdots \succ \overline{G_m} \succ \overline{G_{i-1}} \succ \cdots \succ \overline{G_1} \succ \overline{Z \setminus \{Z_1, Z_2\}} \succ \overline{D}$$

It is useful to note that the vote v_{i+1} can be thought of as a ranking obtained from the vote v_i by "pushing back" the tuple $\overline{G_i}$ to just behind $\overline{G_m}$ and pulling up $\overline{G_{i+1}}$ over Z_2. Therefore, the ordering among the G_i's in v_m is reverse of their ordering in v_1. Observe that if a literal occurs in $C_i \cap C_{i+1}$, then it appears among the top three positions of both v_i and v_{i+1}.

We now move to the second part of our profile, which consists of votes corresponding to the variables. Here, for a subset of candidates X, we will use $\overline{\overline{X}}$ to refer to an ordering of X according to v_m (say $\succ'$).

Now, for every variable x_i for $1 \leqslant i \leqslant n-2$, we first introduce the following $(n+1)$ votes, with $1 \leqslant j \leqslant (n+1)$.

$$v_{i,j} := d[i,j] \succ p_i \succ q_i \succ Z_2 \succ Z_1 \succ \overline{\overline{(V \setminus \{p_i, q_i\})}} \succ \overline{Z \setminus \{Z_1, Z_2\}} \succ \overline{D \setminus \{d[i,j]\}}$$

Next, for $n+2 \leqslant j \leqslant 2n+2$, we introduce the following vote:

$$v_{i,j} := d[i,j] \succ p_i \succ q_i \succ Z_{i+2} \succ Z_2 \succ Z_1 \succ \overline{\overline{(V \setminus \{p_i, q_i\})}} \succ \overline{Z \setminus \{Z_1, Z_2, Z_{i+2}\}} \succ \overline{D \setminus \{d[i,j]\}}$$

Finally, we introduce one vote with ordering:

$$v_{i,j} := d[i,j] \succ Z_2 \succ p_i \succ q_i \succ Z_1 \succ \overline{\overline{(V \setminus \{p_i, q_i\})}} \succ \overline{Z \setminus \{Z_1, Z_2\}} \succ \overline{D \setminus \{d[i,j]\}}$$

Notice that for $n-1 \leqslant i \leqslant n$, we do not have fresh Z_{i+2} (since we only have n *safe committee candidates*) we skip $n+1$ votes for $n+2 \leqslant j \leqslant 2n+2$. We denote the set of these votes corresponding to variable x_i by V_{x_i}. We fix the committee size $k = n$.

This completes the construction of our profile. It is easy to see that the construction can be completed in polynomial time. In the next part of the proof, we will argue the equivalence of two instances.

Forward direction. In the forward direction, we need to show that, given a valid assignment for LSAT there exist a swap of adjacent candidates in some vote such that the set of winning committees for the new election instance is different from the old set.

We assume LSAT instance to be trivial if it is satisfied by setting all the variable to one (we only consider non-trivial instances). Consider the *safe committee* Z, the misrepresentation score for the safe committee is three. Let   the satisfying assignment. Consider the committee of size n formed by the candidates corresponding to the literals which are set to one in   (we abuse the notation and use   to represent this committee). Given a non-trivial instance notice that at least one q_i is present in our committee. Hence, the misrepresentation score of this committee is also three.

Claim 1. *Every committee except for Z,   has dissatisfaction score of at least five or dissatisfaction of four with strictly more than one voter having representative after the fourth position.*

Proof. We'll consider all possible committee structures. Consider the committee $C = V' \cup D'$ where V' V and D' D such that $|V'| + |D'| = n$. Consider the block where V_{x_i} such that $p_i, q_i \notin V'$. Since there are at least $n + 1$ votes in this block with distinct dummy candidate in top five positions in each vote, there exist a vote which has representative C with misrepresentation at least five. Next, consider a committee $C = D' \cup Z'$. If $Z_2 \notin Z'$ then in each block V_{x_i} there exist at least one vote for which the representative is after fourth position. Similarly, if $Z_1 \notin Z'$ then for all *clause votes* have representative on fifth position. For any other $Z_i \notin Z'$ there exist at least two votes amongst $n + 1$ votes in V_{x_i} with Z_i in top four position with representative after fourth position (since there are $n + 1$ such votes). Finally, consider the case when $C = V' \cup D'$ or $C = V' \cup D' \cup Z'$ there exist a block V_{x_i} for which neither p_i, q_i or z_i is in the committee. Note that a similar argument as in the previous case ($C = V' \cup D'$) works with by considering the block V_{x_i}.

Note that any committee consisting of *only* dummy candidates, has dissatisfaction more than five since dummy candidate does not appear in top six positions in votes corresponding to the clauses.

This completes the proof of the claim. □

Given Claim 1, it is easy to see that committees , Z are winners. Consider the swap $q_i \leftrightarrow Z_1$ for i such that $x_i = 0$ in . The misrepresentation score for committee   in the new instance is four whilst the misrepresentation score for Z is three. Hence, we change the set of winning committees.

Reverse direction. Given $RR = 1$ we need show that   is a *YES* instance. Given Claim 1, the only way to change the set of winning committees is by removing committee   by performing a swap described in the forward direction. Hence, this gives the existence of a valid assignment for LSAT instance whenever $RR = 1$.

Analysis for crossings. We claim that the profile is *4-crossing* with respect to following the order of votes,

$$v_1, v_2, \ldots, v_m, v_{1,1}, \ldots, v_{1,2n+3}, \ldots, v_{i,1}, \ldots, v_{i,2n+3}, \ldots, v_{n,1}, \ldots v_{n,n+2}$$

Clearly, there are at most two crossings for any dummy candidate with any other candidate (due to the vote in which it appears in the top four positions and the fact that we use unique dummy candidate in each vote). For *safe candidates* there is only one crossing between Z_1 and Z_2 on the boundary of a clause and variable votes, and there are exactly two crossings between any Z_i and $\{Z_1, Z_2\}$ in V_{x_i}. For $\{Z_1, Z_2\}$ and any other candidate corresponding to $2n$ literals, there are exactly four crossings (two in clause votes when literal comes to top three positions and go back, and another two crossings corresponding to its V_{x_i} block). For any other Z_i and candidate from V, there are exactly two crossings in votes V_{x_i} (i.e. the latter part of the voting profile). At last, we will analyze the crossings in-between the candidates corresponding to the literals. These candidates will have one crossing in clause votes and two crossings in the variable votes which gives a total of three crossings.

This analysis completes the proof for the Theorem 10. □

4.5.2 Domains Close to Single-Peaked Domains

Theorem 11. *Computing the Robustness Radius for ℓ_1-CC with respect to the Borda misrepresentation score is* NP*-hard even when the domain is four-composite single-peaked domain.*

Proof. We note that our construction is similar to that in Theorem 10 with a few changes.

Construction. Let be an instance of LSAT with variables $x_1, \ldots, x_n$ and clauses $C_1, \ldots, C_m$. We introduce one candidate for every literal in . Let p_i and q_i denote the candidates corresponding to the variable x_i. P and Q denote the candidates corresponding to the positive and the negated literals respectively. We also introduce *n safe* candidates $Z = \{Z_1, Z_2, \ldots, Z_n\}$, and $(n+2)$ dummy candidates for each variable (which is a total of $n(n + 2)$ dummy candidates). Let $d[i, j]$ denote the j^{th} dummy candidate corresponding to the variable x_i. We use V to denote the $2n$ candidates corresponding to the literals, Z to denote two *safe* candidates and D to denote the set of dummy candidates. Hence, the candidate set C for election is,

$$C = P \cup Q \cup Z \cup D$$

Let us fix the ordering over the set of candidates as follows. The first n candidates are from *safe* committee followed by $2n$ candidates corresponding to LSAT ordering(ℓ). The last $n(n + 1)$ candidates are from D and are arranged in an arbitrary but fixed order. Let ^{0} be the reverse of . We denote by $_p$ the LSAT ordering over the restricted set of agents corresponding positive literals. We define $_q$ in the similar way. For a subset of candidates X, the notation $\overline{X}$ refers to an ordering of X according to . For a subset of candidates X V, who occupy adjacent positions in the LSAT ordering projected over V, the notation $V \setminus \overleftarrow{X}$ refers to an ordering according to of the candidates from

$V \setminus X$ who appear after X in the LSAT ordering and similarly $V \setminus X$ refers to an ordering according to $\succ'$ of the candidates from $V \setminus X$ who appear before X in the LSAT ordering. This notation easily yields an ordering which is single-peaked — $\overline{X} \succ \overrightarrow{V \setminus X} \succ \overleftarrow{V \setminus X}$. We use $\overline{X} \succ \overrightarrow{C \setminus X}$ for the shorthand of $\overline{X} \succ \overrightarrow{V \setminus X} \succ \overleftarrow{V \setminus X}$.

We would now like to setup the votes in such a way that robustness radius one implies an existence of valid satisfying assignment. We introduce one vote for every clause as follows. Suppose clause C_i consists of literals (l_i, l_j, l_k), and $l_i < l_j < l_k$ be the LSAT ordering, then we introduce the vote:

$$v_i := Z_1 \succ l_j \succ l_i \succ l_k \succ Z_2 \succ \overrightarrow{Z \setminus \{Z_1, Z_2\}} \succ \overrightarrow{(V \setminus \{l_i, l_j, l_k\})} \succ \overline{D}$$

For every variable x_i, we introduce the following $(n+2)$ votes (we denote the set of these votes by V_{x_i}), with $1 \leq j \leq (n+1)$.

$$v_{i,j} := d[i,j] \succ p_i \succ q_i \succ Z_i \succ \overrightarrow{Z \setminus \{Z_i\}} \succ \overrightarrow{P \setminus p_i} \succ \overrightarrow{Q \setminus q_i} \succ \overrightarrow{D \setminus \{d[i,j]\}}$$

We then introduce the following vote as a last vote in block V_{x_i}:

$$v_{i,j} := d[i,j] \succ Z_i \succ p_i \succ q_i \succ \overrightarrow{Z \setminus \{Z_i\}} \succ \overrightarrow{P \setminus p_i} \succ \overrightarrow{Q \setminus q_i} \succ \overrightarrow{D \setminus \{d[i,j]\}}$$

This completes the construction of our profile. We set the committee size is set to n. We now turn to the argument for equivalence.

Since the construction of profile is similar to Theorem 10, it is easy to show claim analogous to 1 in the previous proof. This establishes Z, τ as winning committees given τ is a satisfying assignment for LSAT. Note that we again consider only the non-trivial instances of LSAT where setting all variable to one is not a satisfying assignment.

Given this, the argument for forward and reverse direction hold in a similar way as in Theorem 10.

Analysis for 4-composite SP. We claim that the profile is SP with respect to the partitions Z, P, Q, D of the candidate set. For the candidates from Z, they are single-peaked with peak Z_1 in clause votes and with peak Z_i in variable votes. It is easy to see that candidates from D are SP with respect to the canonical ordering of dummy candidates since for clause votes these candidates appear in the canonical ordering at the end and for variable votes, D is single-peaked with candidate $d[i,j]$ as the peak.

From the construction, the candidates from V form a SP profile in clause votes since they follow the LSAT ordering (and hence P and Q are SP too). In the variable votes, P and Q form a SP profile with candidates p_i and q_i as their respective peaks. Hence, the overall profile is *4-composite single-peaked.* □

4.6 Conclusion and Open Problems

We demonstrated XP algorithms for the Robustness Radius problem, when parameterized by the size of the committee, for both the ℓ_1-CC and the Approval-CC voting rules, using a greedy approach. This complements the known W[1]-hardness of the problem with respect to this parameter. We also explicitly establish the W[2]-hardness of Robustness Radius for the Approval-CC voting rule when parameterized by the size of the committee, even when every voter approves at most three candidates, and when the robustness radius is one. We also established that Robustness Radius for the ℓ_1-CC and ℓ_1-CC voting rules remains intractable on fairly structured preferences, such as six-crossing profiles and four-composite single-peaked profiles.

A natural direction for further thought is if our XP algorithm can be improved to a better running time, especially on structured profiles such as single-peaked or single-crossing domains. A tempting approach is to see if we can exploit the fact that optimal Chamberlin-Courant committees can be computed in polynomial time on these domains. One immediate challenge is the following: if we require our swaps to be such that the resulting profile also remains in the domain that we are working on, then the case when the input profile has multiple winning committees is harder to decide: we can no longer push a committee out of the winning set with one swap, because the said swap may disturb the structure of the profile. We also believe that instead of guessing all possible choices for a nearly winning committee B, on structured profiles one might be able to cleverly anticipate the right choice of B without trying all of them. Another natural parameter is the *Robustness Radius* (r), and it would be a non-trivial to obtain FPT with respect r.

Part II

Matching

Chapter 5

Stable Matching on Restricted Domains

5.1 Introduction

The theory of two-sided matching has been a prevalent research area for the past couple of decades. A driving force for such extensive research is the wide-spread applicability to real life scenarios such as college admissions (TST01), resident-hospitals matching (Rot84), and kidney exchange (Irv07).

In the well known *stable marriage* (SM) problem, we are given agents partitioned into two equal-sized subsets which are typically referred to as men and women, and the men (women) provide their preferences over women (men). Our goal is to match the agents to each other while avoiding *blocking pairs*, also known as *stable matchings*. Given a matching M, a blocking pair is a pair of agents (a, b) who are not matched by M, but who prefer each other over their matched partners with respect to M. In this setting, when the preferences of agents are complete orders, a stable matching involving *all* agents always exists and can be found efficiently (GS62). This polynomial time *deferred acceptance* matching algorithm by Gale-Shapley was a seminal result in the theory of stable matching. A general version of this bipartite matching problem is matching on general graphs (the problem is popularly known as the *stable roommates*). In the *stable roommates* problem, we a are given single set of agents who express preferences over each other and the aim to find a stable matching. In contrast to the bipartite case, here, the stable matching might not always exist. It is efficient to determine when they do if the preferences are strict and complete linear orders (Irv85).

We now return to our passing remark regarding preferences above. While the choice of complete and strict rankings is a natural one, in many natural application scenarios, one would want to relax this to allow for agents to express notions of indifference and absolute non-suitability. Both of these generalizations are, indeed, well-studied: in the stable roommates problems with ties, we allow for agents to indicate indifference between agents in the ranked order; while in the variant of the problem with incomplete lists, we allow for the agents to declare some agents as unacceptable altogether by not featuring them on their preference lists at all. It is convenient to think of the problem of stable roommates where preferences may have both incomplete lists and ties as a graph where the vertices rank the edges incident to them, in a manner that is not necessarily strict. Both *unacceptability* and *indi erence* are defined analogously for the *stable marriage* case (Irv94), (IMMM99).

Once we allow for ties and incomplete lists, the complexity of finding stable matchings, both in the setting of marriages and roommates, changes dramatically. We summarize the highlights here. If preferences are given as incomplete lists, then both problems (SRI – Stable Roommates with Incomplete Preferences and SMI – Stable Marriage with Incomplete Preferences) remain tractable. Note, however, that for the stable marriage problem, there are instances with incomplete lists which do not admit perfect matchings any more - a particularly extreme example would be when the list for every agent is empty. When a non-trivial stable matching exists, then any stable matching matches the same set of agents for both problems. In the setting of ties, note that the notion of stability needs to be clarified further. We will work with "weak" stability, wherein (a, b) forms a blocking pair with respect to a matching M if a and b *strictly* prefer each other over their current matched partner in M. In this setting, although the stable marriage problem (SMT) remains tractable and all stable matchings match the same set of agents as before; the stable roommates problem (SRT) is NP-hard. Finally, consider the case when both ties and incomplete lists are permitted in the preferences. Now, in the setting of stable marriages (SMTI), stable matchings may have different sizes, and the problem of finding the stable matching of the largest size turns out to be NP-hard. In the setting of roommates (SRTI), checking if a stable matching exists at all is NP-hard. We refer the reader to (IM08), (Che19) for a detailed survey on the complexity of variants of stable matching and stable roommates problem.

The intractability of many matching problems when preferences have ties and/or are incomplete has motivated several research directions in the literature. One line of work considers scenarios involving *structurally restricted preferences* of agents. We refer to the work of Bredereck et al (BCFN19) for examples of real-world scenarios where agent preferences may naturally be "single-peaked" or "single-crossing". Examples of results that are known in this setting include the NP-hardness of checking the existence of a stable matching in the setting of SRTI even when profiles are narcissistic, single-peaked, and single-crossing (BCFN19); and that the stable roommates problem for complete and strict rankings (SRT) always admits a (weakly) stable matching that can be found in (linear) sublinear time on profiles that are narcissistic and single-peaked (BIT86) or narcissistic and single-crossing (RBN17).

Another example of a study on structured preferences is the work of Abraham et al (ALMO07) who demonstrate that weakly stable matchings that match all agents are guaranteed to exist for the roommates problem in the setting of ties and incomplete lists if the preferences are derived from "globally ranked pairs". We refer the reader to the next section for formal definitions of the notions of structured preferences alluded to here.

Algorithmic work on these problems has largely focused on approximating the optimization questions, such as those that ask for the maximum-sized stable matchings, for instance, as is the case for the problem of SMTI. Also, the complexity of problems concerning preferences with ties has been studied from a parameterized perspective as well. In this setting, there are three natural parameters: the total number of indifference classes across all agents ($_1$), the size of the largest "indifference class" ($_2$), and the total size of all the indifference classes in the instance ($_3$). The question here is if the problem becomes tractable if any of these quantities are small. For instance, in the SMTI setting, for the problem of finding a largest stable matching, the problem is W[1]-hard when parameterized by $_1$ (MS10), NP-hard when $_2$ is a constant and even when lists have constant size (CIM16), and fixed-parameter tractable with respect to $_3$ (MS10).

A phenomenon ubiquitously observed in matching markets is the *strategic behaviour* of participating agents. The motivation for *manipulation* to obtain a better (or the best) stable matching partner comes due to the inherent bias in the matching mechanisms such as Gale-Shapley 12. The interesting issue here is whether agents have an incentive for misrepresenting their preferences i.e. can they manipulate using such preferences? Unfortunately, Roth (Rot82) has proved that *all* stable marriage mechanisms can be manipulated. This result is analog of the classical Gibbard Satterthwaite theorem in voting (G[+]73), which states that any non-dictatorial rule under the modest assumption is manipulable. Commonly studied form of manipulation is the *truncation manipulation* ((GS85), (RR99), (CS14)) where the agents are allowed to present the preferences over a subset of agents instead of complete lists. *Truncation* is a powerful model of manipulation, which can encode any possible manipulation. Another direction which we consider is the *permutation manipulation*, where only the permutations of complete lists are allowed in the manipulated profiles. Efficient algorithms have been shown for the computation of *optimal permutation* manipulation for individual agents and coalitions ((TST01), (KM10), (VG17), (DST18)).

Recently, there has been an interest in studying various notions of Robustness in stable marriage ((GSOS17), (ML18), (CSS19)). One interpretation of *Robust* solutions to the stable marriage problem was recently presented by Genc et. al. (GSOS17). The authors defines *(a, b)-Supermatch* as a matching M such that to change the partners of *any* a-agents according to M requires changing the partners for at most b other agents. In the follow-up paper (GSSO19), they showed that checking the existence of even $(1, 1)$ - supermatch is NP-complete.

5.1.1 Our contributions and Organization of the Chapter

We are now ready to describe our contributions. First, we address a question from Bredereck et al. (RBN17) of finding a stable matching in the setting of SRTI parameterized by the "degree of incompleteness" of the input instance on structured domains in Section 5.3. We borrow the parameters suggested in the work of Marx and Schlotter (MS10) and note that the W[1]-hardness persists even for domains that are both single-peaked and single-crossing. We suggest a new reduction which demonstrates that the problem is NP-hard when $_2$ is a constant and even when lists have constant size *and* the preferences are single-peaked and single-crossing, strengthening the result in Cseh et al. (CIM16), although we believe that our approach is conceptually simpler.

Next, we consider the questions of finding Sex-Equal stable matchings (SESM) and Egalitarian stable matchings in the setting of the stable marriage problem. These are stability notions with additional desirable properties — both problems are known to be intractable in several situations. We extend these known hardness results to restricted domains. For SESM, the NP-hardness on single-peaked and single-crossing profiles follows from an analysis of the NP-hardness reduction that is already shown by McDermid and Irving (MI14), where all preference lists have at most three candidates and no ties. For the question of finding an Egalitarian stable matching, we show that the problem is W[1]-hard when parameterized by $_1$ even when the input profile is single-peaked and single-crossing. This result follows by a careful adaptation of construction given by Marx and Schlotter (MS10) for the related question of finding a maximum matching in the setting of SMTI. We describe these results in Sections 5.4 and 5.5, respectively.

We also consider the problem of manipulating a stable matching while staying within a domain. In a recent result, Vaish and Garg (VG17) show that an optimal permutation manipulation of the Gale-Shapley mechanism (with men proposing) can be obtained by only changing the position of exactly one agent in the preferences of a manipulating woman. We describe examples showing that this is no longer the case if we insist that the manipulated profile respect the structure of the original profile. On a different note, it also follows from the work of Vaish and Garg (VG17) that profiles which admit a unique stable matching are not vulnerable to manipulation, even when we do not insist for the matching output by Gale-Shapley on the manipulated profile to be stable with respect to the truthful profile. This motivates us to expand our understanding of profiles that have unique stable matchings. We show that preferences that arise from the Euclidean domain or from globally ranked pairs have unique stable matchings and are therefore not vulnerable to manipulation. Although this is implied by a more general result regarding profiles with unique stable matchings, we observe this explicitly so as to be able to highlight the following contrast: we give examples of profiles that are "close to" these structured profiles but admit exponentially many matchings. We initiate a detailed discussion in Section 5.7.

In section 5.8, we introduce the problem of finding a matching that matches a given subset of agents, which we call a critical set. Problems along these lines have already received some attention in the literature — for instance, the question of finding a stable matching with forbidden pairs or the question of finding a stable matching that extends a partial matching have been studied. Our motivation for introducing this variant is to model application scenarios where matching some agents may be more crucial than others, for example, for reasons of time-sensitivity in a kidney-exchange market. In such settings, the only stable matchings that match a specified subset may not be the largest stable matchings, so this is a possibly tangential objective that we believe is worth studying explicitly. We note that this problem is trivial in all settings where a stable matching that matches all agents is known to exist and can be found efficiently. On the other hand, we show that the question is W[1]-hard in the context of SRTI when parameterized by the size of the critical set on profiles that are single-peaked and single-crossing, even when $_2$ is a constant. It is also W[1]-hard parameterized by $_1$ in the context of SMTI on profiles that are single-peaked and single-crossing, even when the critical set has one agent. The first result is obtained by a reduction from a variant of the Independent Set problem, while the second result can be obtained as a direct consequence of a reduction by Marx and Schlotter (MS10).

Finally, for the *(a, b)-Supermatch* problem, we show an efficient algorithm to settle the special case of $(n, 0)$ - supermatch. Our algorithm is essentially a careful run of Gale-Shapley algorithm to check if the given instance admits a $(n, 0)$ - supermatch, and return such matching if there exists one. We further show the hardness for a problem nearby to the *(a,b)-Supermatch* which asks for an existence of re-orientation of a particular sized set of agents given an instance and a stable matching.

5.2 Preliminaries and Background in Matching

In this section, we will provide the definitions that will be relevant to the discussions throughout this chapter. We refer the reader to Section 2.3, to recall the definition of concepts in parameterized complexity.

We write $\binom{[n]}{2}$ for set $\{(i,j) | 1 \leq i \leq j \leq n\}$.

5.2.1 Problem Setup

We will start with defining a stable roommates problem and bipartite stable matching problem (we will refer this as a stable marriage problem) with ties and incomplete lists.

Stable Roommates Problem:

An instance $< V, P >$ of *stable roommates* consists of set Let $V = \{1, 2, \ldots, l\}$ be a set of $\ell = 2n$ agents. Each agent $i \in V$ has a *preference order* $\succeq_i$ over a subset $V_i \subseteq V$ of agents that i finds acceptable as a partner. The set V_i is called the *acceptable set* for the agent i and a preference order $\succeq_i$ over V is a weak order over V_i, that is, a transitive and complete binary relation on V_i.

Stable Marriage Problem:

The stable marriage problem is defined similarly, except that the set of agents is $(M \cup W)$, where $|M| = |W| = n$, and set of acceptable agents for each member of M is a subset of W, and similarly, the set of acceptable agents for each member of W is a subset of M.

In our model, we assume that for any pair of agents (i, j), i is not acceptable to j if and only if j is not acceptable to i (note that this restricts our model since with the stability notions under consideration this will be true without loss of generality). We recall that P is a preference profile, which is a collection of preferences of all participating agents in the instance.

Definition 12. *Matching: Given a preference profile P for a set V of agents, a matching M is a subset of disjoint pairs of agents $\{x, y\}$ with $x \neq y$ who find each other acceptable.*

For a pair $\{x, y\}$ of agents, if $\{x, y\} \in M$, then by $M(x)$ we denote the corresponding partner y and $M(y) = x$; otherwise we call this pair unmatched. We write $M(x) = \emptyset$ if agent x has no partner, that is, if agent x is not involved in any pair in M.
For the stable marriage case, as the acceptable partners $\forall m \in M$ is a subset $w' \subseteq W$, the matching M, in this case, will be a set of (m, w) pairs. Notice that although we use M to denote both the stable matching and the set of men in stable marriage instance, the usage will be clear from the context (more often than not we will precede the word 'matching' before M to denote a stable matching).

Definition 13. *Blocking Pair: An unmatched pair $\{x, y\} \notin M$ is* blocking M *if the pair "strictly prefers" (i.e. $y \succ_x M(x) \& x \succ_y M(y)$) to be matched to each other.*

Example 1.
$$m_1 := (w_1 \sim w_2) \qquad w_1 := m_1 \succ m_2$$
$$m_2 := w_1 \succ w_2 \qquad w_2 := m_2 \succ m_1$$

The matching $M = \{(m_1, w_2), (m_2, w_1)\}$ does not admit any blocking pair under our definition. The blocking pair with this definition sometimes known as *strong blocking pair*.

Definition 14. *Stable Matching: A matching M is stable if no unmatched pair is blocking M.*

Note that this stability concept is called *weak stability* when we allow ties in the preferences (1).

We assume, without loss of generality, that for any pair of agents (i, j), i is not acceptable to j if and only if j is not acceptable to i.

5.2.2 Preferences with ties

For an agent i the rank of an acceptable agent j is defined as $\text{rank}(i, j) = k$ if i strictly prefers the agents from precisely $k - 1$ indifference classes to agent j.

A *tie* for an agent $v \in V$ is a subset $T \subseteq V \setminus v$ of maximum cardinality such that $|T| \geq 2$ and $\text{rank}(v, t_1) = \text{rank}(v, t_2) \neq \infty$ for every $t_1, t_2 \in T$. The length or size of a tie is the number of elements in the tie. In the Section 5.1, we referred to ties as indifference classes. Borrowing the terminology and notation from (MS10), we will use the following parameterization functions, for an instance I of SMTI:

- $\tau_1(I)$ denotes the number of ties in I.
- $\tau_2(I)$ denotes the maximum length of a tie in I.
- $\tau_3(I)$ denotes the total length of the ties in I, which is the sum of the length of each tie in the instance. We have that $\tau_3(I) \leq \tau_1(I) \cdot \tau_2(I)$.

We recall the definitions of single-peaked, single-crossing, Euclidean, and narcissistic domains from section 2.1. For profiles which are both single-peaked *and* single-crossing, we use SPSC for brevity. We say that a preference profile P with ties and incomplete preferences is single-peaked or single-crossing if there exists a linear extension of P to P' where all the preferences are complete orders (i.e. complete preference lists without ties) such that P' satisfies the domain properties. We note that this definition of domain restriction was used recently by Bredereck et al. in (RBN17). We note that there is also a notion of "tie-sensitive" single-crossingness, which does not consider linear extensions, but requires that there exists an ordering of the agents for which all the ties "lie in the middle". In the present work, we do not consider the tie-sensitive notion.

5.2.3 Variants of Stable Matching Problems

We define a *score of matching* M as $\sum_{(x,y)\in M} \text{rank}(x,y) + \text{rank}(y,x)$. Note that an agent $z \in V$ such that $z \notin M$ contributes the score equal to the length of its preference list.

An *Egalitarian matching* is matching with a minimum score. We denote the problem of finding an *Egalitarian Matching* by Egal-SMTI or Egal-SRTI for a stable marriage and stable roommate case, respectively.

The problem of finding a maximum sized matching given an instance of SMTI (SRTI) is denoted by Max-SMTI (Max-SRTI).

A *Sex-Equal Stable Matching (SESM)* is defined as the matching which minimizes $\delta = |\sum_{(x,y)\in M} \text{rank}(x,y) - \sum_{(x,y)\in M} \text{rank}(y,x)|$. The SESM problem asks for the existence of a matching with $\delta' \leq \delta$.

The notion of *Stable Roommates with Globally Ranked Pairs* (SR-GRP) implies a restriction on preferences such that the preferences can be derived from a ranking function $f : E(G) \to \mathbb{N}$. An agent x prefers agent y to agent z if $e = (x,y), e' = (x,z)$ and $f(e) < f(e')$, and x is indifferent between them if $f(e) = f(e')$. (We can similarly define (SM-GRP)).

The concept of *(a, b) Supermatch* was first defined by (GSOS17).
(a,b) supermatch: A stable matching M is called an (a,b)-supermatch if for any set $V' \subseteq V$ of a agents decides to break their marriages from matching M, thereby breaking a pairs, it is possible to find another stable marriage M' by changing the assignments of those a agents and at most b others.

Intuitively, we say that it is possible to "repair" matching M considering the requests by `a' pairs by disturbing (changing the partners of) at most b other candidates.
It is clear from the definition that for any valid (a,b) supermatch instance, $a + b \leq n$.

5.2.4 Background and Gale Shapley Algorithm

In the seminal paper (GS62) of Gale-Shapley in 1962, the authors showed that for the stable matching instance with complete preference lists, there always exists a stable matching. The *deferred acceptance algorithm* (or the Gale-Shapley algorithm) given by the authors find one such stable matching with a special property of *male optimality*. This means that the algorithm returns the matching most preferred by the *men* (or more generally the proposing side) among all possible stable matchings in the instance.

The algorithm proceeds in rounds and each round has *two* phases.
(i) The proposal phase, in which each man proposes to his most preferred woman; *(ii) the rejection phase*, where each woman with multiple proposals rejects all proposals except for the best man according to her preferences. The algorithm terminates when all men are matched.

In (ILG87) the authors make *two fundamental observations regarding GS-algorithm:*

Observation 1. *If m proposes to w, then there is no stable matching in which m is matched to better partner than w according to his preferences.*

Observation 2. *If w receives a proposal from m, then there is no stable matching in which w is matched to someone worse than m according to her preferences.*

The output of Gale-Shapley algorithm has following desirable property for the proposing side:

Theorem 12. *The output of the algorithm is simultaneously optimal for all men and pessimal for women from among all stable matchings (DF81).*

Lattice of Stable Matchings: Given a preference profile P, and two stable matchings μ and μ', define the *join* function ($\mu_\vee = \mu \vee \mu'$) as follows: for each $m \in M$ and $w \in W$:

$$\mu_\vee(m) = \begin{cases} \mu(m) & \text{if } \mu(m) \succ_m \mu'(m) \\ \mu'(m) & \text{otherwise} \end{cases}$$

$$\mu_\vee(w) = \begin{cases} \mu(w) & \text{if } \mu(w) \succ_w \mu'(w) \\ \mu'(w) & \text{otherwise} \end{cases}$$

Similarly, we define *meet* function $\mu_\wedge = \mu \wedge \mu'$ for each $m \in M$ and $w \in W$:

$$\mu_\wedge(m) = \begin{cases} \mu'(m) & \text{if } \mu(m) \succ_m \mu'(m) \\ \mu(m) & \text{otherwise} \end{cases}$$

$$\mu_\wedge(w) = \begin{cases} \mu'(w) & \text{if } \mu(w) \succ_w \mu'(w) \\ \mu(w) & \text{otherwise} \end{cases}$$

In essence, the *join* function assigns the *best* partner among the two matchings and *meet* function assigns the worst partner with respect to the preferences of men.

The following result from (Knu97), attributed to John Convay, asserts that the *join* and *meet* operations on any pair of stable matchings gives a stable matching.

Let S the set of all stable matchings in the given instance.

Theorem 13. *For $\mu \in S$ and $\mu' \in S$, the join ($\mu \vee \mu'$) and meet ($\mu \wedge \mu'$) functions returns a stable matching.*

We will use the above Theorem 12, 13 and the concept of lattice in section 5.9.

5.3 Stable Roommates with Short Lists and Few Ties

In this section, we show the hardness for the problem of finding a stable matching in the context of SRTI. This problem was recently shown to be NP-complete by Cseh et al. (CIM16, Section 3), even for constant-length preference lists that have ties of length at most two. Here, we also demonstrate that the problem is NP-complete, albeit using a simpler approach, from which it becomes apparent that the problem remains hard even when the preference profiles are single-peaked and single-crossing (SPSC for brevity), which is not immediate from the construction by (CIM16).

Theorem 14. *In the setting of SRTI, deciding the existence of a stable matching is* NP-*complete even when the preferences have constant length and are SPSC.*

Proof. We reduce from (2/2/4)-SAT, which is the variant of Satisfiability where every clause has four literals and every literal occurs exactly twice — in other words, every variable occurs in exactly two clauses with positive polarity and in exactly two clauses with negative polarity. The question is if there exists an assignment of truth values to the variables under which exactly two literals in every clause evaluate to true. The (2/2/4)-SAT problem is known to be NP-complete (RW86).

Construction. Let be a (2/2/4)-SAT instance over the variables $V = \{x_1, \ldots, x_n\}$ and clauses $C = \{C_1, \ldots, C_m\}$. Note that $m = n$. For every variable x_i, we introduce four new variables: p_i, r_i and q_i, s_i. We replace the two positive occurrences of x_i with p_i and r_i, and the two negated occurrences of x_i with q_i and s_i. We abuse notation and continue to use $\{C_1, \ldots, C_m\}$ to denote the modified clauses. Corresponding to each variable $x_i \in V$ we introduce six agents with preferences:

$$a_i : (p_i \quad q_i) \quad b_i \quad c_i \qquad\qquad d_i : (r_i \quad s_i) \quad e_i \quad f_i$$
$$b_i : c_i \quad a_i \qquad\qquad e_i : f_i \quad d_i$$
$$c_i : a_i \quad b_i \qquad\qquad f_i : d_i \quad e_i$$

Next, we introduce the following two agents corresponding to each clause. For clarity, we demonstrate with an example: let $(C_i = x_v \vee \overline{x_w} \vee x_y \vee x_z)$, then we have:

$$C_i^1 : p_v \quad q_w \quad p_y \quad p_z$$
$$C_i^2 : p_v \quad q_w \quad p_y \quad p_z$$

where p_i, q_i denote one of the positive and negative occurrences, respectively.

Finally, we introduce following four agents corresponding to the four occurrences (two positive and two negative) of each variable:

$$p_i : (C_i^1 \quad C_i^2 \quad C_j^1 \quad C_j^2) \quad s_i \quad a_i$$
$$r_i : (C_i^1 \quad C_i^2 \quad C_j^1 \quad C_j^2) \quad q_i \quad d_i$$
$$q_i : (C_k^1 \quad C_k^2 \quad C_l^1 \quad C_l^2) \quad r_i \quad a_i$$
$$s_i : (C_k^1 \quad C_k^2 \quad C_l^1 \quad C_l^2) \quad s_i \quad d_i$$

where x_i appears positively in C_i, C_j and negatively in C_k, C_l.

This completes the construction, we now turn to an argument for the equivalence of two instances.

Forward direction. Let be the satisfying assignment for . If sets x_i to true, we match pairs $\{a_i, q_i\}, \{d_i, s_i\}$ (if x_i is false we match pairs $\{a_i, p_i\}, \{d_i, r_i\}$). We match p_i, r_i to one of the two copies of the clauses they appear in, we match the pairs $\{b_i, c_i\}, \{e_i, f_i\}$.

The agents a_i, d_i, C_i^j, and p_i, r_i (q_i, s_i for the case when x_i is set to false) will not participate in any blocking pair since they are matched to their top preference. In the remaining set of agents, for each variable, either its two positive or two negative occurrences are present. These cannot form blocking pair in-between them, since they are not on each other's preference list. Hence, the proposed matching is stable.

Reverse direction. Let M be the stable matching for the instance. We use the Lemma 1 from (RBN17) and infer that in any stable matching, a_i must be matched to p_i or q_i and d_i must be matched to r_i or s_i. This implies the pairs $\{b_i, c_i\}, \{e_i, f_i\} \in M$. We now show the following lemma.

Lemma 6. *In any stable matching M, the pair $\{a_i, d_i\}$ is either matched to $\{p_i, r_i\}$ or matched to $\{q_i, s_i\}$ with respective order of agents.*

Proof. Consider the case when $\{a_i, d_i\}$ are matched to $\{p_i, s_i\}$ respectively, under matching M. Now, consider the pair of agents $\{p_i, q_i\}$, these agents strictly prefer each other over a_i, d_i respectively, hence, the pair is a blocking pair for M. This gives a contradiction. A similar contradiction can be obtained in the other case when $\{a_i, d_i\}$ are matched to $\{r_i, q_i\}$. □

From Lemma 6, it is clear that a_i and d_i are either matched to the two positive occurrences or two negative occurrences of variable x_i for all $i \in [n]$. Observe that for the remaining agents ($2n$ clause agents and $2n$ variable agents), they are top preferences of each other, and hence, will get matched to each other in any stable matching.

We now recover the assignment of variables from this matching. If agents $\{a_i, d_i\}$ are matched to positive copies of the variables x_i then we set the variable to *false* and vice versa. We claim that this assignment satisfies each clause exactly twice. As, two agents corresponding to two copies of all clauses are matched under M, every clause is satisfied by at least two agents. Notice that among the four agents corresponding to a variable, exactly two will be matched to two clause agents. This implies that the number of variables satisfying each clause is at most two (using a counting argument for the number of clauses and remaining variable agents). This concludes the proof of equivalence.

Note that in the above construction the length of a preference list is bounded by six, and the maximum length of ties is four. Hence, the reduction implies that the problem is NP-complete even for constant values of $_2$. This eliminates the possibility of even XP algorithm with respect to $_2$.

To see that the profile is SPSC, consider the axis:

$$: [C_i^j] \quad [P_i] \quad [Q_i] \quad [R_i] \quad [S_i] \quad [A_i] \quad [B_i] \quad [C_i] \quad [D_i] \quad [E_i] \quad [F_i]$$

for $i \in [n]$ and $j \in \{1, 2\}$ where $[P_i]$ denotes set of p_i for $i \in [n]$, and similarly we define all the following sets after $[P_i$. Also, the notation $[S]$ for any set of agents S, denotes an arbitrary fixed ordering of these agents. Note that each of the individual things in denotes the ordered set of all agents of that type. It can be easily checked that the profile is single-peaked and single-crossing with respect to when we resolve the ties in favor of the fixed arbitrary ordering for each set. □

5.4 Sex-Equal Stable Matchings

In this section, we observe the following hardness result.

Theorem 15. *Computing SESM with $\Delta = 0$ is* NP*-complete even for SPSC profiles comprising of incomplete preferences of constant length and no ties.*

Proof. We reduce from Clique. The claim follows by studying the construction obtained in (MI14) (Section 5, Theorem 5.8). For completeness, we will now describe the construction.

Step 1: (The vertex gadget) In this step for each $v_i \in V$ we create $4|E| + 1$ (man, woman) pairs. Each of these men will have either two or three entries in their preference list, while each woman will have exactly three entries. For this step, we'll only define first two entries for men and the second and third entries for women. Third preference of man m_i^j and first preference of women w_i^j will be defined later. The preference lists of m_i^j, w_i^j for $j \in [0, 4|E|]$ looks as follows:

$$
\begin{array}{llllllll}
m_i^0 : w_i^0 & \succ & w_i^1 & \succ & w_i^0 : \_ & \succ m_i^{4|E|} & \succ & m_i^0 \\
m_i^1 : w_i^1 & \succ & w_i^2 & \succ & w_i^1 : \_ & \succ m_i^0 & \succ & m_i^1 \\
m_i^2 : w_i^2 & \succ & w_i^3 & \succ & w_i^i : \_ & \succ m_i^{i-1} & \succ & m_i^i \\
m_i^{4|E|} : w_i^{4|E|} & \succ & w_i^0 & \succ & w_i^{4|E|} : \_ & \succ m_i^{4|E|-1} & \succ & m_i^{4|E|}
\end{array}
$$

Step 2: (The edge gadget) For each edge $(v_r, v_s) \in E$, we introduce two men and two women. Each of the two men and women have two candidates in their preference lists. The preferences of these agents are shown below, again, the entries with asterisk will be defined later.

$$
\begin{array}{llll}
m_{r,s}^1 : w_{r,s}^1 \succ & & w_{r,s}^1 : \_ \succ & m_{r,s}^1 \\
m_{r,s}^2 : w_{r,s}^2 \succ & & w_{r,s}^2 : \_ \succ & m_{r,s}^2
\end{array}
$$

Step 3: (Complete the preference lists) For each edge, $(v_r, v_s) \in E$, with $r < s$ (according to arbitrary defined order over the vertices), we choose two men created in step 1, in correspondence to v_r and v_s by selecting the first man m_r^p (m_s^q) respectively from the sorted list $m_r^0, m_r^1, \ldots, m_r^{4|E|}$ (respectively, $m_s^0, m_s^1, \ldots, m_s^{4|E|}$ whose third choice has not yet been specified. In this step we complete the preferences for $m_r^p, m_s^q, w_r^{p+1}, w_s^{q+1}, m_{r,s}^1, m_{r,s}^2, w_{r,s}^1, w_{r,s}^2$ as described below,

$$
\begin{array}{llll}
m_r^p : w_r^p \succ w_r^{p+1} \succ \underline{w_{r,s}^1} & & w_r^{p+1} : \underline{m_{r,s}^2} \succ m_r^p \succ m_r^{p+1} \\
m_s^q : w_s^q \succ w_s^{q+1} \succ \underline{w_{r,s}^2} & & w_s^{q+1} : \underline{m_{r,s}^1} \succ m_s^q \succ m_s^{q+1} \\
m_{r,s}^1 : w_{r,s}^1 \succ \underline{w_s^{q+1}} & & w_{r,s}^1 : \underline{m_r^p} \succ m_{r,s}^1 \\
m_{r,s}^2 : w_{r,s}^2 \succ \underline{w_r^{p+1}} & & w_{r,s}^2 : \underline{m_s^q} \succ m_{r,s}^2
\end{array}
$$

At this stage, we have completed the preference lists for all the men and women created in step 2 and all men in step 1 (note that some men in step 1 have preference list of length two). However, there is still a set of women w_i^j created in step 1 for which we have not defined their top preference. For each of these women, we create a pair of a dummy man and a dummy woman such that the dummy pair place each other at their top preference, and we place w_i^j on the dummy man's preference and define the dummy man as the top preference for w_i^j. This way, in every stable matching the dummy man and woman are paired with each other.

Step 4: (Pad the instance) In this step, we will pad the instance with more agents to obtain $(\ \ _M) = 0$ i.e. the score of *men optimal* stable matching is zero. Let $t = 8|V||E| + 2|V| + 2|E| - [K(8|E| + 2) + 8\binom{K}{2}]$. To offset the score by one, we introduce two men-women pairs with the following set of preferences:

$$x_0^i : y_1^i \quad y_0^i \qquad\qquad y_0^i : x_0^i$$
$$x_1^i : y_1^i \qquad\qquad y_1^i : x_1^i \quad x_0^i$$

Proof of equivalence: The proof of equivalence for our reduction works similarly as shown in (MI14) and we refer the reader to section 5 of that paper for the further details.

Analysing the preference pro le: Let $M_V(W_V)$ denote the set of all men (women) $m_i^j(w_i^j)$ introduced corresponding to the vertices in the graph. We define the linear ordering over these sets as follows:

$$[M_V] : m_1^0 \quad m_1^1 \qquad m_1^{4|E|} \quad m_2^0 \qquad m_2^{4|E|} \qquad m_n^0 \qquad m_n^{4|E|}$$
$$[W_V] : w_1^0 \quad w_1^1 \qquad w_1^{4|E|} \quad w_2^0 \qquad w_2^{4|E|} \qquad w_n^0 \qquad w_n^{4|E|}$$

Let $M_{[E]}(W_{[E]})$ denote the set of all men (women) introduced corresponding to the edges in the graph. We define similar arbitrary but fixed ordering. We denote these orderings by $[M_E]([W_E])$.

We denote the set of dummy agents introduced in *step 3* by $M_d(W_d)$, the arbitrary but fixed ordering over these agents is denoted by $[M_d]([W_d])$ respectively for dummy men and women. Similarly, the padding agents introduced in step four are denoted by $M_p(W_p)$, and the fixed arbitrary ordering over those is denoted by $[M_p]([W_p])$. Note that the ordering $[M_p]([W_p])$ respects the ordering $y_1^i \quad y_0^i(x_1^i \quad x_0^i)$.

We will now give a linear orders on set of men and women:

$$_M : [M_d] \quad [M_E] \quad [M_V] \quad [M_p]$$
$$_W : [W_d] \quad [W_V] \quad [W_E] \quad [W_p]$$

Single-peaked: It is easy to see that the preferences of all the men are single-peaked with respect to axis $_W$ and with respect to $_M$ for all women. This implies that there exist a valid extension single-peaked extension of the partial preference list of every man and woman.

Single-crossing: It is known that, if every preference list in the profile is a sub-order of two complete preference orders then there a valid extension for each vote in the profile such that the preferences can be ordered in a way that the resultant profile is SP. We now present the two orderings over men and two orderings over women and show that all the preferences in the constructed instance I' is a sub-order of one of these. One set of ordering over men and women are the linear orders $\succ_M$ and $\succ_W$ presented above. The other set of ordering is as follows:

$$\succ_{M'} : [M_d] \quad [M_v] \quad [M_E] \quad [M_p]$$
$$\succ_{W'} : [W_d] \quad [W_E] \quad [W_v] \quad [W_p]$$

The agents from set M_v, M_p are sub-orders of $\succ_W$, and agents from sets M_E, M_d are sub-orders of $\succ_{W'}$. Similarly, agents from sets W_v, W_p are sub-orders of $\succ_M$ and agents from sets W_E, W_d are sub-orders of $\succ_{M'}$.
Hence, the constructed profile is both single-peaked and single-crossing. Note that this reduction shows the NP-completeness of the problem for SPSC profiles even when there are no ties. □

5.5 Egalitarian Stable Matchings

In this section, we show that EGAL-SMTI is W[1]-hard for parameter κ_1. We reduce from CLIQUE which in known to be W[1]-hard for parameter k-size of the clique to EGAL-SMTI through an intermediate reduction to MAX-SMTI. Hence, overall, we start from an instance of CLIQUE, construct an instance of MAX-SMTI using parameter preserving construction by Marx and Schlotter (MS10). Then, we convert this MAX-SMTI instance to EGAL-SMTI to obtain the desired reduction. We further show that the reduced instances for both Marx and Schlotter (MS10) for MAX-SMTI and for our constructed instance of EGAL-SMTI is SPSC.

The decision version of EGAL-SMTI asks for an existence of marriage with score at most some given integer η. Similarly, the decision version of MAX-SMTI asks for a matching of size greater than equal to k where k is some given integer. Note that in Theorem 16 we consider the decision version of EGAL-SMTI. We first show the construction and equivalence, then we turn to structural properties of constructed preferences for both problems.

Theorem 16. *Egal-SMTI is W[1]-hard parameterized by κ_1 even when the profile is SPSC.*

Proof. We first describe the preference profile obtained from CLIQUE for MAX-SMTI instance in the reduction from (MS10). Let $G = (V, E)$ along with a natural number k be the clique instance such that $|V| = n$ and $|E| = m$. Assume the canonical ordering of vertices from set $[n]$. We will construct an instance $I = (X, Y, r)$ with X, Y being the set of women and men, and r being the integer input for decision version of MAX-SMTI.

Agents: For each $i \in [k]$ we introduce, $X^i = \{x^i_u \mid u \in [n]\}$ and a global agent x^i_0, and the corresponding men $Y^i = \{y^i_u \mid u \in [n]\}$, y^i_0. Similarly, for each $(i, j) \in \binom{[k]}{2}$ (we recall from Section 5.2 that we write $\binom{[n]}{2}$ for set $\{(i, j) \mid 1 \leq i \leq j \leq n\}$) we introduce, $X^{i,j} \cup \{x^{i,j}_0\}$ with $X^{i,j} = \{x^{i,j}_{u,z} \mid u < z, (u, z) \in E\}$, and the set of men $Y^{i,j} \cup \{y^{i,j}_0\}$ with $Y^{i,j} = \{y^{i,j}_{u,z} \mid u < z, (u, z) \in E\}$. Additionally, introduce set of women $P = \{p_i \mid i \in [\binom{k}{2} + 2]\}$ and set of men $Q = \{q_i \mid i \in [\binom{k}{2} + 2]\}$. Let $X = X^i \cup \{x^i_0\} \cup X^{i,j} \cup \{x^{i,j}_0\} \cup P$ be the overall set of women, and $Y = Y^i \cup \{y_0^i\} \cup Y^{i,j} \cup \{y_0^{i,j}\} \cup Q$ be the overall set of men in the instance such that $|X| = |Y| = t$. Let be the bijection from $[\binom{k}{2}]$ to set $\binom{[k]}{2}$, and for each $i \in [k], u \in [n]$ let $C(i,u) = \{x^{i,j}_{u,z} | i < j \le k, u < z, (u,z) \in E\} \cup \{x^{j,i}_{z,u} | 1 \le j < i, z < u, (z,u) \in E\}$.

We introduce $2t^2$ dummy women $[D] = [d_1, d_2, \ldots, d_{2t^2}]$ and men $[D'] = [d'_1, d'_2, \ldots, d'_{2t^2}]$

We now move on to the preferences of the agents. For any set S of agents, we denote the indifference among these agents by (S) and a fixed ordering by $[S]$. Let k' denote $\binom{k}{2}$.

Preferences:

$x_u^i : y_u^i \quad y_0^i \quad [D']$	$y_u^i : x_0^i \quad [C(i,u)] \quad x_u^i \quad [D]$
$x_0^i : y_0^i \quad (Y^i) \quad [D']$	$y_0^i : [X^i] \quad x_0^i \quad [D]$
$x_{u,z}^{i,j} : y_{u,z}^{i,j} \quad [y_u^i, y_z^j] \quad y_0^{i,j} \quad [D']$	$y_{u,z}^{i,j} : x_0^{i,j} \quad x_{u,z}^{i,j} \quad [D]$
$x_0^{i,j} : y_0^{i,j} \quad (Y^{i,j}) \quad [D']$	$y_0^{i,j} : [X^{i,j}] \quad p_{^{-1}(i,j)} \quad x_0^{i,j} \quad [D]$
$p_h : q_{h+1} \quad y_0^{(h)} \quad q_h \quad [D']$	$q_{h'} : p_{h'} \quad p_{h'-1} \quad [D]$
$p_{k'+1} : (q_{k'+1}, q_{k'+2}) \quad [D']$	$q_1 : p_1 \quad [D]$
$p_{k'+2} : q_{k'+2} \quad [D']$	$q_{k'+2} : p_{k'+1} \quad p_{k'+2} \quad [D]$
$d_1 : d'_1 \quad [Y]$	$d'_1 : d_1 \quad [X]$
$d_2 : d'_2 \quad [Y]$	$d'_2 : d_2 \quad [X]$
$\vdots$	$\vdots$
$d_{2t^2} : d'_{2t^2} \quad [Y]$	$d'_{2t^2} : d_{2t^2} \quad [X]$

Figure 5.1: Preferences for egalitarian matching instance

In Fig. 5.1 $h \in [k']$ and $h' \in [2, k'+1]$ where $[2, k'+1] = \{2, 3, \ldots, k'+1\}$. For ordered sets $[Y], [D], [D'], [X]$ we will define the ordering later in *analysis of the profile*, for now, it is okay to assume any arbitrary fixed ordering of agents from each set. Note that only $\binom{k}{2} + k + 1$ women – x_0^i for $i \in [n]$, $x_0^{i,j}$ for $(i,j) \in \binom{[k]}{2}$, and $p_{k'+1}$ have indifference. In the above construction, an instance without blue agents is from the original reduction given in (MS10). We use these additional agents for transforming from Max-SMTI to Egal-SMTI to establish the required hardness.

Note that for dummy agents from sets D and D', any stable matching will match d_i to d'_i for $i \in [2t^2]$. Hence, for a stable matching M, any *unmatched agent* in the instance except for dummy agents will contribute a large amount to the total score with respect to M. By setting the score for Egal-SMTI appropriately, we can differentiate between the cases where the original instance had a matching of size greater than or equal to r or strictly less than r where r is the input to the Max-SMTI instance. We set $= t \quad 2r + (2t^2 + t) \quad 2(n - r) + 2 \quad 2t^2$. We next give the proof of equivalence.

Forward direction: Given a YES instance of Max-SMTI, let M be the matching with at least r matched pairs from set $X \cup Y$. We now construct a matching M' for the constructed Egal-SMTI instance. We augment M to M', and add the matching pairs of all $2t^2$ dummy agents as described earlier. We claim that M' is a stable matching with score at most . The stability for M' follows from the stability of M and the constructed preferences of the dummy agents. For the score of M', the matched $2r$ agents from $X \cup Y$ contribute at most $2t$ (since these $2r$ agents are matched to the agents which appear in their top t positions) and the remaining unmatched agents contribute $2(n - r) \quad (t + 2t^2)$. Each dummy pair contributes exactly 2 to the score. Hence, the overall score $t \quad 2r + (t + 2t^2) \quad 2(n - r) + 2 \quad 2t^2 \leq$.

Reverse direction: Here we show that if a stable matching M of Egal-SMTI matches strictly less than r pairs from the set of agents $\{X \cup Y\}$ then score(M) $>$. We now compute the lower bound on the score of such a matching M. We again emphasize that in any stable matching, for $i \in [2t^2]$, agent d_i is matched to agent d_i'. To consider the tightest case, let the matching M contain $(r - 1)$ matching pairs from $\{X \cup Y\}$. These matched pairs will contribute at least $2(r - 1) \quad 1$ to the score(M). The unmatched agents from $X \cup Y$ will add $2(t - r + 1) \quad (1 + 2t^2)$, and the dummy agents will add $2 \quad 2t^2$ to the score. The overall score of $(M) > 2(r - 1) \quad 1 + 2(t - r + 1) \quad (1 + 2t^2) + 2 \quad 2t^2 >$. Hence, any matching M with score(M) $\leq$ matches at least r pairs from $\{X \cup Y\}$ which implies that the given instance is a YES instance for Max-SMTI (since the agents from these pairs are matched to each other in M they must be stable in the original instance).

This completes the argument of equivalence.

Analysis of the profile: Consider the orderings:

$$[X] : [x_0^i] \quad [x_0^{i,j}] \quad X^{i,j} \quad X^i \quad p_1 \quad p_2 \qquad p_{k'+2}$$

$$[Y] : [y_0^i] \quad [y_0^{i,j}] \quad Y^i \quad Y^{i,j} \quad q_{k'+2} \quad q_{k'+1} \qquad q_1$$

The single-peaked ordering for women is $_W : [X] \quad [D]$ and for men is $_M : [Y] \quad [D']$. In the preference profiles constructed in Fig. 5.1; by $[S]$ for some set S of agents, we denote the ordering of these agents according to the single-peaked ordering described above. Given this, it is easy to verify that the preferences for all candidates are single-peaked with respect given ordering.

Preferences of men are single-crossing with ordering:

$$[Y^i] \quad [Y^{i,j}] \quad [y_0^{i,j}] \quad q_1 \quad q_{k'+2} \quad [y_0^i] \quad [q_{h'}] \quad [D']$$

of the agents in the instance, and the preferences of women are SC with respect to:

$$[x_0^i] \quad [x_0^{i,j}] \quad [X^i] \quad [X^{i,j}] \quad p_h \quad p_{k'+1} \quad p_{k'+2} \quad [D].$$

Note that the above analysis shows that the profile from the reduction in (MS10) is also SPSC (since it is a sub-profile of the constructed profile), this implies hardness for Max-SMTI even when preferences are SPSC. The number of agents with ties in the constructed instance is same as that in Max-SMTI, hence, Egal-SMTI is W[1]-hard parameterized by $_1$ even when only women can have ties and the preferences are SPSC. □

5.6 Domain-Restricted Manipulation

In this section, we consider the issue of single-agent manipulation for the Stable Marriage problem. We focus on the notion of *permutation manipulation*, which only allows the preference lists to be a permutation of the complete ordering over the agents of the opposite sex. The first comprehensive study of permutation manipulation was done by Teo et al. in (TST01). They show that unlike truncation manipulation, the permutation manipulation model is far more restrictive in terms of possibilities of manipulation. In particular, in the men-proposing variant of the Gale-Shapley mechanism, women cannot always force the mechanism to return the women optimal matching. An easy illustration is given by the example where men's preferences are narcissistic i.e. for each woman; there exists a unique man which places her at the top of his preference list. In this case, under any permutation manipulation, no women can improve her matching partner under Gale-Shapley. Teo et al. also gave a polynomial time algorithm to find an optimal manipulation for any given women. In this case, the optimal manipulation means the set of preferences which yield the best possible partner under permutation manipulation (which can be different from the women optimal partner).

The work was followed by Vaish and Garg in (VG17). They show that the optimal permutation manipulation can be obtained by only changing the position of exactly one agent in the preferences of the manipulating woman. The authors refer to such manipulation as "inconspicuous manipulation". In what follows in this section, we show that the possibilities of manipulation on restricted domains are limited when we additionally force the manipulated profile to be in the domain under consideration. In particular, we observe that unlike the general domains, an inconspicuous manipulation cannot always achieve an optimal partner while staying within the domain. We also show that the optimal partner obtained while staying within the domain (not necessarily inconspicuously) can be strictly sub-optimal compared to one that can be obtained when there are no restrictions. Now, we provide examples to support this discussion.

Observation 3. *Obtaining an optimal manipulation inconspicuously while staying within the domain is not always possible for a Stable Marriage instance.*

Example 1: *Sub-optimal Manipulation when restricted to inconspicuous manipulation*

m_4 :	1	2	3	4	5	w_4 :	5	4	3	2	1
m_1 :	2	1	3	4	5	w_5 :	5	4	3	2	1
m_5 :	2	1	3	4	5	w_2 :	4	3	2	5	1
m_2 :	3	2	4	1	5	w_1 :	3	4	2	5	1
m_3 :	3	4	2	1	5	w_3 :	2	3	1	4	5

Figure 5.2: Original Preference Profile

Note that in Figure 5.2 the numbers denote the corresponding agent of the opposite sex.

The profile P in Fig. 5.2, is single-peaked with axes:

$_m := m_1 - m_2 - m_3 - m_4 - m_5$ $_w := w_1 - w_2 - w_3 - w_4 - w_5$

The profile is also single-crossing with respect to the given sequence of men and women. If we run the Gale-Shapley algorithm on profile P in Fig. 5.3 we obtain the matching:

$$M : \{(w_1, m_4), (w_2, m_5), (w_3, m_2), (w_4, m_3), (w_5, m_1)\}$$

Consider the profile (P') with the preference list of $w_1 := 3 \quad 2 \quad 1 \quad 4 \quad 5$. We keep preferences of all other agents as it is. The blue colored men represent the new matching (under P').

Notice that in the profile P' the partner for w_1 is m_3 which is an optimal partner according to her true preferences. It is only possible to obtain m_3 as a partner when we place $3 \quad 1 \quad 4$ in the preferences of w_1. It is easy to see that the described preference relation cannot be obtained inconspicuously while staying within the domain but can be obtained conspicuously while staying within the domain (profile P' achieve such a preference relation).

Observation 4. *e optimal partner which can be obtained while staying within the domain can be sub-optimal compared to the partner that can be obtained without domain restriction for a Stable Marriage instance.*

Example 2: *Sub-optimal Manipulation with domain restriction*

m_1 : 5	4	1	2	3		w_1 : 1	3	2	<u>4</u>	5
m_2 : 4	1	2	3	5		w_2 : 4	1	<u>3</u>	2	5
m_3 : 2	1	4	5	3		w_3 : 1	3	2	4	<u>5</u>
m_4 : 1	2	3	4	5		w_4 : 5	4	1	3	<u>2</u>
m_5 : 1	2	4	5	3		w_5 : 3	<u>1</u>	4	2	5

Figure 5.3: Original Preference Profile

The profile (P) in Fig. 5.3, is single-peaked with axes:

$$_m := m_5 - m_4 - m_1 - m_3 - m_2 \qquad _w := w_3 - w_2 - w_1 - w_4 - w_5$$

The matching (M) in the original preferences profile P is:

$$M : \{(w_1, m_4), (w_2, m_3), (w_3, m_5), (w_4, m_2), (w_5, m_1)\}$$

For the manipulated profile (P') we have $w_1 := m_3 \quad m_5 \quad m_4 \quad m_2 \quad m_1$. Note that the manipulated profile is not single-peaked with respect to $_m$ axis. In the manipulated profile (P') the partner for w_1 is m_3 which is her optimal partner in the original profile. We claim that there does not exist any profile which is SP and gives matching with m_3. In order to get a proposal from m_3, w_1 must reject m_4 which can only happen if $m_5 \ _{w_1} \ m_4$ (and m_3 is preferred over both of these). This cannot be achieved by staying within the domain.

5.7 Domains with Unique Stable Matching

In this section, we present a class of preferences which admits unique stable matching (1-D Euclidean Preferences) and study a few properties of domains with Globally Ranked Pairs defined in (ALMO07). An important motivation to study these classes is that they do not suffer from Roth's impossibility theorem (Rot82, Theorems 4 and 6) and are not vulnerable to manipulation.

5.7.1 1-D Euclidean Preferences

Lemma 7. *A two-sided matching instance with strict complete orderings over agents that respect a 1D-Euclidean ordering admits a unique stable matching.*

Proof. The preferences of the Euclidean domain are derived from the ordering of men and women on the common axis (see Fig. 5.4 for one such axis). Using this fact, it is clear that the first choice for every agent will be the next consecutive agent from the opposite gender immediately to its right or to the left (in particular, the closer of the two options).

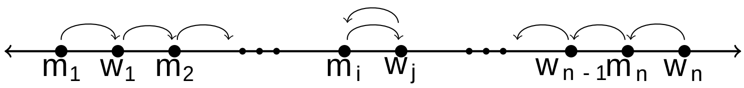


Figure 5.4: Ordering of the agents on the axis along with their first choices

Claim 2. *For a 1D-Euclidean Domain, there exist a pair of agents* (m_i, w_j) *such that they are each other's top preferences.*

Proof. We denote by F_i the first preference of an agent i. We'll prove the claim using contradiction. Assume that there does not exist any (m_i, w_j) pair such that m_i and w_j are the first preferences of each other. Without loss of generality assume that m_1 is the leftmost agent and w_n is the rightmost agent. Using the property of the domain, we know that $F_{m_1} := w_1$ and $F_{w_n} := m_n$. From our assumption it implies that $F_{w_i} := m_j$ only if $j > i$. But this gives a contradiction for w_n since $F_{w_n} := m_k$ for k 6 n. □

It follows that the (m_i, w_j) obtained from Claim 2, has to be matched to one another in any stable matching. We form a pair (m_i, w_j) and remove them from the instance. It is easy to check that the domain follows the hereditary property. Hence, the obtained profile is 1D-Euclidean. We repeatedly apply Claim 2, form a pair, and remove the pair of agents from the instance. Since at every step, the pair removed was the forced matching pair; the instance admits a unique stable matching. □

It is possible to find the argued unique stable matching efficiently. The matching algorithm given by Bartholdi III and Trick in (BIT86) works for the Euclidean domain. The algorithm runs in time $O(n)$ time, which means it does not even read the input instance completely.

5.7.2 Globally Ranked Pairs

We now consider the preferences derived from Globally Ranked Pairs (GRP) and we denote the Stable Matching problem in this setting by SM-GRP. As mentioned earlier, the notion was first defined in (ALMO07). The authors show that these preference profiles follow the *Globally-Acyclic Preferences (GAP)* restriction (Proposition 1.1, (ALMO07)) which implies the that the instance cannot have any odd or even sized ring/cycle. As GRP preferences follow GAP restriction, for any instance (I) of SM-GAP, we do not have any rotations in I (since rotations are even length rings). Since the set of rotations is in one-to-one correspondence with the set of stable matchings, there exists a unique stable matching for I since there are no rotations.

We next give an alternative proof for the Unique Stable Matching in these instances. We show that the profile follow Eckhout's condition 1 from (Eec00) which indeed implies unique stable matching for the instance.

Lemma 8. *e preference pro le derived from the Globally Ranked Pairs follows Eckhout's Condition 1.*

Proof. Let be the global order of the pairs and (m_i, w_j) be the first pair in the order. We claim that both m_i and w_j are the first preferences of each other. Otherwise, if m_i prefers w_k w_j then the pairs (m_i, w_j) and (m_i, w_k) don't follow the global rank of pairs which is a contradiction. In any stable matching, (m_i, w_j) must be matched to each other. We pair up (m_i, w_j) and remove all them from the instance by deleting all the pairs m_i or w_j participate in from (let us call the new ordering ^{0}). It is easy to see that the domain follows the hereditary property. Hence, the modified instance still belongs to the domain with new global ordering ^{0}. Repeating the same procedure for n iterations will return the unique stable matching and ordering of man-woman pairs. We observe that both m_i and w_j matched in the i^{th} iteration must have a partner with rank $6\ i$ in their respective preferences. The ordering of men and women corresponding to the order returned by the algorithm follows Eckhout's condition 1. □

Corollary 1. *For a two-sided stable matching instance I, if all the agents from one of the sides have identical preferences and the agents from other side have any arbitrary preferences, then I admits a unique stable matching.*

Proof. Let us assume that all the men have an identical preference list $m_i := w_1$ w_2 w_n. We will now construct the global ordering of pairs such that all the preferences in the instance comply with the ordering. Place all the pairs w_1 participates in at the top followed by pairs of w_2 and so on to construct . It is easy to see the preferences of both men and women have the same ordering as . It follows that instance I belongs to the globally ranked pairs domain and hence admits a unique stable matching. □

Lemma 9. *ere is no incentive for single-agent manipulation even when the relax the constraint of stability of manipulated matching with respect to original instance.*

Proof. For contradiction, let M be the unique stable matching in the original instance and M' be the stable matching in the manipulated instance and M' is strictly better for manipulating agent (say w_1). Rohit et al. ((VG17) Theorem 5) showed that the matching obtained through an *optimal manipulation* is stable with respect to original preferences. Let the set S be the set of stable matchings obtained from manipulated profiles which give strictly better matching partner for w_1. $S \neq \emptyset$ since $M' \in S$. Let $M'' \in S$ be the *optimal manipulation* for w_1. We know M'' is stable with respect to original preferences. Since the original preferences admit unique stable matching, $M''(w_1) = M(w_1)$. For $w_i := M''(w_1) \succeq M'(w_1)$ (since M'' is optimal). This contradicts a contradiction as w_1's partner in M' was supposed to be strictly better than that in M. □

Next we discuss the properties of preference domains which are close to 1D-Euclidean or Globally Ranked Pairs domain. We show a sharp contrast in the number of stable matchings possible in these domains.

5.7.3 Nearby Domains with large number of stable matchings

We discuss the examples in two sided matching setting. First we will define some terminology for ease in generalization of profiles. P_i denote the ordered set of candidates $(2i-1 \succ 2i)$ and $\overline{P_i}$ denote the set $(2i \succ 2i-1)$. P_i appearing in man's (woman's) preference represents pair $(w_{2i-1} \succ w_{2i})((m_{2i-1} \succ m_{2i}))$ respectively.

Profiles with arbitrary preferences on one side and two preference profiles on the other side:

Example 3: *2-profiles on one side*

$$
\begin{aligned}
m_1 &: P_1 \succ P_2 \succ \cdots \succ P_{\frac{n}{2}} \\
m_2 &: \overline{P_1} \succ P_2 \succ \cdots \succ P_{\frac{n}{2}} \\
m_3 &: P_2 \succ P_3 \succ \cdots \succ \overline{P_1} \\
m_4 &: \overline{P_2} \succ P_3 \succ \cdots \succ P_{\frac{n}{2}} \succ \overline{P_1} \\
&\vdots \\
m_n &: \overline{P_{\frac{n}{2}}} \succ \overline{P_{\frac{n}{2}-1}} \succ \cdots \succ \overline{P_2} \succ \overline{P_1}
\end{aligned}
\qquad
\begin{aligned}
w_{even} &: P_1 \succ P_2 \succ \cdots \succ P_{\frac{n}{2}} \\
w_{odd} &: \overline{P_1} \succ \overline{P_2} \succ \cdots \succ \overline{P_{\frac{n}{2}}}
\end{aligned}
$$

Figure 5.5: Instance with 2 types of preferences on one side

In the above profile the pair (w_1, w_2) appears on the top for (m_1, m_2). Also, (m_1, m_2) appears on the top for all the women; in particular for (w_1, w_2). Hence, the pairs will be matched to each other in any stable matching. We now repeat the same argument on the modified instance for (m_3, m_4) and (w_3, w_4) pair. The argument holds for all $\frac{n}{2}$ pairs P_i's. In each of the pair P_i both $((m_{2i-1}, w_{2i-1}), (m_{2i}, w_{2i}))$ are stable matches. With each pair contributing 2 possible configurations for stable matching, we obtain $2^{\frac{n}{2}}$ stable matchings.

SP-SC Pro les: We note the profile in Example 3 is already SP with axes:

$$_m := m_1 \quad m_2 \qquad m_n$$

$$_w := w_1 \quad w_2 \qquad w_n$$

All the women (men) have their preferences single-peaked with respect to axis $_m(\ _w)$ respectively. Let us analyze the ordered pair of candidate P_i. $w_{2i-1} \quad w_{2i}$ until the preference profile of man m_{2i-1} and $w_{2i} \quad w_{2i-1}$ from thereafter. Now we'll analyze the ordering across the pairs. Every pair P_i crosses all pairs P_j such that $j > i$ for $i, j \ 2 \ [1, n=2]$ once it appears at the top in the preferences of men m_{2i-1}, m_{2i}. This ensures that every block crosses every other block at exactly once. Overall, any two agents cross each other exactly once, hence, the profile is single-crossing.

1D Euclidean Pro les with ties: We borrow the notations from Example 3 and add P_i^0 which indicates the indifference between the pair i.e.$(w_{2i-1} \quad w_{2i})$.

Example 4: *1D Euclidean with ties*

$$\begin{array}{llllll}
m_1 : \bar{P_1} & P_2 & P_{\frac{n}{2}} & w_1 : \bar{P_1} & P_2 & P_{\frac{n}{2}} \\
m_2 : P_1 & P_2 & P_{\frac{n}{2}} & w_2 : P_1 & P_2 & P_{\frac{n}{2}} \\
m_3 : \bar{P_2} & P_3 & P_1 & w_3 : \bar{P_2} & P_3 & P_1 \\
m_4 : P_2 & P_3 & P_{\frac{n}{2}} \quad P_1 & w_4 : P_2 & P_3 & P_{\frac{n}{2}} \quad P_1 \\
\vdots & & & \vdots & & \\
m_n : P_{\frac{n}{2}} & P_{\frac{n}{2}-1} & P_2 \quad P_1 & w_n : P_{\frac{n}{2}} & P_{\frac{n}{2}-1} & P_2 \quad P_1
\end{array}$$

Figure 5.6: 1D-Euclidean Preferences with ties

For each pair $P_i := (w_{2i-1}, w_{2i})$, P_i appears on the top for pair of men (m_{2i-1}, m_{2i}) and vice verca for pair $P_i := (m_{2i-1}, m_{2i})$ in the women's preferences. Hence, in any stable matching the agents from these pairs will get matched to each other. Note that for each such pair P_i, both $M := (m_{2i-1}, w_{2i-1}), (m_{2i}, w_{2i})$ and $M^0 := (m_{2i-1}, w_{2i}), (m_{2i}, w_{2i-1})$ are *weakly stable* matchings. Hence, will $\frac{n}{2}$ such P_i^0s there exist exponential number of weakly stable matchings.

5.8 Matching Critical Set

In this section, we consider the problem of finding, given a subset of "critical" agents, a stable matching that matches all critical agents. We call the problem Matching Critical set. We study this in the context of SRTI, where we already obtain W[1]-hardness when parameterized by the number of critical agents.

Theorem 17. *In the se ing of SRTI, Matching Critical Set is W[1]-hard when parameterized by the size of the critical set even when the preferences are SPSC.*

Proof. We reduce from the MULTI-COLORED INDEPENDENT SET (MCIS) problem. Given a partition of vertices in k color classes, $MCIS$ asks for an independent set of size k which contains exactly one vertex from each color class. For a given instance of $MCIS$, let $U_1, U_2, \quad , U_k$ be the set of vertices corresponding to k color classes. Without loss of generality assume that every color class contains n vertices where n is odd. For every color class, we introduce set U_i of n vertex agents $(u_i^1, u_i^2, \quad , u_i^n)$ and a critical agent C_i for $i \in [k]$. We denote the set of k-critical agents by C. Let a linear order over the vertex agents for a given color class be $\,^{v}_{i} : u_i^1 - u_i^2 - \quad - u_i^n$, and the ordering of critical agents be $\,_{C} : C_1 \quad C_2 \quad \dots \quad C_k$. We next define the preferences of the agents.

Preferences: For each vertex agent u_i^j the preference order is:

$$u_i^j : (u_i^{j-2} \quad u_i^{j-1} \quad u_i^{j+1} \quad u_i^{j+2}) \quad [N(u_i^j)] \quad C_i$$

for $i \in [k], j \in [n]$. Note that indices in the superscript are computed modulo n. Further, each critical agent C_i has preference order:

$$C_i : u_i^1 \quad u_i^2 \quad \dots \quad u_i^n$$

We ask for a stable matching which matches all the agents from set C. This completes the description of the instance, we now turn to the proof of equivalence.

Forward direction: Given a $MCIS$ S, for $u_i^j \in S$, we form a matching pair (u_i^j, C_i) for $i \in [k]$. Now we modify $\,^{v}_{i}$ by removing agent corresponding to the given independent set for every color class and maintaining the rest of the order (let $\,^{v'}_{i}$ denote the modified ordering). With remaining $n - 1$ vertex agents in each color class, we match $\lfloor \frac{n}{2} \rfloor$ pairs of adjacent agents going from left to right according to $\,^{v'}_{i}$.
In the proposed matching, for $u_i^j \in S$, all the agents appearing before C_i in their preference orders are matched to their first choice, hence they don't participate in blocking pair with u_i^j. By the same reasoning, the critical agent C_i does not have any hope of forming a blocking pair with any other u_i^j. Hence, the proposed matching is stable and matches all agents from C.

Reverse direction: Given a stable matching M involving all C_i^0s, we claim that the vertex agents matched to C_i^0s under the matching M form a $MCIS$. It is easy to see that the set (say S^0) of the matching partner of C_i^0s, contains one vertex from *each color class*. This is because each C_i prefer agents from unique color class. For the contradiction, let $u, v \in S^0$ share an edge. Notice that u and v both strictly prefer the set of their neighbors to respective C_i^0s. This implies that (u, v) is a blocking pair for M which is a contradiction. Hence, we can recover $MCIS$ from the matching. This completes the proof of equivalence.

Notice that the preference profile obtained in the construction is single-peaked and single-crossing with respect to:

$$\,_{1} : [U_1] \quad [U_2] \qquad [U_k] \quad \,_{C}$$

where each $[U_i]$ denote the ordering according to $\,^{v}_{i}$. To show the single-crossingness, consider another ordering:

$$_2 : [U_2] \quad [U_3] \qquad [U_{k-1}] \quad [U_k] \quad [U_1] \quad c;$$

$$_3 : [U_3] \quad [U_4] \qquad [U_k] \quad [U_2] \quad [U_1] \quad c;$$

$$\vdots$$

$$_k : [U_k] \quad [U_{k-1}] \qquad [U_3] \quad [U_2] \quad [U_1] \quad c.$$

By resolving the ties and arranging agents in $N(u_i^j)$ according to one of the $_i$ in the preferences of each of u_i^j for $i \in [k], j \in [n]$; we order these agents according to the increasing order of the color classes. This is a single-crossing ordering over the agents. Thus, this reduction shows $W[1]$-hardness on SPSC profiles when parameterized by the size of the critical set, for constant $_2$. □

The following observation follows as a Corollary of (MS10, Theorem 2).

Theorem 18. *In the se ing of SMTI, Matching Critical Set is $W[1]$-hard parameterized by $_1$ for a critical set of size one*

Proof. Recall the construction from fig. 5.1 and ignore the dummy candidates from the construction (i.e. consider the part of the profile from the original reduction from (MS10)). We set the of critical agents $C := \{p^{k'+2}\}$ a singleton set. We claim that the instance admits matching involving $p^{k'+2}$ if and only if there exist a clique of size k.

The main idea of the reduction is that the agent $p^{k'+2}$ is involved only in the maximum-sized matching, and such a matching corresponds to a YES instance of clique. Hence, a matching involving $p^{k'+2}$ is equivalent to existence of a clique. For the further details of the we refer the reader to (MS10, Theorem 2). □

5.9 Special cases of (a,b)-Supermatch

The $(n, 0)$-Supermatch:

In this section, we show that checking if $(n, 0)$ supermatch exists and finding it for the given stable matching instance can be done efficiently. We also give a novel algorithm to check whether a given matching is $(n, 0)$ - supermatch with polynomial running time. Unlike the $(2, 0)$ case shown by (GSOS17) (Theorem 3), the $(n, 0)$ supermatch does not cease to exist for $n > 2$. The simplest example of $(n, 0)$ is considering the Stable Matching instance where *Men optimal (MO)* and *Women optimal (WO)* matchings are completely different. In that case, we can consider one of MO or WO as initial matching and the other one as the final matching; this shows the existence of $(n, 0)$-supermatch in this instance.

Existence and search of $(n, 0)$ - supermatch

We now describe the procedure to check whether given instance of Stable Matching admits a $(n, 0)$ - supermatch.

Consider the Gale-Shapley output ( ) on the given instance. From Theorem 12, we know that ( ) matches each man to its best possible partner and each woman to her worst possible partner among the set of stable matchings. Run the Gale-Shapley by swapping roles of men-women, let the matching obtained be $'$. It is easy to see that $'$ is optimal for all women and pessimal for all men. We go through the partners assigned to each man in and $'$, if $(m) = {}'(m)$ for some $m \in M$ then we declare that the instance does not have $(n, 0)$ - supermatch. This is true because if the optimal (m) and pessimal $'(m)$ partner of m is same then m will have the same partner in all stable matchings, hence, there cannot exist a $(n, 0)$ - supermatch in the instance.
If the partners for all men are different in and $'$ (i.e. $\forall m \in M: (m) \neq {}'(m)$) then we output one of or $'$ as $(n, 0)$ - supermatch for the instance.

In the next section, we describe an algorithm to verify that given matching is $(n, 0)$ - supermatch.

Verification Algorithm

Our algorithm is essentially an extension of simple Gale-Shapley procedure keeping in mind the forbidden pairs (the matching pairs given in the input matching) constraints imposed by initial stable matching given as an input for the $(n, 0)$ - supermatch verification problem.

Description of the algorithm: The input to the algorithm is the preference profile indicating the preferences of all agents and a stable matching $'$ to verify. Our algorithm proceeds in iterations with each iteration consisting of two procedures – *Find Match procedure* and *Cleaning preferences procedure* applied sequentially. We'll now describe the flow of our algorithm using these two procedures.

In the first iteration, we begin with the *Find-Match procedure* which runs Gale-Shalpley on given preference profile and terminates by finding the male optimal matching ( ) (In the first iteration, the arguments to *Find-Match procedure* are m_1 as unmatched man, preferences from 1^{st} rank for m_1, original complete preference profile, and an empty initial matching). We check the partner of each man m in ; if $(m) = {}'(m)$ for some $m \in M$ then we note that contains a forbidden pair. We store the set $M' \subseteq M$ s.t. for $m \in M'$, $(m) = {}'(m)$. If $M' = \emptyset$, then we return *'YES'* since is the witness for $'$ being $(n, 0)$ - supermatch. If $M \neq \emptyset$, we call the *Cleaning preferences procedure*. In this procedure, for each woman $w \in W$ we remove all men appearing after (w) in $_w$. For an arbitrary $m \in M'$, we break the matching and remove m from $_{(m)}$ (we set m' to be this man in Algorithm 1.

From iteration two on-wards, we get the partial matching (where exactly one man m' is unmatched), and we get the modified preference profile ( ). We run the Gale-Shapley algorithm from that state (with $n - 1$ matched pairs). The algorithm starts with m' proposing to the woman appearing further down (after (m)) in his preference list. We stop when all the men are matched (say in matching $''$) and check for the set M' as described earlier. According to the state of set M^0 we either return with output *'YES'* or call *Cleaning preferences procedure*. We continue iterating over these two procedures unless we return *'YES'* or we run out of proposals for any m 2 M (i.e. we reach till the end of preference list of m) in which case we return *'NO'* instance.

Algorithm 1: Algorithm for verifying $(n, 0)$-supermatch

Input: A preference profile and matching ^{0}
Output: 'YES' if ^{0} is $(n, 0)$ - supermatch, 'NO' otherwise.

```
    None                          . initializing the intermediate matching
M⁰  None                          . initializing the set of forbidden pairs
m⁰  m1                            . initializing unmatched man
rank_m⁰  1        . initializing proposing sequence for m⁰ in next
  iteration
while True do
    M⁰                                     . reset set of forbidden pairs
       Find - Match(m⁰, rank_n⁰, , )      . calling Find-Match procedure
    for i = 1 to n do
        if  (mi) ==  ⁰(mi) then
            M⁰   M⁰ [ (mi,  (mi)                . add a forbidden pair to M
        end
    end
    if M⁰ =   then
        return 'YES'              . found matching with no forbidden pairs
    end
    m⁰   m s.t. m 2 M⁰                   . choosing arbitrary man in M⁰
         n (m⁰,  (m⁰))            . breaking the marriage for chosen man
    rank_m⁰ = rank(m⁰,  (m⁰)) + 1             . proposing sequence for m⁰
       Cleaning - Preferences( , )    . calling Cleaning-Preferences
     procedure
end
return 'NO'
```

Proof of correctness:

Lemma 10. *If algorithm return a matching , is stable.*

Proof. We show the proof by contradiction. Let (m_i, w_j) be the blocking pair in . This implies that: w_j $_{m_i}$ (m_i) and m_i $_{w_j}$ (w_j). Since m_i is matched to someone worse than w_j, in the run of Algorithm 1 m_i proposed to w_j and got rejected. This can happen either when m_i is present in w_j^0s list but she rejected him favor a better partner or m_i is not present in the preference list of w_j which again can happen only when w_j is matched to someone better (since the *Cleaning Preferences* procedure only remove men from w_j^0s list when she is matched or m_i is the forbidden partner which cannot be the case since is the outcome of the algorithm). This implies that w_j is matched to someone strictly better than m_i in matching , which contradicts the assumption that (m_i, w_j) forms a blocking pair. □

It is easy to see that whenever algorithm return *'YES'* we have the witness matching for which all men have different partners than in the given matching μ' and hence μ' is a $(n, 0)$ - supermatch.
Let S be the set of all stable matchings which avoid matching the agents paired in input matching μ'. We claim that, if $S \neq \emptyset$, then there exist $s \in S$ such that s is *male optimal* (i.e. simultaneously optimal for all men). This can be shown as follows: Consider matchings $s_1, s_2 \in S$ such that m_1 is assigned better partner in s_1 compared to s_2, and m_2 is assigned better partner in s_2 compared to s_1. Notice that for m_1 and m_2, their partners in both matchings s_1, s_2 are not forbidden since $s_1, s_2 \in S$. Consider the matching $s_3 = s_1 \vee s_2$, s_3 belongs to S as S contains all stable matchings which does not contain forbidden pairs. Hence, there exist $s \in S$ which is simultaneously optimal for all men.
We need to show that whenever $S \neq \emptyset$ the algorithm returns 'YES'.

Lemma 11. *Whenever $S \neq \emptyset$, Algorithm 1 returns 'YES'. Moreover the algorithm terminates by obtaining Stable Matching $s \in S$ such that s is male optimal matching in S.*

Proof. (Proof by contradiction) Consider for a given instance of stable matching the set S is non-empty. Let $s' \in S$ be the *male optimal* matching in S. We are in the case when algorithm returns *'NO'*. This implies that there exist a man $m_i \in M$ who ran out of proposals i.e. we exhausted all women in preference list of m_i in particular, m_i either proposed someone strictly less preferred than $s'(m_i)$ or $s'(m_i)$ is the last woman in m_i's list and she rejected m_i.

Let m_j be the first man who either proposes someone strictly less preferred to $s(m_j)$ or reaches end of the preference list. Let $s(m_j) = w_j$. We next analyze the proposals of m_j and show the contradiction.

- 2 possible scenarios:

  1. m_j proposes to w_j and gets rejected: m_j can get rejected either when m_j is present in the preference list of w_j, and m_j proposes to w_j but w_j is matched to someone better say m_k. Since m_j was the first man to propose someone strictly less preferred to $s(m_j)$, we have,

     $$m_k : w_j \succ_{m_k} s(m_k)$$

     . We also know,

     $$w_j : m_k \succ_{w_j} m_j$$

     due to the case we are in. This implies that (m_k, w_j) forms a blocking pair for s, a contradiction.
     In the other case, when m_j is not present in the preference list of w_j, and m_j proposed to w_j, this can happen only when w_j is matched someone better say m_k. After this point we repeat the above argument.

  2. (m_j, w_j) is a blocking pair (i.e. a matched pair in μ'): This case cannot occur since this contradicts with the membership of s in S.

Hence, we showed that the man m_j cannot exist. This implies that there does exist a man who proposes someone strictly less preferred to $s(m)$ in the run of our algorithm when $S \notin$. Given that the algorithm terminated only when stable matching is found or there exist a man who runs out of proposals, the algorithm must return s when $S \notin$. This completes the proof of our claim. □

Theorem 19. *Deciding whether an instance admits* $(n, 0)$*-supermatch is in* P*.*

Proof. The proof follows from Lemma 10 and 11. The running time of the algorithm is same as that of Gale-Shapley algorithm – $O(n^2)$ for instance with n agents on each side. □

Corollary 2. *Given a stable matching , and xed set of* A *of* a *candidates, it is polynomially veri able if there exists a stable matching* ^{0} *which re-orients the partners for men in set* A *and retrieves the partners for* $m \in M \setminus A$*.*

Proof. Let I^0 be the instance formed by restricting the original instance I to the candidates from set A and their matching partners ((A). We modify (trim) the preferences of agents in I^0 to eliminate any cross blocking pairs i.e. the blocking pairs formed between the agents from I^0 and the agents from $I \setminus I^0$. We refer the reader to (Knu97) (Chapter 6 section 3) for further details on the preference profile modification. Once the preferences are modified, we apply Algorithm 1 to the modified instance I^0. If the algorithm returns *'YES'* then we know that the matching can be re-oriented else the matching cannot be re-oriented without disturbing any other pairs. The proof of correctness for the procedure described can be worked out similarly as done for Algorithm 1. □

5.9.1 Hardness result for nearby problem

In this section we show that the decision problem of existence of some a - sized subset (S) of M (the set of all men) which for given some stable matching of admits a re-orientation of matchings (find a new partner for these a men) without disturbing the matching partner of any man outside set S is NP-hard. Notice that this is analogous to a weaker version of $(a, 0)$ supermatch question which asks for existence of a stable matching ^{0} whether all sets of size a can be re-oriented without affecting any matching outside this set for a given set of preferences.

Theorem 20. *It is* NP*-complete to decide whether there exist an* a - sized *subset* $S \subseteq M$ *which admits re-orientation of matchings in between the elements of* S*.*

Proof. We reduce from the CLIQUE problem. In the CLIQUE problem, we are given $(G : V, E)$ and an integer k; we need to determine whether G has k - sized Clique as a subgraph.

Construction: Our construction is similar to the one presented in (MI14) (Section 5, Theorem 5.8). Since we don't have any notion of the *score* of a matching in our re-orientation problem, we only need the first three steps from their reduction.

Step 1: (e vertex gadget) In this step for each $v_i \in V$ we create $4|E| + 1$ (man, woman) pairs. Each of these men will have either two or three entries in their preference list, while each woman will have exactly three entries. For this step, we'll only define the first two entries for men and the second and third entries for women. Third preference of man m_i^j and first preference of women w_i^j will be defined later. The preference lists of m_i^j, w_i^j for $j \in [0, 4|E|]$ looks as follows:

$$\begin{array}{llll}
m_i^0 : w_i^0 \succ w_i^1 \succ * & & w_i^0 : * \succ m_i^{4|E|} \succ m_i^0 \\
m_i^1 : w_i^1 \succ w_i^2 \succ * & & w_i^1 : * \succ m_i^0 \succ m_i^1 \\
m_i^2 : w_i^2 \succ w_i^3 \succ * & & w_i^i : * \succ m_i^{i-1} \succ m_i^i \\
m_i^{4|E|} : w_i^{4|E|} \succ w_i^0 \succ * & & w_i^{4|E|} : * \succ m_i^{4|E|-1} \succ m_i^{4|E|}
\end{array}$$

Step 2: (The edge gadget) For each edge $(v_r, v_s) \in E$, we introduce two men and two women. Each of the two men and women have two candidates in their preference lists. The preferences of these agents are shown below, again, the entries with asterisk will be defined later.

$$\begin{array}{ll}
m_{r,s}^1 : w_{r,s}^1 \succ * & w_{r,s}^1 : * \succ m_{r,s}^1 \\
m_{r,s}^2 : w_{r,s}^2 \succ * & w_{r,s}^2 : * \succ m_{r,s}^2
\end{array}$$

Step 3: (Complete the preference lists) For each edge, $(v_r, v_s) \in E$, with $r < s$ (according to arbitrary defined order over the vertices), we choose two men created in step 1, in correspondence to v_r and v_s by selecting the first man m_r^p (respectively, $m_s q$) from the sorted list $m_r^0, m_r^1, \ldots, m_r^{4|E|}$ (respectively, $m_s^0, m_s^1, \ldots, m_s^{4|E|}$ whose third choice has not yet been specified. In this step we complete the preferences for $m_r^p, m_s^q, w_r^{p+1}, w_s^{q+1}, m_{r,s}^1, m_{r,s}^2, w_{r,s}^1, w_{r,s}^2$ as described below,

$$\begin{array}{ll}
m_r^p : w_r^p \succ w_r^{p+1} \succ \underline{w_{r,s}^1} & w_r^{p+1} : \underline{m_{r,s}^2} \succ m_r^p \succ m_r^{p+1} \\
m_s^q : w_s^q \succ w_s^{q+1} \succ \underline{w_{r,s}^2} & w_s^{q+1} : \underline{m_{r,s}^1} \succ m_s^q \succ m_s^{q+1} \\
m_{r,s}^1 : w_{r,s}^1 \succ \underline{w_s^{q+1}} & w_{r,s}^1 : \underline{m_r^p} \succ m_{r,s}^1 \\
m_{r,s}^2 : w_{r,s}^2 \succ \underline{w_r^{p+1}} & w_{r,s}^2 : \underline{m_s^q} \succ m_{r,s}^2
\end{array}$$

At this stage, we have completed the preference lists for all the men and women created in step 2 and all men in step 1 (note that some men in step 1 have preference list of length two). However, there is still a set of women w_i^j created in step 1 for which we have not defined their top preference. For each of these women, we create a pair of a dummy man and a dummy woman such that the dummy pair place each other at their top preference, and we place w_i^j on the dummy man's preference and define the dummy man as the top preference for w_i^j. This way, in every stable matching, the dummy man and woman are paired with each other.

This completes the construction for our reduction. We set $a = k \cdot (4|E| + 1) + 2\binom{k}{2}$ where k is the size of the clique. We next show the proof of equivalence for the clique and matching instance.

We use some structural properties of the construction as shown in (MI14).

Lemma 12. *The men optimal matching in the derived instance of matching re-orientation problem matches every man in step 1 and step 2 to his first choice. Consequently, each woman on step 1 and step 2 is matched to her last choice.*

Proof. The proof for this lemma is evident from our construction. □

Lemma 13. *For each vertex* $v_i \in V$*, there exist a rotation* $\rho_i = (m^0_i, w^0_i), (m^1_i, w^1_i), \ldots, (m^{4|E|}_i, w^{4|E|}_i)$.

Proof. We refer reader to the proof of Lemma 5.4 from (MI14). Note the difference that for our reduction we do not need the notion of weights of the rotation. □

Lemma 14. *For every edge* $v_r, v_s \in E$*, where* $r < s$*. The elimination of both* $\rho_r = (m^0_r, w^0_r), (m^1_r, w^1_r), \ldots, (m^{4|E|}_r, w^{4|E|}_r)$ *and* $\rho_s = (m^0_s, w^0_s), (m^1_s, w^1_s), \ldots, (m^{4|E|}_s, w^{4|E|}_s)$ *exposes a rotation* $\rho_{r,s} = (m^1_{r,s}, w^1_{r,s}), (m^q_s, w^{q+1}_s), (m^2_{r,s}, w^2_{r,s}), (m^p_r, w^{p+1}_r)$ *for some* $p, q \in \{0, 1, 2, \ldots, 4|E|\}$.

Proof. We refer the reader to the proof of Lemma 5.5 in (MI14). □

Lemma 15. *The rotation poset for Sex-Equal Stable Matching instance contains exactly one rotation* ρ_i *for every vertex* $v \in V$*, and one rotation* $\rho_{r,s}$ *from every edge* (v_r, v_s) *with* $r < s$*. The predecessor of* $\rho_{r,s}$ *are exactly* ρ_r *and* ρ_s *and there is no predecessor for* ρ_i.

Proof. We refer the reader to the proof of Lemma 5.6 in (MI14). □

Forward direction: Given the yes-instance of clique, let the set $S = \{v_1, v_2, \ldots, v_k\}$ be the set of vertices which participate in clique. We eliminate the rotations ρ_i corresponding to k - vertices in the clique. This re-orients the partners of $k \cdot (4|E| + 1)$ agents since we change the partners of all the copies of a vertex. Consider the set E' of $\binom{k}{2}$ edges shared between the clique vertices. We also eliminate the rotations for each $e = (v_r, v_s) \in E'$. Each of these rotations changes the partner for four men. But notice that for two of these four men agents, the partner was changed when we eliminated the rotations corresponding to clique vertices. Hence, there are only two new men who change the partner for each eliminated edge rotation. By summing over all the k vertex rotations and $\binom{k}{2}$ edge rotations, it can be verified that exactly a agents re-orient their partners.

Reverse direction: From Lemma 6, we know that there is exactly one rotation corresponding to each vertex and one rotation corresponding to every edge. The elimination of vertex rotation ρ_i changes partners for exactly $4|E| + 1$ agents and the elimination of edge rotation changes the partner for four agents. By simple counting argument, it can be verified that no subset of edge rotation eliminations can change partners for $4|E| + 1$ agents. This implies that with $a = k \cdot (4|E| + 1) + 2 \cdot \binom{k}{2}$ one must have eliminated exactly k vertex rotations ρ_i and $\binom{k}{2}$ edge rotations. We also know that the edge rotation $\rho_{r,s}$ is exposed only when ρ_r and ρ_s are eliminated (lemma 5). Let S be the set of k vertices for which we eliminated the rotations. Since we were able to eliminate $\binom{k}{2}$ edge rotations; these edges must have both their endpoints in set S. This implies that the vertices form set S forms a k - clique.

This completes the proof for Theorem 20.

□

5.10 Experimental Results on Restricted Domains

In this section, we study the performance of the Gale-Shapley algorithm on single-peaked preferences. A few studies have been done on the performance of Gale-Shapley for the case of general preferences (CS06), (NR09). For restricted domains, our analysis is novel to the best of our knowledge.

Experimental Setup.

We generate the general and single-peaked profiles as follows: For general profiles, we choose preference order for each candidate uniformly at random from all possible $n!$ ways, where n is the number of women (men). To generate single-peaked profiles, we first fix the single-peaked ordering of the agents along an axis arbitrarily. For each voter, we choose a 'peak' uniformly at random from n possible options. Next, we toss a coin with sides L and R. Depending upon the outcome of the toss, we fill-up subsequent positions in preference order by moving *right* or *le* of the peak on single-peaked axis. When we reach the end of either side of the axis, we place the rest of the agents in the unique order.

Observations and results.

We run Gale-Shapley on the instances with ten agents to 490 agents on each side with a step of 10. In each iteration, we run for a hundred instances of a particular size and record number of proposals made in each such run. We plot the average number of proposals vs the size of the instance in Fig. 5.7.

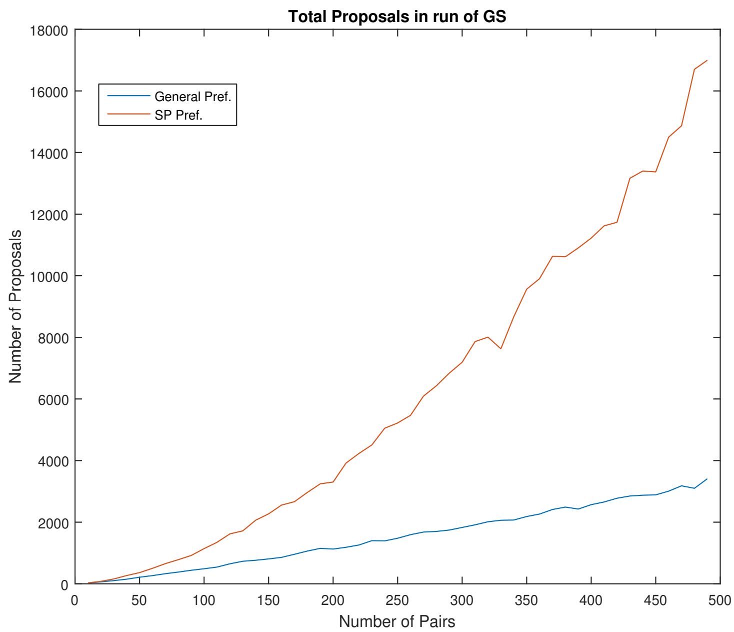


Figure 5.7: Comparison of proposals in Gale-Shapley for general vs restricted domain (SP)

From the observations, it is clear that Gale-Shapley needs a significantly large amount of proposals to construct men optimal stable matching for single-peaked domain compared to the general domain.

Conclusion.

The popular Gale-Shapley algorithm takes a significantly longer time on restricted domains. One solution would be designing a mechanism that takes advantage of structured domains, but this task seems daunting. The decentralized mechanisms such as (RV90) or (Tam93) might work well or might be easier to tweak to account for the domain restriction.

5.11 Concluding Remarks and Open Problems

We studied several problems related to finding stable matchings in the bipartite setting as well as roommates in the context of preferences that might have ties and/or incomplete lists. Our goal was to understand if restricting the preference profile to a structured domain is useful for getting better algorithms for various problems related to stable matchings, especially for problems that are intractable for general profiles. Our overall finding is that most problems, including ones capturing strategic behavior like manipulation, continue to remain hard for restricted domains. One question we would like to deliberate upon is the following: we have been working with the idea that a profile with incomplete and indifferent preferences structured if it admits *any* extension to a structured profile — possibly this is too weak a notion, as evidenced by all the hardness results. Possibly a more stringent notion of extensions will help the design of efficient algorithms.

Open Problems.

In the context of SRTI, does the problem of finding a stable matching remain W[1]-hard when parameterized by $_1$ for profiles that are narcissistic SPSC? Similarly, does the problem remain NP-hard for constant values of $_2$ for profiles that are narcissistic SPSC? The problem is already NP-hard in this setting (BCFN19), so it would be interesting to settle its parameterized complexity.

Apart from this, we believe it would be interesting to resolve the complexity of finding an optimal permutation manipulation that is restricted to respect a given domain (say, single-peaked or single-crossing). Also, one can study the notion of permutation manipulation on mechanisms other than Gale-Shapley.

We would like to point the attention of the reader to a long-standing interesting open problem in the stable marriage setting, the *characterization* of preference profiles which admits *unique* stable matching. There have been several attempts in last two decades which includes the result for single-peaked narcissistic profiles for stable roommates (BIT86), a general condition by Eckhout (Eec00), (Cla06), domain with globally ranked pairs (ALMO07), and recent result by Karpov (Kar19); these results discover several classes of unique matching preference profiles.

Finally, in the case of *(a, b)-Supermatch* problem, we settle the case of *(n,0)-Supermatch* by giving a polynomial time procedure, and it is shown to be NP-complete for the case of *(1, 1)-Supermatch*. The intermediate case of *(a, 0)-Supermatch* where a is an integer input parameter remains open. We suspect that the problem can be shown to be hard with respect parameter a.

Chapter 6

Extension Problems in Stable Matching

6.1 Introduction

We borrow the acronyms and notations from Chapter 2 and Section 5.2.

We define *extension problem* in the context of stable matching. Given a preference profile with incomplete and indifferent lists (a general SRTI instance), we are asked if there exists an extension of this profile a profile with complete orderings for all agents.

In Section 6.3, we consider the problem of finding *special* extension of a preference profile. These *special* preferences have a property that the profile has a unique stable matching. Such profiles are of interest since they have many desirable properties; one such property is resistance from manipulation.

Our problem is inspired from the previous works in qualitative analysis of stable matching problems for the case of incomplete and indifferent preferences. The work by Aziz et al. (ABG+16), (ABF+17) focuses on the questions such as finding the necessary/ possible winner when the preferences in the profile are derived from a certain known collection of preferences. Another direction was considered by Menon and Larson in (ML18); they consider the problem of finding a matching which admits the least number of blocking pairs when summed over all possible complete extensions of given partial profile.

6.2 Preliminaries

Definition of extension problem: Given a partial profile P over the set of agents A, our aim is to find an extension of P' of P such that each $\in P'$ is complete strict ordering over rest of the agents (or agents of opposite sex in case of *stable marriage*).

6.3 Extension to Unique Matching Profile

6.3.1 Extension to general preferences

In this section, we start with the question described above for the general preferences and show the hardness result. We follow this result with two special cases of this problem in the hope to get efficient algorithms.

Theorem 21. *It is* NP*-hard to determine the existence of a complete profile with strict orderings which admits a unique stable matching given an SRTI instance.*

Proof. We show the reduction from Multi-Colored Independent Set (MCIS). Given a partition of vertices in k color classes, MCIS asks for an independent set of size k which contains exactly one vertex from each color class.
Construction. For a given instance of MCIS, let $U_1, U_2, \ldots, U_k$ be the set of vertices from k-color classes. Without loss of generality assume that every color class contains n vertices where n is odd. For every color class, we introduce set U_i of n vertex agents $(u_i^1, u_i^2, \ldots, u_i^n)$ and set S_i of n selector agents $(s_i^1, s_i^2, \ldots, s_i^n)$ for $i \in [k]$. We denote the set of agents by $A = \cup_{i \in [k]}(U_i \cup S_i)$. Let the ordering of the vertex agents for a given color class be $\sigma_i^v : u_i^1 \succ u_i^2 \succ \cdots \succ u_i^n$ for $i \in [k]$. We will now define the preferences of these agents.

For each selector agent s_i^j for $i \in [k], j \in [n]$, the preferences are:

$$s_i^j : u_i^j \succ s_i^{j-1} \succ s_i^{j+1} \succ [rest]$$

and for each vertex agent u_i^j:

$$u_i^j : (u_i^{j-2} \sim u_i^{j-1} \sim u_i^{j+1} \sim u_i^{j+2}) \succ [N(u_i^j)] \succ s_i^j \succ [rest]$$

The indices in the superscript are computed modulo n. By $[N(u_i^j)]$ we denote the neighbors from other color classes of j^{th} vertex from color class i in an arbitrary order. For every agent, the set $[rest]$ at the end of the preference list is unique for each agent and denotes the set of agents apart from the ones appearing before $[rest]$ in some arbitrary preference order. For example consider s_i^j, in this case the ordered set $[rest]$ contains the agents $A \setminus \{u_i^j, s_i^{j-1}, s_i^{j+1}\}$. Note that for each s_i^j we already have a complete strict ordering, and for each u_i^j we have a complete preference list with ties.

We now turn to the equivalence of the two instances.

Forward direction. Given a $k-sized$ multicolored independent set S; if j^{th} vertex from color class i is present in S, then we resolve ties for u_i^j arbitrarily. Now we modify σ_i^v by removing agents corresponding to independent set for every color class and maintaining the rest of the order (let $\sigma_i^{v'}$ denote the modified ordering), each of these ordering contains even number of agents. For each odd ranked agent in $\sigma_i^{v'}$ we place the agent next to it at the top position and do vice versa for even numbered agent i.e.

for $j \in \{1, 3, \ldots, n-2\}$ we place $(j, j+1)$ at each other's top position. Break rest of the ties for u_i^j arbitrarily. We claim that the profile obtained attains a unique stable matching.

In any stable matching, agents in set V_i' are paired in a way that they are matched to their top preference. At this stage, exactly one vertex agent is unmatched from each color class, and this vertex is a part of S. For each such agent u_i^j, the topmost unmatched agent from its preference list is the s_i^j, and u_i^j is at the top position for s_i^j. Hence, in any stable matching, u_i^j is matched to s_i^j. After matching all the vertex agents and s_i^j, a unique stable matching is forced amongst remaining $n-1$ vertex agents. We match agent (s_i^{j+1}, s_i^{j+2}) and continue matching the successive pair of agents until all selector agents from a given color class are matched (note that we consider the indices modulo n). The described matching is the unique stable matching for the instance.

Reverse direction. Given a profile with a unique stable matching, we will construct a MCIS. We'll first show the following lemma.

Lemma 16. *In the stable matching obtained in the reverse direction, there exist at least one selector agent per color class, which is matched to the corresponding vertex agent.*

Proof. (Proof by contradiction) Since no selector agents are matched to the corresponding vertex agent in color class i, all the selector agents are either matched to other selector agent from color class i or to an agent who lies below the third position in its preference list. Since there is an odd number of selectors, we cannot match them among themselves. Let s_i^j be one such agent which is not matched to agent from same color class. Consider the pair (s_i^j, s_i^{j+1}), since s_i^{j+1} is not matched to u_i^{j+1} (as no agent is matched to corresponding vertex agent) and s_i^j, this is a blocking pair for the current matching, which is a contradiction. Hence, in order to match every agent in S_i to one of its top three preferences, we must match at least one s_i^j to u_i^j. □

Using the above lemma, we obtain a $k-\text{sized}$ set S' of agents vertex agents consists of one vertex from each color class such that the agent is matched to the selector agent (if there are more than one agents matched to selector agents, we choose arbitrarily). It is easy to see that for all $u, v \in S'$, u is not adjacent to v. Otherwise, (u, v) will form a blocking pair, since they prefer each other over selector agents. Hence, we have recovered the required Multi-colored Independent set. This completes the proof of equivalence. □

6.3.2 Extension to a special preference profile

We now show a restricted case of preference extension where the profile extended to complete strict orderings admits a unique stable matching, and additionally satisfy the property that for every matching pair, the agents are matched to their top choices. We show an efficient algorithm to check if such a profile exists for the given instance and, in the positive case, find one such profile.

Lemma 17. *There exist an efficient algorithm for the problem of finding an extension for a given SMTI instance, such that the obtained instance admits a unique stable matching where every agent is matched to its top preferences.*

Proof. Our algorithm is a simple application of maximum matching algorithm in bipartite graphs. Let the set of agents be $A := M \cup W$, and an input profile P. For each $m_i \in M$, w_{m_i} be the set of women missing from m_i's partial order. Similarly, we define m_{w_i} for $w_i \in W$. For each $m_i \in M$ ($w_i \in W$), we add w_{m_i} (m_{w_i}) to the set of most preferred agents (i.e. the first equivalence class) in the preference list of m_i (w_i) respectively. We denote the obtained profile by P'.

Next, we construct a bipartite graph with the set of men and women as the two partitions. We add an edge between (m_i, w_j) iff they m_i, w_j are present in the first equivalence class of each other according to P'. At this stage, we run an algorithm to find a maximum matching in this graph. If we obtain a perfect matching, we place the matching partners as a unique top choice for each other and resolve the remaining ties arbitrarily. It is easy to see that the profile obtained using this procedure is a valid profile for the extension problem with all required properties. If the graph does not admit a perfect matching, we return NO (i.e. there does not exist any extension which has the required properties). □

Correctness. We observe that whenever there exists an extension to a profile with complete ordering with the given property that the agents are matched to their top preferences, this matching must be present in the constructed bipartite graph. Hence the maximum matching algorithm will return the required matching. Also, whenever the algorithm returns a perfect matching, the preference profile constructed using the procedure described above has all the required properties.

Though we show the result for the stable marriage case, it is easy to see that the approach can be easily extended to Stable Roommates problem where we will invoke the algorithm for maximum matching in general graphs.

6.3.3 Extension to 1D-Euclidean Profiles

Next we consider a slight general version of the unique extension problem. We now ask for an extension of given SMTI instance to an instance with complete strict orderings where the resultant profile is 1D Euclidean. Note that it has been shown () that the profiles with 1D Euclidean preferences admit a unique stable matching. We show hardness for detecting such extension and present a sharp contrast with the previous case of an efficient algorithm for a slight generalization of the problem.

Theorem 22. *The problem of extension of incomplete partial orderings to a profile with complete orderings which belong to 1D Euclidean domain is* NP*-hard.*

Proof. We show a reduction from Betweenness. The problem is known to be hard from (Opa79). Given a ground set $S = \{s_1, s_2, \dots, s_n\}$ and a set of triples T over S; we have to decide whether it is possible to derive an order over the elements of S such that for

each triple $(s_i, s_j, s_k) \in T$ either $s_i \quad s_j \quad s_k$ or $s_k \quad s_j \quad s_i$ holds.
For each triple $t_i : (s_i, s_j, s_k) \in T$, we introduce two men t_i^1, t_i^2 with preferences,

$$t_i^1 : s_i \quad s_j \quad s_k$$

$$t_i^2 : s_k \quad s_j \quad s_i$$

For all $s_i \in S$, we introduce a woman s_i with all t_i^j for $i \in |T|$, $j \in \{1, 2\}$. This partial profile is given as input to the extension problem. This completes our construction.

Forward direction. Given an ordering over S, let be the 1D euclidean axis over the set of women for preference profile for the extension instance. We complete the partial ordering of each agent to a complete ordering according to . Note that it is always possible to do that for all t_i^1 and t_i^2 since either has $s_i \quad s_j \quad s_k$ or $s_k \quad s_j \quad s_i$. Let $' : t_1 \quad t_2 \quad \ldots \quad t_{|T|}$. We extend the preferences of all women to strict ordering according $'$. This is a valid extension of given partial ordering to a complete strict ordering such that the profile is 1D-Euclidean.

Reverse direction. Let P be the 1D-Euclidean profile with strict complete orderings. It is known that given a profile, one can efficiently check if it is 1D Euclidean and find the corresponding axis (Kno10). We run this algorithm on men's preferences to obtain an axis over women (i.e. an ordering over set S). Since is coherent with the partial ordering of men, the strict order constructed along the axis (i.e. the order where the left-most agent on axis appears at the top and rightmost agent appears at the bottom) trivially satisfies either $s_i \quad s_j \quad s_k$ or $s_k \quad s_j \quad s_i$. Hence we obtain the required ordering for Betweenness.

This completes the proof of equivalence. Note that the problem is hard even for the Stable Marriage case.

□

Part III

Fair Division

Chapter 7

Connected Fair Division on a Path with Envyfreeness

7.1 Introduction

We refer the reader to subsection 1.1.3 for a gentle introduction to *fair division*. In this work, we consider *connected fair division* of goods on a path. We consider the model where we have n agents and m goods, such that the agents have distinct valuations (different preferences) for the goods, and the valuations are *binary* where each agent either approve a good or disapprove it. In this Chapter, we study Envyfree allocations along with other desirable properties.

Consider a team of researchers as agents, and time slots on a shared high-performance computer as goods. Every researcher has its preferences of time slots. Consider an allocation which allocated an agent a time slot for a certain time, followed by some other researcher and then again, the time slot for some duration to the first researcher. Clearly, such allocation will not be convenient for researchers at all who will rather prefer a contiguous slot of access to the computer. Similarly, if the agents are nations, and resource is a coastal area of a sea. Again, in this case, each country would prefer a contiguous piece of land (CDP13). Hence, it is essential to obtain a contiguous allocation for certain applications.

In the context of *fair division*, *only* considering fairness notions such as *Envyfreeness* or *Equitability* might not always be a prudent model. Since often, there exists an easy way to achieve these with trivial allocations. Consider an example of empty allocation, such an allocation is both *Envyfree* and *Equitable* but not meaningful. Hence, we complement fairness properties with *e ciency* notions. Completeness, Proportionality, and Pareto Optimality are widely used efficiency notions in the fair division. Completeness simply means allocating *all* the goods amongst the agents in the instance. Proportionality demands each agent to get an allocation, which is at least $\frac{1}{n}$ fraction of goods according to its utility in the instance. At last, Pareto Optimality implies the non-existence of allocations in which every agent gets at least as much utility as in current allocation, and there exists at least one agent for which the utility is strictly higher. For the following study, we consider *Envyfreeness* along with Completeness, Pareto Optimality and Non-wastefulness (where Non-wastefulness implies allocations such every agent receives only those goods which it values). We study the problems of existence and computation of these allocations.

In Chapter 8, we consider same set of questions with *Equitability* as our fairness notion.

7.1.1 Summary of Results

Restrictions #	Existence			Computation		
	EF1 + Comp	EF1 + PO	EF1 + NW	EF1 + Comp	EF1 + PO	EF1 + NW
Binary	Yes (Theorem 24)	No (Prop. 5)	No (Example 2)	?	NP-hard (Prop. 5)	NP-hard (Theorem 25)
Binary interval	Yes (Theorem 24)	No (Prop. 5)	No (Exmample 2)	?	?	?
Binary k-interval	Yes (Theorem 29)	Yes (Theorem 29)	Yes (Theorem 29)	P (Theorem 29)	P (Theorem 29)	P (Theorem 29)
Binary extremal	Yes (Theorem 28)	Yes (Theorem 28)	Yes (Theorem 28)	P (Theorem 28)	P (Theorem 28)	P (Theorem 28)
Binary left-extremal	Yes (Theorem 27)	Yes (Theorem 27)	Yes (Theorem 27)	P (Theorem 27)	P (Theorem 27)	P (Theorem 27)

Table 7.1: Summary of results for Connected Fair Division EF1. For every fairness/efficiency requirement (columns) and preference restriction (rows), the entries in the *Existence* section pertain to answers to the question "Does a desired allocation always exist?" and those in the *Computation* section pertain to the computational complexity of determining whether a desired allocation exists. The entries marked with P pertain to polynomial-time algorithms that also return the desired allocation (whenever it exists).

7.2 Preliminaries

We will follow the notation used by (BCE+17) and (BCF+19).

The Model. Let $[n] = \{1, 2, \ldots, n\}$ be a set of $n \in \mathbb{N}$ *agents*, and $G = (V, E)$ be an undirected graph. Each vertex $v \in V$ of the graph G corresponds to a *good* (or an *item*) with $m := |V|$ goods overall. A set of goods $S \subseteq V$ is said to be *connected* if it induces a connected subgraph of G. We let $C(V) \subseteq 2^V$ denote the set of all connected subsets of V. Each $S \in C(V)$ is called a (connected) *bundle*.

A (connected) *allocation* $A : [n] \to C(V)$ assigns to each agent $i \in [n]$ a connected bundle $A(i) \in C(V)$ such that no good is assigned to more than one agent. For simplicity, we denote an allocation as an ordered tuple $A = (A_1, A_2, \ldots, A_n)$, where $A_i := A(i)$.

The preferences of agent $i \in [n]$ over the connected bundles are specified by a *valuation function* $u_i : C(V) \to \mathbb{N} \cup \{0\}$. We will assume that the valuation functions are *additive*, i.e., for each $i \in [n]$ and each $S \in C(V)$, $u_i(S) := \sum_{v \in S} u_i(\{v\})$, where $u_i(\emptyset) := 0$. An n-tuple of valuation functions $U = \{u_1, \ldots, u_n\}$ is called a *valuation profile*. We will extensively focus on *binary valuations*, where for every good $v \in V$ and every agent $i \in [n]$, we have $u_i(\{v\}) \in \{0, 1\}$.

Unless stated otherwise, the graph $G = (V, E)$ will be assumed to be a *path*. In this case, we will denote the set of goods (or vertices) by $\{1, 2, \dots, m\}$, where $\{j, j+1\} \in E$ for every $j \in [m-1]$. For simplicity, we will write $u_{i,j}$ instead of $u_i(\{j\})$.

Definition 15 (Connected Fair Division)**.** *An instance of the* Connected Fair Division *problem is denoted by the tuple* $I = \langle G, [n], U \rangle$. *The goal is to determine whether* I *admits a* connected *allocation satisfying the desired fairness notion. In this work, we will focus on two notions of fairness:* Envy-freeness *(16) and* equitability *(19). Fairness notions are often coupled with efficiency (otherwise, an empty allocation also counts as "fair"), and we will consider three such notions:* completeness, Pareto optimality, *and* non-wastefulness *(17).*

Notice that if G is a clique, then Connected Fair Division becomes equivalent to the standard fair division problem with indivisible items without the connectedness constraint.

Definition 16 (Envy-freeness and its Relaxations)**.** *An allocation* $A = (A_1, A_2, \dots, A_n)$ *is said to be (a) envy-free* (EF) *if for any pair of agents* $i, k \in [n]$, *we have* $u_i(A_i) \geq u_i(A_k)$; *(b)* ε*-envy-free* (ε-EF) *if for any pair of agents* $i, k \in [n]$, *we have* $u_i(A_i) \geq u_i(A_k) - \varepsilon$; *(c) envy-free up to one good* (EF1) *if for any pair of agents* $i, k \in [n]$, *either* $u_i(A_i) \geq u_i(A_k)$ *or there exists some good* $v \in A_k$ *such that* $u_i(A_i) \geq u_i(A_k \setminus \{v\})$; *and (d) (c) envy-free up to one outer good* (EF1-out) *if for any pair of agents* $i, k \in [n]$, *either* $u_i(A_i) \geq u_i(A_k)$ *or there exists some good* $v \in A_k$ *such that* $A_k \setminus \{v\}$ *is connected and* $u_i(A_i) \geq u_i(A_k \setminus \{v\})$.

It follows from the definitions that EFo1 $\Rightarrow$ EF1. The notions of EF1 and EFo1 are due to (Bud11) and (BCF⁺19) respectively.

Definition 17 (Efficiency Notions)**.** *A (connected) allocation* $A = (A_1, A_2, \dots, A_n)$ *is (a) complete* (Comp) *if no good is left unallocated by* A, *i.e., for any good* v, *there exists some agent* $i \in [n]$ *such that* $v \in A_i$; *(b) Pareto optimal* (PO) *if for no other connected allocation* B, *we have* $u_i(B_i) \geq u_i(A_i)$ *for every agent* $i \in [n]$, *with the inequality being strict for at least one agent; and (c) non-wasteful* (NW) *if it is complete and each good is assigned to an agent that has a non-zero value for it, i.e., for any good* v, *there exists some agent* $i \in [n]$ *such that* $v \in A_i$ *and* $u_i(\{v\}) > 0$.

An allocation that is not complete is called a *partial* allocation. A non-wasteful allocation is, by definition, complete. A Pareto optimal allocation is, without loss of generality, also complete.[1] In general, non-wastefulness and Pareto optimality are incomparable notions, even when G is a path.[2] However, for binary valuations, NW $\Rightarrow$ PO $\Rightarrow$ Comp.

Definition 18 (Preference Restrictions)**.** *We study various preference restrictions for the special case of* Connected Fair Division *when* G *is a path.*[3] *Recall that in this setting,*

[1] This is because a partial allocation that is PO can be extended to a complete and PO allocation.

[2] Indeed, imagine three goods v_1, v_2, v_3 on a path and two agents $\{1, 2\}$ whose valuations are $u_{1,1} = 1$, $u_{1,2} = 10$, $u_{1,3} = 0$, $u_{2,1} = 10$, $u_{2,2} = 1$, and $u_{2,3} = 1$. The allocation $A := (\{v_1\}, \{v_2, v_3\})$ is non-wasteful, but is Pareto dominated by the allocation $B := (\{v_2, v_3\}, \{v_1\})$. Also, B is Pareto optimal but not non-wasteful because it assigns v_3 to agent 1.

[3] Some of these preference restrictions are inspired from similar assumptions in the theory of voting (EL15).

the goods are indexed from left to right by $1, 2, \ldots, m$. *We say that the valuation functions are (a)* ***binary*** *if for each agent* $i \in [n]$ *and each good* $j \in [m]$, $u_{i,j} \in \{0, 1\}$; *(b)* ***binary interval*** *if for each agent* $i \in [n]$, *there exist* $\ell_i \in [m]$ *and* $r_i \in [m]$ *(with* $\ell_i \leq r_i$*) such that* $u_{i,j} = 1$ *for all* $j \in \{\ell_i, \ell_{i+1}, \ldots, r_i\}$ *and* 0 *otherwise; (c)* ***binary*** k***-interval*** *if the valuations are binary interval and each agent values exactly* k *goods; (d)* ***binary left-extremal*** *if for each agent* $i \in [n]$, *there exists* $\ell_i \in [m]$ *such that* $u_{i,j} = 1$ *for all* $j \in \{1, \ldots, \ell_i\}$ *and* 0 *otherwise; and (e)* ***binary extremal*** *if for each agent* $i \in [n]$, *there exist* $L_i, R_i \subseteq [m]$ *such that (1)* L_i *is either empty or there exists* $\ell_i \in [m]$ *such that* $L_i = \{1, \ldots, \ell_i\}$, *(2)* R_i *is either empty or there exists* $r_i \in [m]$ *(with* $\ell_i \leq r_i$*) such that* $R_i = \{r_i, \ldots, m\}$, *and (3)* $u_{i,j} = 1$ *for all* $j \in L_i \cup R_i$ *and* 0 *otherwise.*

We will assume, without loss of generality, that each agent has non-zero valuation for at least one good, and that each good is valued by at least one agent.

Organization of the Chapter.

- **Existence:** In Section 7.4, we discuss about the existence of EF1 allocations with various efficiency guarantees. We show that EF1+complete allocations always exists (Section 7.4.1). For EF1+PO we point to a counter example by (IP19), and for EF1+NW we show a counter example (Section 7.4.3).
- **Hardness Results:** In Section 7.5, we show hardness for checking existence of EF1+NW allocations, and for EF1+complete on a collection of paths.
- **Restricted Domains:** In Section 7.6, we show polynomial time algorithms for computing EQ1+NW allocations for *Binary left-extremal*, *Binary extremal*, and *Binary k-interval* valuations. We also show that a similar set of result holds for EF1+PO using an implication that for extremal and fixed length interval valuations, EF1+PO $\implies$ EF1+NW.

7.3 Results for Efficiency Notions

In this section, we show hardness for finding the existence of Non-wasteful allocations, even without considering any fairness notion. Later in section 7.6 we show that finding a Non-wasteful EF1 allocation for special restrictions on valuations does admit efficient algorithms.

Theorem 23. *Determining the existence of a connected non-wasteful allocation* (NW) *is* NP*-complete even when* G *is a path and the valuations are binary.*

Proof. Our construction is similar to the one used by (BCL18) in the context of analyzing connected fair division for chores. Specifically, we will show a reduction from the NP-complete problem 2P2N-3-SAT, which is a restriction of 3-SAT where each variable occurs four times, twice as a positive literal and twice as a negative literal (BKS04). An instance of 2P2N-3-SAT consists of s boolean variables $x_1, \ldots, x_s$ and t clauses $c_1, \ldots, c_t$. Each variable x_i is associated with two positive and two negative literals x_i^1, x_i^2, and $\overline{x}_i^1$, $\overline{x}_i^2$ respectively, numbered according to the order of appearance of x_i in the clauses. Notice that $3t = 4s$.

Given an instance of 2P2N-3-SAT, we construct a fair division instance as follows: There are $t + 5s$ goods, consisting of t *clause goods* $c_1, \ldots, c_t$; s *dummy goods* $d_1, \ldots, d_s$, and $4s$ *literal goods* $\bigcup_{i=1}^{s}\{x_i^1, x_i^2, \overline{x}_i^1, \overline{x}_i^2\}$. The graph $G = (V, E)$ is a path defined as follows (see 7.1):

$$V := \{c_1, \ldots, c_t\} \cup \{x_i^1, x_i^2, d_i, \overline{x}_i^1, \overline{x}_i^2\}_{i=1}^{s}, \text{ and}$$

$$E := \{\cup_{i=1}^{t-1}(c_i, c_{i+1})\} \cup \{(c_t, x_1^1)\} \cup \{\cup_{i=1}^{s}(x_i^1, x_i^2) \cup (x_i^2, d_i) \cup (d_i, \overline{x}_i^1) \cup (\overline{x}_i^1, \overline{x}_i^2)\} \cup \{\cup_{i=1}^{s-1}(\overline{x}_i^2, x_{i+1}^1)\}.$$

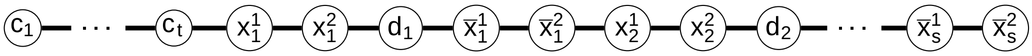


Figure 7.1: The path G constructed in the proof of Theorem 23.

There are $6s$ agents, consisting of $4s$ *literal agents* $\{z_i^1, z_i^2, \overline{z}_i^1, \overline{z}_i^2\}_{i=1}^{s}$; s *variable agents* $y_1, \ldots, y_s$, and s *dummy agents* $w_1, \ldots, w_s$.

The preferences of the agents are defined as follows: Each dummy agent w_i approves all s dummy goods $d_1, \ldots, d_s$. For every $i \in \{1, \ldots, s\}$ and $k \in \{1, 2\}$, the literal agent z_i^k (respectively, $\overline{z}_i^k$) approves two goods, namely the literal good x_i^k (respectively, $\overline{x}_i^k$) and the unique clause good c_j such that $x_i^k \in c_j$ (respectively, $\overline{x}_i^k \in c_j$). Each variable agent y_i (corresponding to the variable x_i) approves four literal goods $x_i^1, x_i^2, \overline{x}_i^1, \overline{x}_i^2$. All other valuations are set to zero.

($\Rightarrow$) Suppose there exists a truth assignment for the 2P2N-3-SAT instance. Then, the desired allocation can be constructed as follows: Assign each dummy good d_i to the corresponding dummy agent w_i. For each $i \in \{1, \ldots, s\}$, if the variable x_i is true, then assign the literal goods x_i^1, x_i^2 to the variable agent y_i, and the other two literal goods $\overline{x}_i^1, \overline{x}_i^2$ to the corresponding literal agents $\overline{z}_i^1, \overline{z}_i^2$. Otherwise, if x_i is false, then assign $\overline{x}_i^1, \overline{x}_i^2$ to y_i, and x_i^1, x_i^2 to z_i^1, z_i^2. Finally, for each $j \in \{1, \ldots, t\}$, if clause c_j is such that $c_j = \ell_1 \vee \ell_2 \vee \ell_3$, then assign the clause good c_j to (any one of) ℓ_k if ℓ_k is true. Notice that the above allocation assigns each good to an agent that approves it, and is therefore non-wasteful. Furthermore, if a literal agent z_i^k (or $\overline{z}_i^k$) gets a clause good, then it does not get any of the literal goods. Therefore, the allocation is also connected.

($\Leftarrow$) Now suppose that there exists a connected and non-wasteful allocation A. Notice that A must assign each dummy good to one of the dummy agents (due to non-wastefulness), and no two dummy goods can be assigned to the same dummy agent (otherwise, due to connectedness, all intermediate goods will also be assigned to the dummy agent, thus violating non-wastefulness). Therefore, without loss of generality, A assigns the dummy good d_i to w_i. Next, observe that if for some $i \in \{1, \ldots, s\}$ and some $k, k' \in \{1, 2\}$, A assigns clause goods to both z_i^k and $\overline{z}_i^{k'}$, then the corresponding literal goods x_i^k and $\overline{x}_i^{k'}$—lying on either side of the dummy good d_i—will need to be assigned to the variable agent y_i, thus violating connectedness. Therefore, if a clause good is assigned to a literal agent z_i^k (respectively, $\overline{z}_i^k$) for some $k \in \{1, 2\}$, then neither $\overline{z}_i^1$ nor $\overline{z}_i^2$ (respectively, neither z_i^1 nor z_i^2) is assigned any clause good. In other words, the assignment of clause goods cannot simultaneously trigger a positive and a negative literal associated with any variable. We can now infer the following truth assignment:

For every $i \in \{1, \ldots, s\}$, set x_i to be true if either z_i^1 or z_i^2 is assigned a clause good, and false otherwise. □

7.4 Existence of EF1 allocations

7.4.1 EF1 and Completeness

For the case of two agents and G = path, (SHS18) show, via a discrete version of the "moving knife" procedure, that a connected allocation that is EF1 and Complete (but possibly wasteful) always exists and can be efficiently computed.

Proposition 3. *[(SHS18); eorem 4.1]EFoneTwoagents When there are $n = 2$ agents and G is a path, there always exists a connected allocation that is envy-free up to one good* (EF1) *and complete* (Comp). *Moreover, such an allocation can be computed in polynomial time.*

(Suk19) proves a similar result for a general number of agents but with a weaker fairness guarantee ($2u_{max}$-EF instead of EF1). Unlike Proposition 3, this result does not provide an efficient algorithm for finding the corresponding allocation.

Proposition 4. *[(Suk19)]EFmargNagents When G is a path, there always exists a connected and complete allocation such that agent i has envy at most $2u_{i,max}$ towards any other agent, where $u_{i,max} := \max_{j \in [m]} u_{i,j}$. In particular, there exists a connected $2u_{max}$-*EF *and complete allocation, where $u_{max} := \max_{i \in [n]} u_{i,max}$.*

Derived Cake-cutting Instance When G is path, we will often find it convenient to reduce a discrete instance I of Connected Fair Division to its continuous analogue, which we call a *derived cake-cu ing instance* I^D. The derived instance $I^D = \langle [0, m], [n], U^D \rangle$ consists of a divisible resource (or a *cake*) denoted by $[0, m]$, the same set of n agents, and a set of derived valuation functions over $[0, m]$ defined as follows: If $u_{i,j} = b$ in I, then $u_i([j-1, j]) = b$ in I^D. That is, agent i values good j in I at b if and only if it has a piecewise constant valuation of b for the interval $[j-1, j]$ in I^D. A connected allocation in I^D refers to a division of $[0, m]$ into n contiguous subintervals, one for each agent.

The proof of proposition 4 by (Suk19) is via reduction to *cake cu ing*. Briefly, consider the derived cake-cutting instance I^D for the given instance I of Connected Fair Division. It is known that for piecewise constant valuations, a connected, envy-free and complete allocation of a cake always exists (Su99). Such an allocation (say, A) in I^D could, in general, correspond to a fractional allocation of the goods in I and must therefore be rounded. The rounding scheme in (Suk19) assigns good j to agent i if the integral point j is assigned to agent i by the allocation A in I^D. In this process, an agent could lose out on a good j at the right extreme of its fractional allocation (even when it is assigned the interval $[j-1, j)$ in I^D but not the right extreme point $\{j\}$). Similarly, some other agent could get a good j at its left extreme (simply by having the point $\{j\}$ allocated to it in I^D). Thus, the rounded allocation need not be envy-free. However, the envy experienced by agent i is bounded by *less than or equal to* $2u_{i,max}$. In lemma 18, we describe a different rounding scheme that strengthens the envy bound in proposition 4 from a weak to a strict inequality.

Lemma 18 (**Rounding scheme**). *Let A be a connected, complete and envy-free allocation in the derived cake-cutting instance I^D. Then, one can compute in polynomial time an integral allocation A' in the original instance I that is connected and complete such that agent i has envy strictly less than $2u_{i,max}$ towards any other agent, where $u_{i,max} := \max_{j \in V} u_{i,j}$.*

Proof. Given a (possibly fractional) allocation A, we can compute an integral allocation A' via the following rounding scheme: Each fractional cut is rounded to its nearest integer value, with the additional condition that for any integer p, a cut at $p + 0.5$ is consistently rounded either to p or to $p + 1$.

It is easy to see that A' is connected and complete in I. To see why the envy guarantee holds, note that under the rounding step, an agent could "gain" at most 0.5 fraction of an interval to its right and at most 0.5 fraction of an interval to its left, with at least one of the gains being strictly less than 0.5. A similar observation holds for an agent losing its extreme goods. Therefore, the envy of agent i towards any other agent must be *strictly less than* twice its maximum value for any good (namely, $2u_{i,max}$), as desired. □

When instantiated for *binary* valuations, Suksompong's proof of Proposition 4 shows the existence of a 2-EF allocation. Lemma 18 helps us improve the fairness guarantee from 2-EF to EF1 Theorem 24.

Theorem 24 (**Existence of EF1 for binary valuations**). *Given a* Connected Fair Division *instance where G is a path and valuations are binary, a connected allocation that is envy-free up to one good* (EF1) *and complete* (Comp) *always exists.*

Proof. As mentioned above, our proof is similar to that of (Suk19) in that we also perform a reduction to cake cutting followed by a rounding step. The existence of a connected, complete and envy-free allocation (say, A) in the derived cake-cutting instance is guaranteed by a result of (Su99). Such a division of the cake could, in general, correspond to a fractional allocation of the goods in the Connected Fair Division instance I. By using the rounding scheme of lemma 18, we obtain a connected and complete allocation (say, A') in which agent i has envy strictly less than $2u_{i,max}$ towards any other agent. For binary valuations, $u_{i,max} = 1$; thus $u_i(A'_k) - u_i(A'_i) < 2$ for any $i, k \in [n]$. By integrality of valuations, this implies that $u_i(A'_k) - u_i(A'_i) \leq 1$. Hence, the rounded allocation is 1-EF. The claim now follows by observing that for binary valuations, an allocation is 1-EF if and only if it is EF1. □

7.4.2 EF1 and Pareto Optimality

(IP19) study the problem of finding a connected allocation of a graph under the Pareto optimality (PO) constraint. One of their results is that an allocation that is simultaneously EF1 and PO allocation need not exist when G is a path, and finding one can be computationally hard. The non-existence result of (IP19) is in stark contrast to the setting where G is a clique (i.e., the standard model of fair division without the connectedness constraints), where an EF1 and PO allocation is guaranteed to exist (CKM⁺16).

Proposition 5. *[IP19] When G is a path, there need not exist a connected allocation that is both* EF1 *and* PO*, even for binary valuations. Moreover, it is* NP*-hard to determine if such an allocation exists, even for binary valuations.*

7.4.3 EF1 and Non-Wastefulness

Example 2 (**Non-existence of EF1 and NW with binary interval valuations**). *When* G *is a path, there need not exist a connected allocation that is both* EF1 *and* NW, *even for binary interval valuations, as the following example demonstrates: Consider a* CONNECTED FAIR DIVISION *instance with* $n = 2$ *agents and* m *goods* $g_1, \dots, g_m$ *on a path* $G = (V, E)$ *such that* $(g_i, g_{i+1}) \in E$ *for every* $i \in [m-1]$. *Agent* 1 *values all the* m *goods at* 1. *Agent* 2 *values* $g_2, \dots, g_{m-1}$ *(i.e., all but the extreme goods) at* 1 *and the rest at* 0. *Any non-wasteful allocation must assign the extreme goods to agent* 1. *By connectedness, all the intermediate goods must also be assigned to agent* 1, *resulting in arbitrarily large envy from agent* 2.

7.5 Hardness Result for Envyfreeness

7.5.1 EF1 and Non-wasteful Allocation

We now present a hardness for determining the existence of non-wasteful EF1 allocations on a path.

Theorem 25. *When* G *is a path, it is* NP*-hard to determine whether there exists a connected allocation that is both* EF1 *and* NW, *even for binary valuations.*

Proof. Our reduction is similar to the reduction by (IP19) (Theorem 3). We show the reduction from EXACT-3-COVER (X3C). Let $X = \{x_1, x_2, \dots, x_{3r}\}$ be the universal set, $\mathcal{S}$ be the set of 3-element subsets of X and $|\mathcal{S}| = s$. The given instance is 'YES' instance of X3C if and only if there exist a set $\mathcal{S}' \subseteq \mathcal{S}$ s.t. $|\mathcal{S}'| = r$ and $\bigcup_{S \in \mathcal{S}'} S = X$. Consider an instance $(X, \mathcal{S})$ of X3C. For each $S \in \mathcal{S}$ we denote three elements in S by (x_S^1, x_S^2, x_S^3). We construct following instance I of CFD using this.

Goods: For each $S_i \in \mathcal{S}$ we introduce three goods $x_{S_i}^1, x_{S_i}^2, x_{S_i}^3$ (denoted by x_{S_i}) corresponding to three elements in the set and for each $k \in [r]$, we introduce set of three goods (d_k^1, d_k^2, d_k^3) (We will call these two types of three good chunks as x_{S_i} and d_k). Apart from this we introduce $s + r - 1$ separator goods t_i's.

Agents: For each $S_i \in \mathcal{S}'$ we create an agent i_{S_i} who only approves the goods $x_{S_i}^1, x_{S_i}^2, x_{S_i}^3$, and each d_k^h for $k \in [r]$ and $h \in [3]$; for every $x \in X$ we create an agent i_x which only approves x_S^h such that $x_S^h = x$. We also introduce $s + r - 1$ separator agents i_t each of which approve a unique separator good.

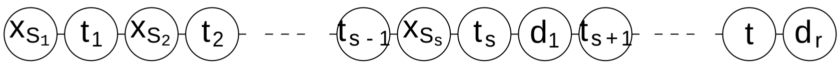


Figure 7.2: Sequence of goods on path for Theorem 25

Note that by construction, every agent i_S can get at most three goods; every agent i_x can get at most one good and every agent i_t can get at most one good in any connected allocation.

Forward direction: Let the set $\mathcal{S}' \subseteq \mathcal{S}$ be the exact cover of X ($|\mathcal{S}'| = r$).

- Each agent i_S such that $S \in S'$, we allocate one of d_k^1, d_k^2, d_k^3 arbitrarily for $k \in [r]$.
- Each agent i_{S_i} (s.t. $S \in S \setminus S'$) we allocate each one of them the set of three goods $x_{S_i}^1, x_{S_i}^2, x_{S_i}^3$ corresponding to set S_i.
- Each agent i_x is allocated the good x_S^h such that $x_S^h = x$
- Each agent t_i is allocated the unique separator good it approves.

According to the construction it is easy to see that this is a complete, connected, EF1 allocation (since all agents get the maximum connected chunk that they can get).

Reverse direction: Consider that there exist a complete, connected, non-wasteful EF1 allocation. Notice that in any complete, non-wasteful allocation each separator good has to go to a unique separator agent (since these goods are approved only by unique agents). At this point we remain with $s + r$ three good chunks corresponding to (x_{S_i}) and (d_k), and s unallocated agents i_{S_i} and $3r$ unallocated agents i_x. Hence in any **complete** allocation we must allocate three goods to each i_S and one good to each i_x in order to follow the connectivity constraint according to construction.

Assume that the given instance (X, S) of X3C is a no instance. This implies that $|S'| > r$ such that $S' \subseteq S$ and $\cup_{S \in S'} S = X$. Hence, for any complete allocation, after allocating one good to each agent i_x we remain with more than $s - r + 1$ partially or completely unallocated sets of goods from x_{S_i}. These combined with r sets d_k we have strictly more than s distinct sets of partially or completely unallocated goods from $x_{S_i} \cup d_k$. We know that each agent i_{S_i} can be allocated the goods from at most one of these set of goods. Since we have at most s of these agents (i_{S_i}), by pigeon-hole argument, there will exist some set of goods which are unallocated. This is a contradiction to our assumption of complete allocation.

This completes the proof.

□

7.5.2 EF1 and complete allocation on a collection of Paths

The existence of a complete-EF1 allocation is known for binary valuations on a path is known is from Theorem 24. In this section, we show that if we relax the underlying graph of goods to a collection of paths instead of one single path, the existence does not hold anymore. An easy counter-example for existence is the case when the number of agents in the instance is strictly less than the number of paths of goods. Such an instance cannot admit complete-connected allocation. We next show that deciding the existence of complete-EF1 allocations which are connected is hard.

Theorem 26. *When G is a collection of paths, it is NP-hard to determine whether there exist a connected allocation that is both complete and EF1, even for binary valuations.*

Proof. We show the reduction from a variant of Exact-3-Cover (X3C) problem. X3C is known to be hard even when each element in the universe appears exactly three times. We will start from this bounded frequency variant of X3C. Let $X = \{x_1, x_2, \ldots, x_{3r}\}$ be

the universal set, $\mathcal{S}$ be the set of 3-element subsets of X and $|\mathcal{S}| = s = 3r$ (due to bounded frequency). Consider an instance $(X, \mathcal{S})$ be the X3C instance. Let $\mathcal{S}'$ be the set containing two copies of all the sets $S \in \mathcal{S}$. $|\mathcal{S}'| = 6r$ and each element in the universe now appears exactly six times. Notice that changing the set $\mathcal{S}$ with $\mathcal{S}'$ does not alter the fate of given X3C instance. For each $S \in \mathcal{S}'$ we denote three elements in S by (x_S^1, x_S^2, x_S^3). We construct following instance I using this.
Goods: For each $S_i \in \mathcal{S}'$ we introduce $3k$ goods with k (for $k > 8r$) copies of each $x_{S_i}^1, x_{S_i}^2, x_{S_i}^3$ (denoted by x_{S_i}) corresponding to three elements in the set.
Agents: For every $x \in X$ we introduce an element agent i_x which only approves k-copies of x_S^h such $x_S^h = x$ for $h \in [3]$. We also introduce $|\mathcal{S}'| - r$ i.e. $5r$ dummy agents which approves all the goods.

This completes the construction of I'. By construction, every agent can get at most $3k$ goods.

Forward direction. Let $\mathcal{S}'' \subseteq \mathcal{S}'$ be the exact cover of X.

- Each agent i_x is allocated a set of k goods such that $S \in \mathcal{S}''$ and $x_S^h = x$ for $h \in [3]$.
- Allocate each of the remaining $5r$ paths of $3k$ goods to one dummy agent each arbitrarily.

As each dummy agent receives exactly $3k$ goods and at most k goods valued by any agent i_x, the allocation is complete, connected and EF1.

Reverse Direction. Let σ be a complete, connected and EF1 allocation for I. Let a be the maximum number of goods allocated to any dummy agent under σ. Since σ is EF1, all the dummy agents either receive a or $a - 1$ goods and each i_x can be allocated at most a goods for some natural number a. Since the allocation is complete,

$$a > \frac{6r \cdot 3k}{(5r + 3r)} \Rightarrow a > 2k$$

This implies that no two dummy agents can share a single connected piece of length $3k$. Let $a < 3k$. Such an allocation will create $5r$ partially allocated paths. As there are at most $3r$ element agents, the partial allocation cannot be completed. Combining these two facts, we get $a = 3r$. We now show that all dummy agents will be allocated an entire connected piece of $3k$ goods. Assume this is not the case, d_i is allocated $3k - 1$ goods under σ. The remaining good in the connected piece has to be allocated to an element agent (say i_x). Consider the following two cases:

- i_x approves the allocated good: This implies i_x approves exactly k goods in from the connected piece under consideration. $k - 1$ of these approved goods are allocated to d_i, which contracts the assumption that σ is EF1.
- i_x does not approve the allocated good: Since $k > 8r$, there exist at least one agent who receives more than one good approved by i_x, and i_x does not receive any good it approved. We obtain a similar contradiction as in the previous case.

Let $S \subseteq S'$ be the subset of paths allocated among element agents i_x. From the above argument, it is clear that each i_x should receive at least one good it approves. Hence, set S contains at least one copy of each i_x. As $|S| = r$, it contains exactly one copy of each element. Hence, we can recover the required exact cover from any allocation .

□

Notice that the above reduction also works for EF-k for any constant $k \in \mathbb{N}$.

7.6 Efficient Algorithms for Restricted Preference Domains

In this section, we show the existence and efficient computation of Non-wasteful EF1 allocation for the case of restricted classes of valuations. We refer the reader to definition 18 to recall these classes.

7.6.1 An Algorithm for Binary Left-Extremal Valuations

This section presents an algorithm (Algorithm 2) for computing an envy-free up to one good (EF1) and non-wasteful (NW) allocation for binary left-extremal valuations.

Description of the algorithm: Recall that for binary left-extremal valuations, each agent $i \in [n]$ is associated with a segment $L_i := \{1, \ldots, \ell_i\}$ such that $u_{i,j} = 1$ for all $j \in L_i$ and 0 otherwise. Without loss of generality, reindex the agents such that $\ell_1 \leq \ell_2 \ldots \leq \ell_n$, or, equivalently, $L_1 \subseteq L_2 \ldots \subseteq L_n$.

Algorithm 2 works in two phases. In Phase 1, the algorithm works with the derived cake-cutting instance I^D of the instance I. Recall that the derived instance comprises of a divisible resource $[0, m]$ over which the agents have piecewise constant valuations. At iteration t of Phase 1, let N^t denote the set of currently available agents, and ℓ_i^t denote the length of segment of agent i over the currently available resource. For each agent $i \in N^t$, let $N_i^t := \{k \in N^t : \ell_k^t \leq \ell_i^t\}$ denote the set of currently available agents whose segments (with respect to the currently available resource) are contained within that of agent i. Additionally, let $n_i^t := |N_i^t|$. The algorithm computes the ratio $\alpha_i^t := \ell_i^t / n_i^t$ for each agent i. Let $i_t \in \text{Arg}\min_{i \in N^t} \alpha_i^t$ denote the agent with the smallest ratio (Line 8). The algorithm allocates part of the resource, namely [start, start + $\ell_{i_t}^t$], by evenly dividing it (in contiguous pieces) among the $n_{i_t}^t$ agents in the set $N_{i_t}^t$. The agents in $N_{i_t}^t$ are then removed from further consideration, and the algorithm re-enters Phase 1 with the remaining agents $N^t \setminus N_{i_t}^t$ and the leftover resource [start + $\ell_{i_t}^t$, m].

At the end of Phase 1, the algorithm has constructed an allocation A of the resource $[0, m]$ in the derived instance I^D. Such an allocation could, in general, correspond to a fractional allocation in I. Therefore, in Phase 2, the algorithm uses the rounding scheme of lemma 18: Each fractional cut is rounded to its nearest integer value, with the additional condition that for any integer p, a cut at $p + 0.5$ is consistently rounded either to p or to $p + 1$.

Algorithm 2: Algorithm for binary left-extremal valuations

Input: An instance $I = \langle G, [n], \mathcal{U} \rangle$ with binary left-extremal valuations and a path G.
Output: An allocation A.

1 $A \leftarrow \{\emptyset, \emptyset, \ldots, \emptyset\}$ ▷ Initialize the current allocation
2 $t \leftarrow 0$ ▷ Initialize the iteration index
3 $N^t \leftarrow [n]$ ▷ Initialize the set of remaining agents
4 $\ell_i^t \leftarrow \ell_i$ for all $i \in [n]$ ▷ Initialize the lengths of segments
5 start $\leftarrow 0$ ▷ Initialize the starting point of the cake

▷ Phase 1: Compute fractional allocation of the derived cake instance

6 **while** $N^t \neq \{\emptyset\}$ **do**
7 For each $i \in N^t$, compute $N_i^t := \{k \in N^t : \ell_k^t \leq \ell_i^t\}$ and $\alpha_i^t := \ell_i^t / n_i^t$, where $n_i^t := |N_i^t|$.
8 Pick $i_t \in \arg\min_{i \in N^t} \alpha_i^t$. ▷ Tie-break in favor of larger ℓ_i^t
9 Divide the interval [start, start + $\ell_{i_t}^t$] evenly into n_{i_t} contiguous subintervals (of width $\alpha_{i_t}^t$ each) and allocate them from left to right to the agents in $N_{i_t}^t$ in increasing order of ℓ_i^t values. Update the partial allocation A accordingly.
10
11 start $\leftarrow$ start + $\ell_{i_t}^t$ ▷ Shift the starting point of the cake
12 $N^{t+1} \leftarrow N^t \setminus N_{i_t}^t$ ▷ Update the set of remaining agents
13 $\ell_i^{t+1} \leftarrow \ell_i^t - \ell_{i_t}^t$ ▷ Update the lengths of segments
14 $t \leftarrow t + 1$ ▷ Update the iteration index
15 **end**

▷ Phase 2: Rounding Step

16 If A is fractional, then use the rounding scheme of Lemma 18.
17 **return** A

Lemma 19 describes a useful property of Algorithm 2: For any two consecutive iterations t and $t+1$ of Phase 1, the smallest ratio at t is at most that at $t+1$, i.e., $\alpha_{i_t}^t \leq \alpha_{i_{t+1}}^{t+1}$.

Lemma 19. *Let t and $t+1$ denote any two consecutive iterations of Phase 1 of Algorithm 2. Then, $\alpha_{i_t}^t \leq \alpha_{i_{t+1}}^{t+1}$.*

Proof. By the agent selection rule at iteration t, we have that $\alpha_{i_t}^t \leq \alpha_{i_{t+1}}^t$. That is, $\frac{\ell_{i_t}^t}{n_{i_t}^t} \leq \frac{\ell_{i_{t+1}}^t}{n_{i_{t+1}}^t}$. In particular, this implies that $\frac{\ell_{i_t}^t}{n_{i_t}^t} \leq \frac{\ell_{i_{t+1}}^t - \ell_{i_t}^t}{n_{i_{t+1}}^t - n_{i_t}^t}$. From the update rules in Lines 12 and 13, we know that $\ell_{i_{t+1}}^{t+1} = \ell_{i_{t+1}}^t - \ell_{i_t}^t$ and $n_{i_{t+1}}^{t+1} = n_{i_{t+1}}^t - n_{i_t}^t$. Therefore,

$$\alpha_{i_t}^t = \frac{\ell_{i_t}^t}{n_{i_t}^t} \leq \frac{\ell_{i_{t+1}}^t - \ell_{i_t}^t}{n_{i_{t+1}}^t - n_{i_t}^t} = \frac{\ell_{i_{t+1}}^{t+1}}{n_{i_{t+1}}^{t+1}} = \alpha_{i_{t+1}}^{t+1},$$

as desired. □

Theorem 27 (EF1 and NW for binary left-extremal valuations). *When G is a path and the valuations are binary left-extremal, there always exists a connected allocation that* *is both envy-free up to one good* (EF1) *and non-wasteful* (NW). *Moreover, such an allocation can be computed in polynomial time.*

Proof. We will argue that given any instance I of Connected Fair Division where G is path and the valuations are binary left-extremal, Algorithm 2 always returns a connected, EF1 and NW allocation for I in polynomial time.

Running time: Recall our assumption from Section 7.2 that each agent has non-zero valuation for at least one good. Thus, for every $i \in [n]$, $\ell_i \geq 1$. Therefore, in each iteration t of Phase 1, we have that $N_i^t \geq 1$. The set of remaining agents must therefore shrink by at least one after each iteration (i.e., $|N^{t+1}| < |N^t|$). Hence, Phase 1 can continue for at most n iterations. Each individual iteration of Phase 1 runs in polynomial time. In addition, Phase 2 also runs in polynomial time (18). Overall, Algorithm 2 has a polynomial running time.

Correctness: It is easy to verify that the allocation returned by Algorithm 2 is connected.

To see why the allocation is non-wasteful, recall that by our assumption from Section 7.2, each good is valued by at least one agent. In particular, this implies that $\ell_n = m$, and therefore the allocation is complete (Comp). Furthermore, in any iteration t of Phase 1, we have $\delta_i^t \geq \delta_{i_t}^t$ for any agent $i \in N_{i_t}^t$. The allocation step in Line 9 of Algorithm 2 assigns pieces (of width $\delta_{i_t}^t$ each) from left to right in the order of ℓ_i^t values. Therefore, the piece assigned to agent i is within

$$\delta_{i_t}^t \cdot n_i^t \leq \delta_i^t \cdot n_i^t = \frac{\ell_i^t}{n_i^t} \cdot n_i^t = \ell_i^t,$$

implying that agent i only gets a piece for which it has a non-zero valuation. Thus, the fractional allocation at the end of Phase 1 is non-wasteful. Finally, observe that rounding scheme of Phase 2 preserves non-wastefulness, since each fractionally allocated good is only contested among the agents that have a non-zero valuation for it. Overall, the allocation returned by Algorithm 2 is non-wasteful (NW) with respect to the original instance I.

We will now argue that the allocation returned by Algorithm 2 is envy-free up to one good (EF1). Note that in light of 18, it suffices to argue that the fractional allocation A constructed at the end of Phase 1 is envy-free in the derived instance I^D. Let $j, j' \in [n]$ be any two agents. If j and j' are assigned their respective pieces in the same iteration of Phase 1, then, by the assignment step in Line 9, the pieces must be of equal width. Non-wastefulness and binary valuations assumption together imply that $u_j(A_j) = u_j(A_{j'})$, which gives envy-freeness.

Next, assume, without loss of generality, that agent j is allocated a piece in an earlier iteration of Phase 1 than agent j'. Specifically, suppose the agents j and j' are allocated a piece each by Algorithm 2 in the iterations t and t' respectively, where $t < t'$. After iteration t, any allocation made by the algorithm is to the right of ℓ_{i_t}, and therefore, to the right of ℓ_j. This means that agent j has zero value for the piece allocated to agent j', and therefore does not envy j'. Next, recall that the width of all pieces assigned in iteration t is $\delta_{i_t}^t$. From 19, we have that $\delta_{i_t}^t \leq \delta_{i_{t'}}^{t'}$. Therefore, agent j' is allocated a piece that is at least as large as that of agent j. Along with binary left-extremal valuations assumption, this means that $u_{j'}(A_{j'}) \geq u_{j'}(A_j)$, implying that j' does not envy agent j. This finishes the proof of 27. □

Theorem 28 (EF1 and NW for binary extremal valuations). *When* G *is a path and the valuations are binary extremal, there always exists a connected allocation that is both envy-free up to one good* (EF1) *and non-wasteful* (NW)*. Moreover, such an allocation can be computed in polynomial time.*

Theorem 29 (EF1 and NW for binary k-interval valuations). *When* G *is a path and the valuations are binary* k*-interval, there always exists a connected allocation that is both envy-free up to one good* (EF1) *and non-wasteful* (NW)*. Moreover, such an allocation can be computed in polynomial time.*

We note that the basic idea of the algorithms for *binary extremal* and *binary k-interval* valuations is similar to Algorithm 2, hence, we skip the description of the algorithm and the proof of correctness.

Chapter 8

Connected Fair Division on a Path with Equitability

8.1 Preliminaries and Known Results

We refer the reader to section 7.2 to recall the basic problem setup, and the definitions of efficiency notions (17) and preference restrictions (18).

Definition 19 (Equitability and its Relaxations)**.** *An allocation* $A = (A_1, A_2, \ldots, A_n)$ *is said to be (a) equitable* (EQ) *if for any pair of agents* $i, k \in [n]$*, we have* $u_i(A_i) = u_k(A_k)$*; (b)* ε*-equitable* (ε-EQ) *if for any pair of agents* $i, k \in [n]$*, we have* $u_i(A_i) \geq u_k(A_k) - \varepsilon$*; (c) equitable up to one good* (EQ1) *if for any pair of agents* $i, k \in [n]$*, either* $u_i(A_i) \geq u_k(A_k)$ *or there exists some good* $v \in A_k$ *such that* $u_i(A_i) \geq u_k(A_k \setminus \{v\})$*; and (d) equitable up to one outer good* (EQo1) *if for any pair of agents* $i, k \in [n]$*, either* $u_i(A_i) \geq u_k(A_k)$ *or there exists some good* $v \in A_k$ *such that* $A_k \setminus \{v\}$ *is connected and* $u_i(A_i) \geq u_k(A_k \setminus \{v\})$*.*

Next, we define a notion of (a, b)-*sparse instances*.

(a, b)-*sparse instances:* Given any $1 \leq a \leq m$ and $1 \leq b \leq n$, we say that an instance with binary valuations is (a, b)-*sparse* if each agent approves at most a goods and each good is approved by at most b agents.

8.1.1 Known Results for Equitability

- **Existence**: For *any* ordering of the agents, there exists a connected EQ division of a cake consistent with that ordering (CDP13). For one of these orderings, the equitable allocation is also *proportional* (CDP13).

- For indivisible goods, an EQ allocation might fail to exist, but an EQx allocation always exists and can be computed in polynomial time (GMT14).

- **Computability**: (CP12) first showed that no finite protocol can find a connected division that is simultaneously EQ and proportional. (PW17) showed that this impossibility holds even without the connectedness and proportionality constraints (also see (Chè18)). Though, an "-equitable and connected division can be computed by a *nite* procedure (CP12).

- **EQ and friends**: An EQ+EF+PO division of a cake might not exist (BJK13). However, if we relax PO to completeness, then existence is guaranteed (DS61; Alo87), although the pieces need not be contiguous. If we insist on contiguous pieces, then an EQ division always exists (CDP13; AD15; Chè17) but an EQ+EF division might fail to exist (BJK06). Note that a connected EQ and proportional division always exists (CDP13).

Apart from these, (BCL18) and (Suk19) have studied equitability for connected allocations.

8.1.2 Our Contributions and Organization of the Chapter

Restrictions #	Existence			Computation		
	EQ1 + Comp	EQ1 + PO	EQ1 + NW	EQ1 + Comp	EQ1 + PO	EQ1 + NW
Binary	Yes (Theorem 30)	No (Example 3)	No (Example 3)	P (Theorem 31)	NP-hard (Theorem 34)	NP-hard (Theorem 32)
Binary interval	Yes (Theorem 30)	No (Example 3)	No (Example 3)	P (Theorem 31)	?	?
Binary k-interval	Yes (Theorem 30)	No (Example 4)	No (Theorem 37)	P (Theorem 31)	P (Theorem 38)	P (Theorem 37)
Binary extremal	Yes (Theorem 30)	No (Example 5)	No (Theorem 36)	P (Theorem 31)	P (Theorem 39)	P (Theorem 36)
Binary left-extremal	Yes (Theorem 30)	No (Example 5)	No (Theorem 35)	P (Theorem 31)	P (Theorem 37)	P (Theorem 35)

Table 8.1: Summary of results for Connected Fair Division-EQ1. For every fairness/efficiency requirement (columns) and preference restriction (rows), the entries in the *Existence* section pertain to answers to the question "Does a desired allocation always exist?" and those in the *Computation* section pertain to the computational complexity of determining whether a desired allocation exists. The entries marked with P pertain to polynomial-time algorithms that also return the desired allocation (whenever it exists).

Organization of the Chapter.

- **Algorithms:** In Section 8.3, we present a polynomial time algorithm to compute *complete*+EQ1 allocations for any ordering of agents Theorem 31. (CDP13) showed a similar result in the contiguous setting, our result can be perceived as an analog of their result in the discrete world.
- **Hardness Results:** In Section 8.4 we show hardness for checking existence of EQ1+PO and EQ1+NW allocations. All of our results follow from a *single* construction that also has implications for other fairness notions such as *envy-freeness up to one good* (EF1) as well as negatively valued items (or *chores*).
- **Restricted Domains:** In Section 8.5 we show polynomial time algorithms for computing EQ1+PO and EQ1+NW allocations for *Binary le -extremal*, *Binary extremal*, and *Binary k-interval* valuations.

8.2 Existence of EQ1 allocations

EQ1 and completeness. Cechlárová et al. in (CDP13) showed an existence of complete *EQ* allocation of the cake for any ordering of the agents. Here, the ordering agents signifies the order in which of agents receive a bundle of connected goods from left to right on the path of goods according to final allocation. Similar to Theorem 24 for EF1 allocation, we give an existence result for connected complete *EQ1* allocations as follows:

Theorem 30. *Given a* Connected Fair Division *instance where* G *is a path and valuations are binary, a connected allocation that is equitable up to one good* (EQ1) *and complete* (Comp) *always exists.*

Proof. (Sketch) We convert the discrete connected fair division instance to *derived cake cu ing instance* (we refer the reader to Section 7.2 for the definition and properties of these instances). From the existence result (CDP13) we know there exist an EQ allocation for any ordering of agents for this instance. We now invoke the *Rounding Scheme* from lemma 18 in the way as described in the context of EQ1 allocations. Employing this lemma, we get the desired existence of EQ1 allocations for the discrete connected fair division instance for *any* binary valuations. □

8.2.1 EQ1 and Pareto Optimality

When G is a path, there might not exist a connected allocation that is both EQ1 and PO (or both EQ1 and NW), even for binary interval valuations.

Example 3. *e example due to Igarashi and Peters (IP19) for showing non-existence of* EF1 + PO *on paths works.*

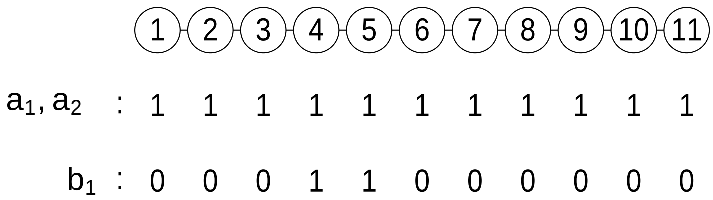


Figure 8.1: Non-existence of EQ1+ PO allocation even for interval valuations

In any EQ1 + PO *allocation (say) at least one of* a_i^0s *get three goods that she approves on the right side of goods* f4, 5g. *At this point,* EQ1 *implies* b_1 *gets both the goods she approve. It is easy to see that if both the* a_i^0s *are allocated on one side of the goods* 5, 6 *then the allocation will be* PO *since at least one agent will get the utility of at most 3. Hence,* a_1 *and* a_2 *are allocated on either side of goods* f4, 5g. *At this stage, it is easy to see that there cannot exist an* EQ1 *allocation which* PO.

Note that similar argument works for showing non-existence of EQ1 + *Non-Wasteful allocations.*

Next, we show example for non-existence of EQ1+ PO and EQ1 + NW allocations for extremal valuations.

Example 4. *We show an example with* 3 *agents and* k-interval *valuations to show non-existence of* EQ1+PO *(Figure 8.2).*

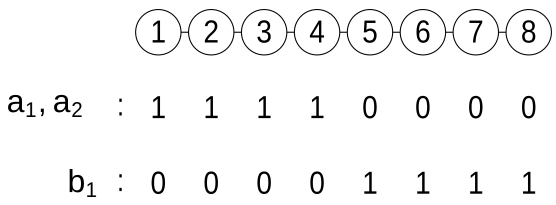


Figure 8.2: Non-existence of EQ1+ PO allocation even for *k-interval* valuations

Example 5. *We show a small example with* 2 *agents and* left extremal *valuations (Figure 8.3).*

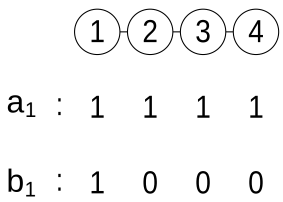


Figure 8.3: Non-existence of EQ1+ PO allocation even for extremal valuations

It is easy to see that any PO *allocation will allocate agent* a_1 *goods* $\{2, 3, 4\}$ *and good* $\{1\}$ *to agent* b_1. *This is true even for* NW. *Hence, there cannot exist* EQ1+PO *or* EQ1 + NW *allocation.*

8.2.2 A few remarks

Remark 1. *For the case with non-connected allocations, it is easy to see that the Non-wastefulness and Pareto Optimality are equivalent conditions. This does not hold when we insist on connected allocations.*

Proof by example with two agents A, B with *exclusive valuations*:

$$a_1 - a_2 - b_1 - a_3 - b_2$$

Consider an allocation where agent A is allocated $\{a_1, a_2, b_1, a_3\}$ and agent B is allocated good b_2. This allocation is PO but not NW.

Remark 2. *For the binary Non-wasteful case* $EQ \implies EF$ *but* $EF \not\Rightarrow EQ$.

The proof for the above remark is as follows: In EQ allocations with binary valuations, every agent is allocated the equal number of goods that she likes. The non-wastefulness condition implies that no agent is allocated any good that she did not approve. Hence, for any agent, all other agents can receive at the most same number of goods as she has received from the set of goods valued by her. Hence the allocation is EF.

Consider an example with two agents A, B with *exclusive valuation*:

$$a_1 - a_2 - a_3 - a_4 - b_1 - b_2$$

The allocation which assigns every agent, all the goods that she approves is EF but not EQ.

8.3 Efficient Algorithms for Equitable Allocations

8.3.1 Algorithm for complete-EQ1 Allocations

In this section, we present an efficient algorithm for finding complete EQ1 allocation for Connected Fair Division instance I. Our results a proves a stronger statement that for *any* given ordering of agents , there exist a complete-EQ1 allocation which can be computed in polynomial time. Here, by ordering of agents we mean the ordering respected by the desired allocation in terms of allocating connected bundle of goods from left to right on a path. For the derived cake cutting instance, Cechlárová et al. in (CDP13) showed an existence of complete *EQ* allocation but their result was non-constructive. Our result can be perceived as a positive complement of this result in the discrete world.

Theorem 31. *ere is a polynomial time algorithm that return a* complete-EQ1 *connected allocation of a path for* any *ordering of agents for binary valuations.*

Before describing the algorithm we first set up some notations. Let $[\ ,\ ]$ denote the connected of bundle of goods from good to good such that $<$. A connected fair division instance $I = \langle N, [\ ,\ ]\rangle$ denote a sub-instance of original instance restricted on agents from set N and goods from range to from the original instance. Note that the valuations are projection of valuations in the original instance. m denotes the last good on the path. We first give an intuitive description of the algorithm.

Our algorithm runs in two phases. We fix an arbitrary ordering of agents amongst all possible $n!$ orderings. Given , for i^{th} round in Phase 1, we try to find a *minimal* allocation which respects the fixed ordering of agents, and provides utility of i to every agent. We repeat this until we reach the utility p for which we run out of goods before we allocate the utility of p to every agent, according to the fixed ordering. We call p the 'critical utility'. Next, in Phase 2, we move along and try to allocate a bundle of utility p to maximum number initial agents. We find such allocations until we reach an agent q such that it is not possible to find an allocation which gives utility of $p - 1$ to all remaining agents after q, if we allocate a bundle of utility p to q. We call q the 'critical agent'. At this stage, for all the remaining agents including q, we find an allocation with utility $p - 1$ in reverse order of agents with respect to .

We now turn to the formal description of the algorithm.

Description of the algorithm. Let be the fixed ordering of agents. Once is fixed, our algorithms works in two phases. In Phase 1, we iteratively find out the maximum utility p that any agent can have in the final allocation A. Let $\ ^t_i$ be the latest (earliest) good from the left for which $u_t([\ ^{t-1}_i + 1,\ ^t_i]) = i$. Let $\ ^0_i + 1 = 1 8i$ that is in each iteration, the starting point of allocation for the first agent according to is the first good on the path. Intuitively, at each iteration i we check if every agent can be allocated a bundle of goods with utility i. If we are able find such allocation and there are still some goods remaining at the end, then we proceed to iteration $i + 1$. Note that in each iteration i, the allocated bundle for every agent t is connected bundle of goods – $[\ ^{t-1}_i + 1,\ ^t_i]$. We stop when we reach $i = p$ for which it is not possible to allocate all agents the utility of p (i.e. the goods are finished before all agents can achieve desired utility) for an allocation according to the order . We call such utility p a *'critical utility'* see Fig. 8.4. The aim of the first phase is to find the *critical utility* p. We note that for every fixed value of utility, phase 1 finds a minimal allocation which respects and provides every agent with given utility. The utility p is the highest utility that any agent can have for a connected-EQ1 allocation which respects ordering .

In the second phase, initially we fix the utility to p. Let N_j denote the ordered set of j^{th} to n^{th} agent according to . We start from first agent according to (say n_1). We temporarily allocate a bundle $[1,\ ^1_p]$ with utility p to agent one. For the modified instance $I^0 = hN_2, [\ ^1_p + 1, m]i$ we check if each agent can achieve utility of $p - 1$ with a procedure used in Phase 1. If we can find such allocation we fix the allocation for agent one, remove agent one and corresponding allocated goods to obtain a modified instance. We move onto the next agent according and repeat the procedure until we find agent q such that for instance $I^{00} = hN_{q+1}, [\ ^q_p + 1, m]i$ does not admit allocation with utility $p - 1$ for all agents in I^0. We call such q *'critical agent'*.

Once we find *critical agent* q, we go back to the instance $I\ = hN_q, [\ ^{q-1}_p + 1, m]i$. Let $\ ^0$ be the ordering reverse of projected on agents from N_q. We start allocating agents a connected bundle of utility $q - 1$ according to $\ ^0$ from the last good (i.e. we allocate by computing bundles from right to left in a similar way as we did from left to right). Once we reach the last agent, we allocate her the connected bundle of all the remaining goods.

This concludes the description of our algorithm. It is clear that the algorithm runs in polynomial time. Next we move onto the correctness.

Proof of correctness. We recall that p is the *critical utility* and q is the *critical agent*.

Lemma 20. *e allocation A returned by the algorithm is* complete*, and each agent receives an utility of p or $p - 1$.*

Proof. From the description the algorithm the completeness of allocation follows since we allocate all the remaining goods to agent the last agent at the end of Phase 2. For the latter part of the Lemma, consider the agents from a_1 to a_{q-1} each of these agent receives a bundle with utility p in A. Next, consider all agents from set N_q. We claim that these agents gets a bundle of goods with utility $p - 1$ in A. We need to argue that such allocation can be done without exhausting all the remaining goods. Consider the instance $I = hN_q, [\ ^{q-1}_p + 1, m]i$. For this instance, we know that there exist an

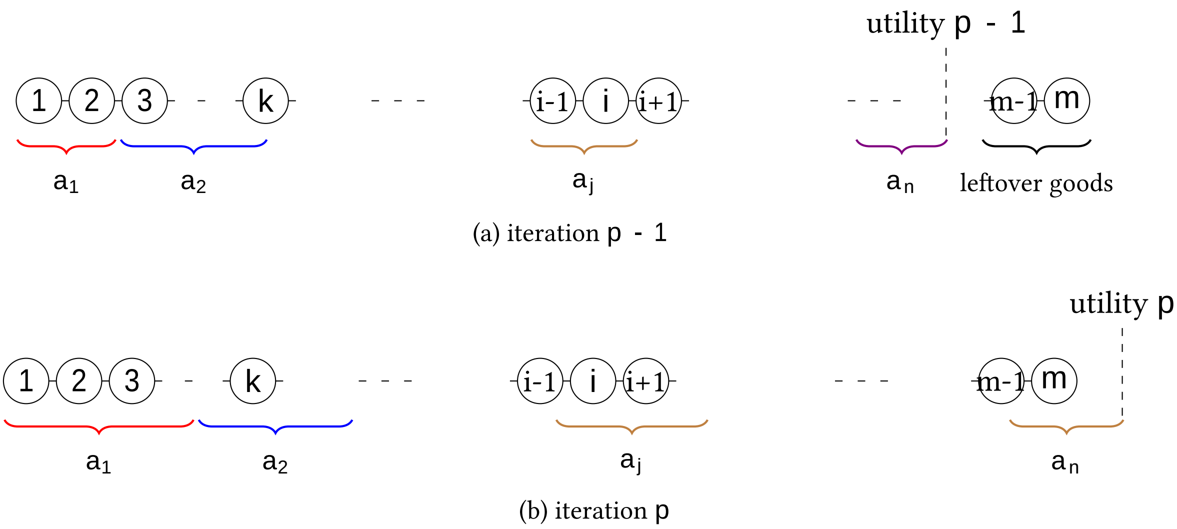


Figure 8.4: Phase I: Critical utility

allocation which provide utility of $p-1$ for each agent in set N_q since the algorithm performed this check before allocating agent $q-1$ a bundle with utility p. Consider the following allocation A returned by the algorithm on instance I: We start according to I' and minimal allocation from the last good. For first $|N_q|-1$ agents we allocate the bundle of utility $p-1$, the existence for such an allocation is clear since there is such allocation for all agents in set N_j in I. Now we give rest of the goods to the last agent. We claim that last agent has utility of $p-1$. It is easy to see that the utility is at least $p-1$ since instance I has an allocation which provides each agent in set N_q. To show the upperbound consider contradiction that is the last agent (a_q) has utility strictly greater than $p-1$. Then, in such an allocation q has utility of at least p, and all other agents appearing after q has utility of $p-1$ which contradicts the fact that q is a *critical agent*. Hence, a_q can have utility at most $p-1$. This completes the proof for the lemma. □

Given Lemma 20 the proof for Theorem 31 is trivial.

8.4 Hardness Results

8.4.1 Existence of EQ1 and Non-Wasteful

In this section, we start with showing a hardness for deciding an existence of Non-Wasteful EQ1 allocations. We recall that Non-Wasteful allocation implies completeness by definition (Section 7.2).

We first describe the version of SATISFIABILITY that we will reduce from. Our instance consists of $(4p+q)$ clauses which we will typically denote as follows:

$$C = \{A_1, B_1, A'_1, B'_1, \cdots, A_p, B_p, A'_p, B'_p\} \cup \{C_1, \cdots, C_q\}$$

We refer to the first $4p$ clauses as the *core* clauses, and the remaining clauses as the *auxiliary* clauses. The core clauses have two literals each, and also enjoy the following structure:

$$\forall i \in [p], A_i \cap B_i = \{x_i\} \text{ and } A_i' \cap B_i' = \{\overline{x}_i\}$$

We refer to the x_i's as the *main* variables and the remaining variables that appear among the core clauses as *shadow* variables. The shadow variables occur exactly once with negative polarity among the core clauses. Therefore, using $\ell(\,)$ to denote the set of literals occuring amongst a subset of clauses, we have:

$$\ell\left(\bigcup_{i=1}^{p} \{A_i, B_i, A_i', B_i'\}\right) = 6p.$$

The auxiliary clauses have the property that they only contain the shadow variables, which occur exactly once amongst them with positive polarity. Also, every auxiliary clause contains exactly four literals. Note that this also implies, by a double-counting argument, that $q = p$. We say that a collection of clauses is a *chain* if it has all the properties described above. An instance of Linear Near-Exact Satisfiability (LNES) is the following: given a set of clauses that constitute a chain, is there an assignment of truth values to the variables such that exactly one literal in every core clause and two literals in every auxiliary clause evaluate to true under ?

For ease of discussion, given an assignment of truth values we often use the phrase " satisfies a literal" to mean that the literal in question evaluates to true under . For instance, the question from the previous paragraph seeks an assignment that satisfies exactly one literal in every core clause and two literals in every auxiliary clause. We also refer to such an assignment a near-exact satisfying assignment. The following observation is a direct consequence of the definitions above.

Proposition 6. *Let C be a collection of clauses that form a chain. For any assignment of truth values , the main variables satisfy exactly two core clauses and the shadow variables satisfy either one core clause or one auxiliary clause.*

We first establish that LNES is NP-complete:

Lemma 21. *Linear Near-Exact Satisfiability is NP-complete.*

Proof. We reduce from (2/2/4)-SAT, which is the variant of Satisfiability where every clause has four literals and every literal occurs exactly twice — in other words, every variable occurs in exactly two clauses with positive polarity and in exactly two clauses with negative polarity. The question is if there exists an assignment of truth values to the variables under which exactly two literals in every clause evaluate to true.

Let be a (2/2/4)-SAT instance over the variables $V = \{x_1, \ldots, x_n\}$ and clauses $C = \{C_1, \ldots, C_m\}$. For every variable x_i, we introduce four new variables: p_i, r_i and q_i, s_i. We replace the two positive occurrences of x_i with p_i and r_i, and the two negated occurrences of x_i with q_i and s_i. We abuse notation and continue to use $\{C_1, \ldots, C_m\}$ to denote the modified clauses. Also, introduce the clauses:

$$A_i = (x_i, \overline{p}_i), B_i = (x_i, \overline{r}_i), A'_i = (\overline{x}_i, \overline{q}_i), B'_i = (\overline{x}_i, \overline{s}_i),$$

for all $1 \leqslant i \leqslant n$. Note that the collection of clauses described form a chain, as required. We use to refer to this formula. We now turn to the argument for equivalence.

In the forward direction, let be an assignment that sets exactly two literals of every clause in to true. Consider the assignment given by:

$$(x_i) = (x_i), \ (p_i) = (r_i) = (x_i); \ (q_i) = (s_i) = 1 - (x_i),$$

for all $1 \leqslant i \leqslant n$. It is straightforward to verify that satisfies exactly one literal in every core clause and exactly two literals in every auxiliary clause.

In the reverse direction, let be an assignment for the variables of that satisfies exactly one literal in every core clause and exactly two literals in every auxiliary clause. Define as the restriction of on the main variables. Let C be a clause in . To see that satisfies exactly two literals of C, note that the following:

$$(p_i) = (r_i) = (x_i) = (x_i); \ (q_i) = (s_i) = 1 - (x_i) = 1 - (x_i)$$

is forced by the requirement that must satisfy exactly one literal in each core clause. Therefore, if satisfies more or less than two literals of any clause C, then that behavior will be reflected exactly in the auxiliary clause corresponding to C, which would contradict the assumed behavior of . We make this explicit with an example for the sake of exposition. Let C from be the clause $(x_1, \overline{x}_3, \overline{x}_6, x_7)$, and let the clause constructed in be (p_1, q_3, q_6, r_7). Suppose $(x_1) = (x_7) = (x_6) = 1$ and $(x_3) = 0$. Then we have $(p_1) = (r_7) = 1$ and $(q_6) = 0$, while $(q_3) = 1$. This demonstrates that satisfies three literals in the auxiliary clause corresponding to C, in one-to-one correspondence with the literals that were satisfied by . This completes our argument. □

Theorem 32. *Determining the existence of a connected non-wasteful* EQ1 *allocation is* NP*-complete even when* G *is a path and the valuations are binary, and every agent approves a constant number of goods.*

Proof. Let be an instance of LNES in standard notation. We introduce one good for every core clause, denoted A_i, B_i, A'_i, B'_i, and three goods for every auxiliary clause, denoted C^L_i, S_i, C^R_i. We also introduce $2p$ dummy goods denoted by $D_1, D'_1, \ldots, D_p, D'_p$.

We introduce two agents for each main variable, which is useful to interpret as one agent for every literal corresponding to the main variables. For a main variable x_i, the agent corresponding to the positive literal x_i approves the goods A_i, B_i, D_i, D'_i, while the agent corresponding to the negative literal $\overline{x}_i$ approves the goods A'_i, B'_i, D_i, D'_i.

We also introduce an agent for every shadow variable. If y is a shadow variable occurring in the core clause A_i and auxiliary clause C_j, then the agent corresponding to y approves the goods A_i, C^L_j and C^R_j. The set of goods approved by y is analogously defined if the core clause it appears in were to be B_i, A'_i or B'_i. Finally, we introduce q dummy agents $T_1, \ldots, T_q$ and have T_i approve S_i for all $i \in [q]$.

The goods appear in the following order:

$$A_1, B_1, A_1^0, B_1^0, \quad , A_p, B_p, A_p^0, B_p^0, C_1^L, S_1, C_1^R, \quad , C_q^L, S_q, C_q^R, D_1, D_1^0, \quad , D_p, D_p^0.$$

This completes the description of the construction. We now turn to the proof of equivalence.

The Forward Direction. Let be the satisfying assignment for the LNES. We construct the desired allocation follows: (We show the allocation with respect to a particular variable being set to one, the case for when a variable is set to zero is symmetric)

- If $(x_i) = 1$, we assign A_i, B_i to agent A_{x_i} and D_i, D_i^0 to agent $A_{\overline{x}_i}$.
- Allocate one good each to the two shadow agents corresponding to negative x_i from the set (A_i^0, B_i^0) such that both agents get one good that they approve
- If literal x_i occurs in clauses i and j, allocate exactly one of (C_j^L, C_j^R) and one of (C_k^L, C_k^R) respectively to the two shadow agents corresponding to positive x_i according to the allocation status of these goods .
- Allocate all the S_i^0s to the respective T_i^0s (this is true irrespective of any variable assignment)

As mentioned above, the assignment is symmetric when $(x_i) = 0$. It is easy to verify that the above assignment is complete and non-wasteful (NW). We allocate at least one and at most two approved goods to any agent in the assignment, hence, the assignment is EQ1.

The Reverse Direction. We will now show the procedure to recover an assignment for LNES given the EQ1 + NW allocation for the fair division instance. We first study the structure of any valid allocation .

Since the fate of all dummy agents along with all the S_i^0s is fixed in all valid allocations (each dummy agent A_{d_i} is assigned a good S_i that she likes), we do not consider them in our analysis. The fair division instance has total $2p$ (corresponding to the positive and negative literals for each variable) $+4p$ (shadow agents) $= 6p$ agents. Similarly, there are $4p$ (corresponding to shadow agents) $+2q$ (for the clauses) $+2p$ (dummy) $= 8p$ (since $p = q$) goods present in the constructed instance. The $2p$ dummy good needs to be allocated to at least p literal agents due to NW. We next show that these $2p$ goods will be allocated between exactly p literal agents one corresponding to each variable for the allocation to be NW + $\mathsf{complete}$.

For the sake of contradiction, let us assume that strictly more than p literal agents get their allocation from the set of dummy goods. One needs exactly $2p$ shadow agents to pick-up the $2p$ goods corresponding to p clauses since these are separated by S_i^0s. This leaves us with $4p$ goods from core clauses which needs all of the remaining $2p$ shadow agents (each can pick one) and at least p (each can pick two) literal agents for complete

NW allocation. Hence, exactly p literal agents get allocation from the set of dummy goods and all literal agents get **two** goods each in any complete NW allocation.

Given that we want to allocate two shadow goods to a literal agent, these two goods are either goods corresponding to the two positive shadow variables or to the two negative shadow variables due to non-wastefulness. Hence, for each variable x_i, either two positive or two negative shadow agents occupy the clause goods. At this stage, we propose the assignment for LNES. We set the variable $x_i = 1$ iff the dummy goods $(D_i, D_i^0$ are allocated agent $A_{\overline{x}_i}$. Otherwise we set $x_i = 0$. We show that the proposed assignment is indeed a satisfying assignment. It is clear that each variable is set to either zero or one. Also, for each clause C_j, the variables corresponding to the agents which received C_j^L and C_j^R, are the two variables satisfying that clause. For each variable, since either the positive literal or negative literal corresponding to the variable satisfies the two clauses it appears in, the assignment extracted is a valid assignment (i.e. does not have any conflict). □

In the above reduction, we do not specifically use the notion of EQ1, and its just a consequence of reduction; the same construction also shows hardness for determining the existence of only $\mathsf{Non-wasteful}$ allocations on the path even without any other fairness notion. It is interesting to note that every agent approves only a constant number of goods and consequently have only constant number of approval intervals.

8.4.2 Complexity of finding EQ1+complete allocation with maximum utility

For complete EQ1 allocations, we know the existence from Theorem 30. An efficient algorithm to compute a complete EQ1 allocation for *any ordering* of agents was shown in section 8.3.1. We note that, the utility obtained by each agent through an allocation from 8.3.1 is dependent upon the initial ordering of the agents. It is easy to see that, among all these complete EQ1 allocations, there exist an allocation which maximizes the total utility over the given instance (i.e. maximizing the summation of utilities of all agents from the allocation) for some initial ordering over the agents.

In this section, using *turing reduction* from LINEAR NEAR-EXACT SATISFIABILITY(LNES) we show that an algorithm which finds complete EQ1 allocation which maximizes the utility is unlikely to run in polynomial time.

Theorem 33. *Unless $P = NP$, there is no polynomial time algorithm for computing complete* EQ1 *allocation which maximizes the utility over given instance.*

Proof. Our construction is similar to the one used in Theorem 8.4.3. Let be an instance of LNES in the standard notation. We introduce one good for every core clause, denoted A_i, B_i, A_i^0, B_i^0, and three goods for every auxiliary clause, denoted C_i^L, S_i, C_i^R. We also introduce $2p$ dummy goods denoted by $D_1, D_1^0, \ldots, D_p, D_p^0$ and a separator good S_0.

We again, introduce agent corresponding to every literal and a an agent for every shadow variable as we did in Theorem 8.4.3 with the same set of approved goods for each agent. For the sake of completeness, we describe the valuations for each agent. For a main variable x_i, the agent corresponding to the positive literal x_i (A_{x_i}) approves the goods A_i, B_i, D_i, D_i^0, while the agent corresponding to the negative literal $\overline{x}_i$ ($A_{\overline{x}_i}$) approves the goods A_i^0, B_i^0, D_i, D_i^0. If y is a shadow variable occurring in the core clause A_i and auxiliary clause C_j, then the agent corresponding to y approves the goods A_i, C_j^L and C_j^R. The set of goods approved by y is analogously defined if the core clause it appears in were to be B_i, A_i^0 or B_i^0. Finally, we introduce q dummy agents $T_1, \dots, T_q$ and have T_i approve S_i for all $i \in [q]$. Lastly, we introduce a separator agent (A_0) which approve exactly one good S_0.

The goods appear in the following order:

$$A_1, B_1, A_1^0, B_1^0, \quad , A_p, B_p, A_p^0, B_p^0, S_0, C_1^L, S_1, C_1^R, \quad , C_q^L, S_q, C_q^R, D_1, D_1^0, \quad , D_p, D_p^0$$

This completes the description of the construction.

In any EQ1 allocation, the maximum utility for agents corresponding to the literals is 2 (i.e $u_i(A_{x_i}) = u_i(A_{\overline{x}_i}) \le 2$ since the separator agent can have utility at most one. Similarly, any agent corresponding to shadow variable can have utility at most one. Notice that in order to obtain utility strictly greater than one, the allocation for shadow agent must contain the separator good S_0. Again, as separator agent only approves S_0 this contradicts EQ1. It is easy to see that the maximum utility for separator agent is one.

Next, we show that if there exist an algorithm A which finds complete EQ1 allocation which maximizes the utility then by running A on the constructed instance (say I) we can solve the standard decision version of LNES. We run A on I and say return *YES* instance of LNES, iff the allocation admits the following structure:

$$\begin{aligned} u_i(A_{x_i}) &= u_i(A_{\overline{x}_i}) = 2 \\ u_i(y) &= 1 \\ u_i(A_0) &= 1 \end{aligned} \tag{8.1}$$

Clearly, conditions from equation 8.1 can be checked in polynomial time. Hence, using A, the NP-complete LNES problem can be solved in polynomial time which is a contradiction.

Given *YES* instance of , let be the satisfying assignment for the LNES. Consider the following allocation ( ): (We show the allocation with respect to a particular variable being set to one, the case for when a variable is set to zero is symmetric)

- If $(x_i) = 1$, we assign A_i, B_i to agent A_{x_i} and D_i, D_i^0 to agent $A_{\overline{x}_i}$.
- Allocate one good each to the two shadow agents corresponding to negative x_i from the set (A_i^0, B_i^0) such that both agents get one good that they approve
- If literal x_i occurs in clauses i and j, allocate exactly one of (C_j^L, C_j^R) and one of (C_k^L, C_k^R) respectively to the two shadow agents corresponding to positive x_i according to the allocation status of these goods .

- Allocate all the $S_i^{\prime}$s to the respective $T_i^{\prime}$s (this is true irrespective of any variable assignment)

- Allocate S_0 to A_0.

Clearly, exists for *YES* instance of and satisfies equation 8.1. Since is allocates each agents its maximum possible utility the overall utility is maximum (we call this utility u). Observe that an allocation which maximizes the individual utilities is the only way to achieve utility u in I, hence, A must return allocation which satisfies 8.1.

Conversely, if A returns allocation which satisfies 8.1. Then, it is easy to see that must follow the structure described above for allocation due to the way valuations are set up. Hence, there exist a satisfying assignment for (since we can recover a valid solution using the structure of allocation , for further details refer to the reverse direction in Theorem 8.4.3).

□

8.4.3 Existence of Pareto Optimal Allocation

In this section, we study the complexity of Pareto Optimal allocations with equitability and envyfreeness.

Theorem 34. *Checking the existence of a connected allocation that is (a)* EQ1 *and* PO*, (b)* EF1 *and* PO*, (c)* EQ1 *and has egalitarian welfare at least* 2*, or (d)* EF1 *and has egalitarian welfare at least* 2 *is* NP*-complete even for a path and a* $(6, 4)$*-sparse binary valuations instance.*

Recently, Igarashi and Peters (IP19, Theorem 7) have shown of checking the existence of a connected EF1+PO allocation of a path even for binary valuations. Their construction involves items that are valued by *all* agents, thus requiring $O(n)$ sparsity. By contrast, our result in Theorem 34 shows hardness even for $O(1)$ sparse instances. Finally, we note that the proof of Theorem 23 can also be adapted to show for egalitarian or utilitarian-optimal EQ1 allocations of *chores*.[1]

We will show a reduction from Linear Near-Exact Satisfiability (LNES) and our construction will be similar to that in the proof of . Recall that an instance of LNES consists of $5p$ clauses (where $p \in \mathbb{N}$) denoted as follows:

$$C = \{U_1, V_1, U_1^{\prime}, V_1^{\prime}, \quad , U_p, V_p, U_p^{\prime}, V_p^{\prime}\} \cup \{C_1, \quad , C_p\}.$$

We will refer to the first $4p$ clauses as the *core* clauses, and the remaining clauses as the *auxiliary* clauses. The set of variables consists of p *main variables* $x_1, \ldots, x_p$ and $4p$ *shadow variables*.

Each core clause consists of two literals and has the following structure:

$$\forall i \in [p], U_i \cap V_i = \{x_i\} \text{ and } U_i^{\prime} \cap V_i^{\prime} = \{\overline{x}_i\}.$$

[1]The relevant transformation is $u_{i,j}^{\prime} = u_{i,j} - 1$.

Each main variable x_i occurs exactly twice as a positive literal and exactly twice as a negative literal. The main variables only occur in the core clauses. Each shadow variable makes two appearances: as a positive literal in an auxiliary clause and as a negative literal in a core clause. For $i \in [p]$, we will let p_i, r_i, q_i, and s_i denote the shadow variables that appear (as negative literals) in the core clauses U_i, V_i, U'_i and V'_i, respectively. That is, $U_i := (\overline{p_i} \wedge x_i), V_i := (\overline{r_i} \wedge x_i)$, $U'_i := (\overline{q_i} \wedge \overline{x_i})$, and $V'_i := (\overline{s_i} \wedge \overline{x_i})$. Each auxiliary clause consists of four literals, each corresponding to a positive occurrence of a shadow variable.

The LNES problem asks whether, given a set of clauses with the aforementioned structure, there exists an assignment of truth values to the variables such that *exactly one* literal in every core clause and *exactly two* literals in every auxiliary clause evaluate to TRUE under .

We will start by discussing the proof of part (a) of Theorem 34, followed by that of part (c) which uses the same construction.

Construction of the reduced instance. Let be an instance of LNES. We will begin with the description of the reduced instance.

Goods: For every $i \in [p]$, we introduce one good for every core clause denoted by U_i, V_i, U'_i, V'_i, and six goods for every auxiliary clause denoted by $C_i^{L_1}, C_i^{L_2}, S_i^1, S_i^2, C_i^{R_1}, C_i^{R_2}$. We refer to U_i, V_i, U'_i, V'_i as the *core* goods, $C_i^{L_1}, C_i^{L_2}, C_i^{R_1}, C_i^{R_2}$ as the *auxiliary* goods, and S_i^1, S_i^2 as the *separator goods*. Next, we introduce two goods for each shadow variable, i.e., corresponding to each of p_i, q_i, r_i, s_i, we introduce the following *shadow* goods: $p_i^1, p_i^2, r_i^1, r_i^2, q_i^1, q_i^2, s_i^1, s_i^2$. Finally, we introduce $2p$ *dummy* goods denoted by $D_1, D'_1, \ldots, D_p, D'_p$, two additional *separator* goods S_0^1, S_0^2, and three *special* goods S_1, S_2, S_3. Thus, the total number of goods is $m = 4p + 6p + 8p + 2p + 2 + 3 = 20p + 5$. The goods are arranged as shown in Figure 8.5.

$$U_1, p_1^1, p_1^2, r_1^1, r_1^2, V_1, U'_1, q_1^1, q_1^2, s_1^1, s_1^2, V'_1, \quad , U_p, p_p^1, p_p^2, r_p^1, r_p^2, V_p, U'_p, q_p^1, q_p^2, s_p^1, s_p^2, V'_p$$

(Core and shadow goods)

$$S_0^1, S_0^2, C_1^{L_1}, C_1^{L_2}, S_1^1, S_1^2, C_1^{R_1}, C_1^{R_2}, \quad , C_p^{L_1}, C_p^{L_2}, S_p^1, S_p^2, C_p^{R_1}, C_p^{R_2}$$

(Separator and auxiliary goods)

$$D_1, D'_1, D_2, D'_2, \quad , D_p, D'_p, S_1, S_2, S_3$$

(Dummy and special goods)

Figure 8.5: The instance used in the proof of part (a) of Theorem 34. The path graph is such that the goods in the top row are to the left of those in the middle row, which are to the left of those in the bottom row.

Agents: For every main variable x_i, we will introduce two agents a_{x_i} and $a_{\overline{x}_i}$ for the two literals; these are referred to as *main agents* of the *positive* and *negative* type, respectively. For every $i \in [p]$, the agent a_{x_i} approves (i.e., values at 1) the goods U_i, V_i, D_i, D'_i, while the agent $a_{\overline{x_i}}$ approves the goods U'_i, V'_i, D_i, D'_i. We also introduce a *shadow agent* for every shadow variable. If p_i is a shadow variable occurring in core clause U_i and auxiliary clause C_j, then the corresponding shadow agent p_i approves the shadow goods p_i^1, p_i^2 and the auxiliary goods $C_j^{L_1}, C_j^{L_2}, C_j^{R_1}, C_j^{R_2}$. The valuations of the other shadow agents r_i, q_i, s_i are defined analogously. Next, we introduce $p+1$ *separator agents* $t_0, \ldots, t_p$ such that for every $i \in [p]$, t_i approves two separator goods S_i^1, S_i^2. Lastly, we introduce *special agent* a_s that approves the special goods S_1, S_2, S_3.

This completes the construction of our reduction. Notice that the constructed instance is $(6,4)$-*sparse*. Before presenting the proof of equivalence, we will establish in Lemma 22 that the each agent (except for the special agent) has a utility of 2 under any EQ1 and PO allocation in the fair division constructed above.

Lemma 22. *In any* EQ1 + PO *allocation, the utility of the special agent* a_s *is equal to* 3 *and that of every other agent is equal to* 2.

Proof. Notice that in any PO allocation A, the special goods S_1, S_2, S_3 must be allocated to the special agent a_s. This is because these goods lie at the end of the path and are uniquely valued by a_s, and therefore any allocation A' that does not assign these goods to a_s can be shown to be Pareto dominated by another allocation that is identical to A' except for the assignment of the special goods to the special agent. Therefore, the utility of a_s under PO allocation must be equal to 3 (recall that a_s does not value any good other than the special goods).

Now let A denote any EQ1 and PO allocation. Since the utility of the special agent in A is equal 3, the EQ1 condition implies that the utility of every other agent in A is at least 2.

Since each separator agent $t_0, t_1, \ldots, t_p$ approves exactly two goods, it must be that for every $i \in \{0, 1, \ldots, p\}$, the separator goods S_i^1, S_i^2 are assigned to t_i in A. Furthermore, since the separator goods S_i^1, S_i^2 are placed next to each other on the path and these are the only goods approved by t_i, we can assume, without loss of generality, that these are the only goods assigned to t_i.

Now consider a shadow agent p_i that appears in the core clause U_i and the auxiliary clause C_j. Thus, p_i approves two shadow goods p_i^1, p_i^2 and four auxiliary goods $C_j^{L_1}, C_j^{L_2}, C_j^{R_1}, C_j^{R_2}$. Note that p_i cannot receive more than two goods that it approves; if it does, then, due to connectedness constraint, its bundle should necessarily include separator goods whose assignment has already been fixed. Thus, each shadow agent p_i (analogously q_i, r_i, s_i) will have a utility of exactly 2 in A.

A similar argument shows that for any $i \in [p]$, the main agent of positive (or negative) type a_{x_i} (or $a_{\overline{x_i}}$) will have a utility of at most 2 since all such agents approve two core goods and two dummy goods. We therefore have that in any EQ1 and PO allocation, all agents other than the special agent achieve a utility of exactly 2. This completes the proof of Lemma 22. □

The Forward Direction. Given a satisfying assignment for LNES, we will construct the desired allocation as follows:

- Allocate the special goods S_1, S_2, S_3 to the special agent a_s.

- For each $i \in \{0, 1, \dots, p\}$, the separator agent t_i receives the separator goods S_i^1 and S_i^2.

- If $\sigma(x_i) = 1$, then allocate $\{U_i, p_i^1, p_i^2, r_i^1, r_i^2, V_i\}$ to agent a_{x_i} and $\{D_i, D_i'\}$ to agent $a_{\overline{x}_i}$. In addition, allocate $\{U_i', q_i^1, q_i^2\}$ to q_i, and $\{s_i^1, s_i^2, V_i'\}$ to s_i. Recall that q_i and s_i are the shadow variables that appear as negated literals in the core clauses U_i' and V_i', respectively, along with $\overline{x_i}$.

  Otherwise, if $\sigma(x_i) = 0$, then allocate $\{U_i', q_i^1, q_i^2, s_i^1, s_i^2, V_i'\}$ to agent $a_{\overline{x}_i}$ and $\{D_i, D_i'\}$ to agent a_{x_i}. In addition, allocate $\{U_i, p_i^1, p_i^2\}$ to p_i, and $\{r_i^1, r_i^2, V_i\}$ to r_i.

- Finally, for every $j \in [p]$, allocate the sets $\{C_j^{L_1}, C_j^{L_2}\}$ and $\{C_j^{R_1}, C_j^{R_2}\}$ to the two shadow agents whose corresponding literals satisfy the auxiliary clause C_j.

Observe that each good is assigned to exactly one agent in the aforementioned allocation. Furthermore, each agent's bundle is connected; in particular, each shadow agent either receives a set of adjacent core and shadow goods (if the corresponding shadow variable evaluates to false under σ), or a set of adjacent auxiliary goods (if the corresponding shadow variable evaluates to true).

It is easy to verify that the utility of the special agent is equal to 3, and that of every other agent is equal to 2. Thus, the allocation is EQ1.

We will now argue that the above allocation, say A, is Pareto optimal. Suppose, for contradiction, that another allocation A' Pareto dominates A. Since the special agent and each separator agent receives all of its approved goods under A, the utilities of these agents under A and A' must be equal. Furthermore, if a main agent has a strictly higher utility under A', then by the connectedness constraint, its bundle must contain a separator good, which leads to an infeasible assignment since these goods are necessarily allocated to the separator agents. A similar argument shows that a shadow agent, too, cannot receive a higher utility under A'. Therefore, A must be Pareto optimal.

The Reverse Direction. We will now show how to recover an LNES assignment given a connected EQ1 and PO allocation, say A.

Since A is EQ1 and PO, we know from Lemma 22 that the special agent receives three approved goods and every other agent receives two approved goods under A. Thus, in particular, for every $i \in \{0, 1, \dots, p\}$, the separator goods S_i^1, S_i^2 are allocated to the separator agent t_i. Along with the connectedness constraint, this implies that for every $i \in [p]$, at least one of the main agents a_{x_i} or $a_{\overline{x}_i}$ will achieve a utility of 2 by either receiving the interval $U_i, p_i^1, p_i^2, r_i^1, r_i^2, V_i$ or $U_i', q_i^1, q_i^2, s_i^1, s_i^2, V_i'$. This, in turn, forces *at least* one pair of shadow agents—either $\{p_i, r_i\}$ or $\{q_i, s_i\}$—to obtain their utilities from the auxiliary goods.

We will now show that *exactly* one of these two pairs of agents derive their utility from the shadow goods, while the other pair meets the utility requirement though the auxiliary goods. Indeed, since there are $4p$ auxiliary goods (corresponding to p auxiliary clauses), at most $2p$ shadow agents can obtain the desired utility from the auxiliary goods. Therefore, for every $i \in [p]$, exactly one pair of shadow agents—either $\{p_i, r_i\}$ or $\{q_i, s_i\}$—are assigned shadow goods, while the other pair receives auxiliary goods. Note that this observation also shows that for every $i \in [p]$, exactly one out of a_{x_i} or $a_{\overline{x_i}}$ is assigned the dummy goods $\{D_i, D_i'\}$.

Overall, we have that one set of p main agents gets exactly two core goods each (we will refer them as the “lucky” agents), while the other set of p main agents gets two dummy goods each (the “unlucky” agents). Notice that the two main agents corresponding to a main variable cannot both be lucky, nor can both be unlucky due to the argument presented earlier.

This brings us to a natural way of deriving an LNES assignment from the allocation A. If the main agent of the positive (respectively, negative) type is unlucky, then we let $(x_i) = 0$ (respectively, $(x_i) = 1$). Furthermore, if A allocates a core good to a shadow agent, then the corresponding shadow variable is set to 0, while shadow variables corresponding to shadow agents who receive auxiliary goods are set to 1. Note that exactly $2p$ of the $4p$ shadow variables are set to 1 under this assignment and there are no conflicting assignments, implying that is indeed a valid solution to the LNES instance. This completes the proof of part (a) of Theorem 34.

To prove part (c), we first observe that the argument in the forward direction remains the same as in part (a), since the allocation constructed in the proof is EQ1 and satisfies the desired egalitarian welfare condition. In the reverse direction, it is possible that under the given allocation, say A, the special agent a_s no longer receives all three special goods. By connectedness, this means that either S_1 or S_3 is not assigned to a_s under A. Then, we can modify A to obtain another allocation, say A', that is identical to A except for the allocation of the special goods, which are all assigned to the special agent. It is easy to see that A' is connected, satisfies EQ1, and is Pareto optimal. One could now use an identical argument as in part (a) to infer a satisfying LNES assignment.

We will now proceed to proving part (b) of Theorem 34, which requires a slightly different construction.

Let be an instance of LNES. We will begin with the description of the reduced instance. We note that the construction is similar to as that of **part (a)**.

Goods: For every $i \in [p]$, we introduce one good for every core clause denoted by U_i, V_i, U_i', V_i', and two goods for every auxiliary clause denoted by C_i^L, C_i^R. We refer to U_i, V_i, U_i', V_i' as the *core* goods, and C_i^L, C_i^R as the *auxiliary* goods. Next, we introduce two goods for each shadow variable, i.e., corresponding to each of p_i, q_i, r_i, s_i, we introduce the following *shadow* goods: $p_i^1, p_i^2, r_i^1, r_i^2, q_i^1, q_i^2, s_i^1, s_i^2$. Finally, we introduce $3p$ *dummy* goods denoted by $D_1^1, D_1^2, D_1^3, \ldots, D_p^1, D_p^2, D_p^3$ and two *separator* goods S_0^1, S_0^2. Thus, the total number of goods is $m = 4p + 2p + 8p + 3p + 2 = 17p + 2$. The goods are arranged as shown in Figure 8.6.

Agents: For every main variable x_i, we will introduce two agents a_{x_i} and $a_{\overline{x}_i}$ for the two literals; these are referred to as *main agents* of the *positive* and *negative* type, respectively. For every $i \in [p]$, the agent a_{x_i} approves the goods $U_i, V_i, D_i^1, D_i^2, D_i^3$, while the agent $a_{\overline{x_i}}$ approves the goods $U_i', V_i', D_i^1, D_i^2, D_i^3$. We also introduce a *shadow agent* for every shadow variable. If p_i is a shadow variable occurring in core clause U_i and auxiliary clause C_j, then the corresponding shadow agent p_i approves the shadow goods p_i^1, p_i^2 and the auxiliary goods C_i^L, C_i^R. The valuations of the other shadow agents r_i, q_i, s_i are defined analogously. Lastly, we introduce a *separator agent* a_0 that approves the two

$U_1, p_1^1, p_1^2, r_1^1, r_1^2, V_1, U_1^0, q_1^1, q_1^2, s_1^1, s_1^2, V_1^0, \quad , U_p, p_p^1, p_p^2, r_p^1, r_p^2, V_p, U_p^0, q_p^1, q_p^2, s_p^1, s_p^2, V_p^0$

(Core and shadow goods)

$S_1, S_2, C_1^L, C_1^R, \quad , C_p^L, C_p^R$

(Separator and auxiliary goods)

$D_1^1, D_1^2, D_1^3, D_2^1, D_2^2, D_2^3, \quad , D_p^1, D_p^2, D_p^3$

(Dummy goods)

Figure 8.6: The instance used in proof of part (b) of Theorem 34.

separator goods S_0^1, S_0^2. This completes the construction of our reduction. Notice that the constructed instance is $(5,4)$-*sparse*. Before presenting the proof of equivalence, we will prove a structural result in Lemma 23.

Lemma 23. *In any* EF1 + PO *allocation, the utility of each main agent is at least* 2*, and exactly one of a_{x_i} and $a_{\overline{x_i}}$ is allocated an interval of three dummy goods that it approves.*

Proof. Assume that for the variable x_i, neither a_{x_i} nor $a_{\overline{x_i}}$ is assigned the interval fD_i^1, D_i^2, D_i^3g. Notice that these goods are only valued by a_{x_i} and $a_{\overline{x_i}}$. In this case, the maximum utility attained by any of a_{x_i} and $a_{\overline{x_i}}$ is *two* (through allocation of goods among *core clause goods*). We pick one of them arbitrarily and allocate three dummy goods (D_i^1, D_i^2, D_i^3) to it in order to obtain a pareto dominant allocation which is a contradiction. Hence, at least one of the literal agents for each variable will be assigned to the corresponding chunck of three dummy goods.

Consider the case when both a_{x_i} and $a_{\overline{x_i}}$ share only D_i^1, D_i^2, D_i^3 in some EF1+PO allocation. In this case, without loss of generality assume a_{x_i} receives only one dummy good. In this case, we can construct a new allocation where $a_{\overline{x_i}}$ is assigned all three dummy goods, and a_{x_i} receives a good u_i keeping everything else same. Again, the newly constructed allocation in pareto dominant to the original allocation as the utility of $a_{\overline{x_i}}$ increases and for all other agents utility stays the same which is a contradiction. Hence, we showed that at least one of $a_{x_i}, a_{\overline{x_i}}$ will receive an entire chunck of corresponding three dummy goods. Hence, in any EF1+PO allocation both a_{x_i} and $a_{\overline{x_i}}$ will have utility at least two. □

Lemma 23 implies that for each variable one of a_{x_i} or $a_{\overline{x_i}}$ will be allocated goods from *core clause goods* (say a_{x_i}). In order to achieve utility of at least *two*, a_{x_i} will be allocated an entire interval $u_i, p_i^1, p_i^2, r_i^1, r_i^2, v_i$ of goods (and will be allocated exactly this interval in any PO allocation). Hence, in any EF1 allocation, agents corresponding to the *shadow variables* p_i and r_i have to receive utility of at least *one* through *auxiliary clause goods*. Note that there will be at least $2p$ such *shadow agents* out of total $4p$, which need to obtain utility of at least *one* from *auxiliary goods* (in order to have EF1 allocation). Since there are at most $2p$ auxiliary goods except for (S_1, S_2), hence, each of these $2p$ agents will be allocated exactly one auxiliary goods which they value.

Hence, for *any* EF1 + PO allocation we will have the structure described in Lemma 23 and the paragraph above.

Given the structural properties of *any* EF1 + PO allocation, we omit the arguments for forward and reverse directions as the they are similar to the proof of part (a) of Theorem 34. This completes the proof for **part (b)** of Theorem 34.

Next, to show the construction for **part (d)**, we only need to make following minor change: For each *auxiliary clause* we have four auxiliary goods $C_i^{L_1}, C_i^{L_2}, C_i^{R_1}, C_i^{R_2}$ instead of C_i^L, C_i^R. Note that constructed instance in this case is $(6, 4)$-*sparse*. Given this modification, the arguments for forward and reverse directions remain similar except each *shadow agent* can now obtain a utility of 2.

8.5 Efficient Algorithms for Restricted Preference Domains

8.5.1 EQ1+NW allocations

In this section, we describe a common strategy for non-wasteful EQ1 allocations for special classes of interval valuations. We first state the theorem statements and follow that with a high level idea of our algorithms.

Theorem 35 (EQ1 and NW for binary left-extremal valuations). *ere is a polynomial-time algorithm that returns an* EQ1 *and* NW *connected allocation of a path for binary le -extremal valuations, whenever such an allocation exists.*

Theorem 36 (EQ1 and NW for binary extremal valuations). *ere is a polynomial-time algorithm that returns an* EQ1 *and* NW *connected allocation of a path for binary extremal valuations, whenever such an allocation exists.*

Theorem 37 (EQ1 and NW for binary k-interval valuations). *ere is a polynomial-time algorithm that returns an* EQ1 *and* NW *connected allocation of a path for binary k-interval valuations, whenever such an allocation exists.*

Overview of the algorithm. Consider the case of Non-wasteful EQ allocations. In this case, equitability implies same utility for each agent, and non-wastefulness implies each agent is only allocated the goods it approves. Combining these two we obtain that each agent should be allocated exactly same number of goods. Hence, given n agents and m goods, we know that each agent will receive a bundle of $\frac{m}{n}$ goods.

Now, consider the case of Non-wasteful EQ1 allocations. For such cases, it is easy to see that every agent will receive a bundle of size either $b\frac{m}{n}c$ or $d\frac{m}{n}e$. Given m, n it is straightforward to compute the number of agents with bundle of size $b\frac{m}{n}c$ (say a) and number of agents with bundle of size $d\frac{m}{n}e$ (say b).

For *le -extremal valuations* and *extremal valuations*, we initially allocate the bundles of $b\frac{m}{n}c$ goods to a agents which finishes their intervals earliest (i.e. first a agents when we sort the agents in ascending order according to their size of intervals). For the remaining agents, we allocate bundles of size $d\frac{m}{n}e$, again, in ascending order of interval lengths. During execution of algorithm, if at any stage, if we are unable to find the allocation of the sizes mentioned, we return *NO* instance, else we return the desired allocation.

We repeat a similar procedure for *k-interval valuations* except, here, when we are allocating according to the sorted order of agents, we aim to maximize the number of agents with larger sized bundles. We check the validity of such larger sized allocations by running a *sanity check* which verifies that the current allocations does not cause any future agent to lose an allocation with utility $b\frac{m}{n}c$. Similar to the previous case, we return *NO* if at some point we fail to find the allocations with given utilities. Otherwise, we return the desired allocation.

8.5.2 EQ1+PO allocations

It is easy to see that NW allocations are Pareto Optimal. In this section, we will show that for *Left-Extremal*, *Extremal* and *Interval* valuations, *Pareto Optimality* implies *Non-wastefulness*. Given this, we can conclude that EQ1+PO is equivalent to EQ1+NW for these special valuations. Hence, algorithms from Section 8.5.1 will work in the same way for finding EQ1+PO allocations.

Theorem 38 (EQ1 and PO for binary left-extremal valuations). *There is a polynomial-time algorithm that returns an* EQ1 *and* PO *connected allocation of a path for binary left-extremal valuations, whenever such an allocation exists.*

Proof. Let A be an EQ1+PO allocation which is *wasteful*. We borrow the notations from Algorithm 2. Let $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ be an ordering of agents, and $\{l_1, l_2, \ldots, l_n\}$ be the length of intervals such that $L_1 \quad L_2 \quad \ldots \quad L_n$. Let the utilities of agents under A be $\{u_1, u_2, \ldots, u_n\}$. Consider another allocation $A^{\star}$, we start allocating from leftmost good on the path in order $\mathcal{A}$ of agents such that we allocate consecutive intervals of utility u_i to each a_i for $1 \leq i \leq n-1$. Note that this is possible since each agent receives a utility of u_i in A, and $L_1 \quad L_2 \quad \ldots \quad L_n$. Notice that the partial allocation $(A^{\star})$ is non-wasteful. At this stage, we allocate all the remaining goods to the last agent a_n. Since $A^{\star}$ is non-wasteful, and utilities of first $n-1$ agents are same in A and $A^{\star}$; the utility of last agent$(u_n^{\star})$ under $A^{\star}$ is strictly greater than u_n. This contradicts the pareto optimality of A. Hence, for the case of binary left-extremal valuations, PO $\implies$ NW. □

Theorem 39 (EQ1 and PO for binary extremal valuations). *There is a polynomial-time algorithm that returns an* EQ1 *and* PO *connected allocation of a path for binary extremal valuations, whenever such an allocation exists.*

We note that a similar argument as presented in Theorem 38 works for Theorem 39.

Theorem 40 (EQ1 and PO for binary k-interval valuations). *There is a polynomial-time algorithm that returns an* EQ1 *and* PO *connected allocation of a path for binary k-interval valuations, whenever such an allocation exists.*

Proof. Let A be ab EQ1+PO allocation which is wasteful. Let $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ be an ordering of agents sorted according to the earliest finishing order of their valued interval. Let $\{u_1, u_2, \ldots, u_n\}$ be the utilities of agents according to A. Consider another allocation $A^{\star}$. Again, we start allocating from leftmost good according to the order $\mathcal{A}$ of agents. For each agent a_i for $1 \leq i \leq n$ we first allocate an interval of utility u_i, if the next good after this allocation is not valued by a_{i+1}, then we extend the interval of a_i until the good g_i such that g_{i+1} is the first good valued by a_{i+1}. We start the allocation of a_{i+1} from g_{i+1}. We repeat the same procedure for every agent. Note that it is possible to allocate in this way since each agents a_i for $1 \leq i \leq n$ received utility of u_i in A. Since $A^{\star}$ is a non-wasteful allocation, at least one agent receives utility strictly more than that in A which contradicts the pareto optimality of A. □

8.6 Conclusion and Open Problems

We initiated the study of EQ1 allocations under connectedness constraints. The pursuit of connected EQ1 allocations satisfying non-trivial efficiency guarantees resulted in computational hardness. This result motivated the exploration of two avenues for tractability: relaxing the efficiency requirement and assuming structured preferences. Some of our results found broader applicability to other fairness notions (e.g., EF1) and negatively valued items.

Going forward, it would be very interesting to map the intractability frontier for binary valuations in terms of $(a.b)$-*sparsity*. Our results establish hardness of a number of problems even under $(4, 4)$-sparsity. On the other hand, $(1, b)$-sparse instances are efficiently solvable for any b. Resolving the complexity of intermediate cases is a natural next step. On the domain restriction front, the case of *binary intervals* without the extremal structure could be of interest. Finally, extensions to *general graphs* (BCE+17) or settings with *mixed items* (ACIW19) involving goods as well as chores are interesting avenues for future research.

Bibliography

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009. (Cited on page 13)

[ABC+15] Esther M. Arkin, Aritra Banik, Paz Carmi, Gui Citovsky, Matthew J. Katz, Joseph S. B. Mitchell, and Marina Simakov. *Choice Is Hard*, pages 318–328. 2015. (Cited on pages 29 and 50)

[ABF+17] Haris Aziz, Péter Biró, Tamás Fleiner, Serge Gaspers, Ronald De Haan, Nicholas Mattei, and Baharak Rastegari. Stable matching with uncertain pairwise preferences. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 344–352, 2017. (Cited on page 91)

[ABG+16] Haris Aziz, Péter Biró, Serge Gaspers, Ronald De Haan, Nicholas Mattei, and Baharak Rastegari. Stable matching with uncertain linear preferences. In *International Symposium on Algorithmic Game eory*, pages 195–206, 2016. (Cited on page 91)

[ACIW19] Haris Aziz, Ioannis Caragiannis, Ayumi Igarashi, and Toby Walsh. Fair Allocation of Indivisible Goods and Chores. In *Proceedings of the 28th International Joint Conference on Arti cial Intelligence*, pages 53–59, 2019. (Cited on page 134)

[AD15] Yonatan Aumann and Yair Dombb. The Efficiency of Fair Division with Connected Pieces. *ACM Transactions on Economics and Computation*, 3(4):23, 2015. (Cited on page 114)

[ALMO07] David J Abraham, Ariel Levavi, David F Manlove, and Gregg O'Malley. The stable roommates problem with globally-ranked pairs. In *International Workshop on Web and Internet Economics*, pages 431–444, 2007. (Cited on pages 60, 76, 77, and 89)

[Alo87] Noga Alon. Splitting Necklaces. *Advances in Mathematics*, 63(3):247–253, 1987. (Cited on pages 5 and 114)

[Arr51] Kenneth J Arrow. *Social choice and individual values*. Yale university press, 1951. (Cited on page 2)

[Bar05] Julius B Barbanel. * e geometry of e cient fair division*. Cambridge University Press, 2005. (Cited on page 4)

[BCE+16] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel Procaccia. *Handbook of Computational Social Choice*. Cambridge University Press, 2016. (Cited on pages 1, 2, 21, and 41)

[BCE+17] Sylvain Bouveret, Katarína Cechlárová, Edith Elkind, Ayumi Igarashi, and Dominik Peters. Fair Division of a Graph. In *Proceedings of the Twenty-Sixth International Joint Conference on Arti cial Intelligence (IJCAI)*, pages 135–141, 2017. (Cited on pages 100 and 134)

[BCF+19] Vittorio Bilò, Ioannis Caragiannis, Michele Flammini, Ayumi Igarashi, Gianpiero Monaco, Dominik Peters, Cosimo Vinci, and William S Zwicker. Almost Envy-Free Allocations with Connected Bundles. In *10th Innovations in eoretical Computer Science Conference (ITCS 2019)*, volume 124, pages 14:1–14:21, 2019. (Cited on pages 100 and 101)

[BCFN19] Robert Bredereck, Jiehua Chen, Ugo Paavo Finnendahl, and Rolf Niedermeier. Stable Roommates with Narcissistic, Single-Peaked, and Single-Crossing Preferences. *arXiv preprint arXiv:1903.05975*, 2019. (Cited on pages 7, 60, and 89)

[BCL18] Sylvain Bouveret, Katarína Cechlárová, and Julien Lesca. Chore Division on a Graph. *arXiv preprint arXiv:1812.01856*, 2018. (Cited on pages 102 and 114)

[BCW13] Robert Bredereck, Jiehua Chen, and Gerhard J. Woeginger. A characterization of the single-crossing domain. *Social Choice and Welfare*, pages 989–998, 2013. (Cited on page 5)

[BCW16] Robert Bredereck, Jiehua Chen, and Gerhard J. Woeginger. Are there any nicely structured preference profiles nearby? *Mathematical Social Sciences*, pages 61–73, 2016. (Cited on page 22)

[BF02] Steven J Brams and Peter C Fishburn. Voting procedures. *Handbook of social choice and welfare*, pages 173–236, 2002. (Cited on page 2)

[BFK+17] Robert Bredereck, Piotr Faliszewski, Andrzej Kaczmarczyk, Rolf Niedermeier, Piotr Skowron, and Nimrod Talmon. Robustness Among Multiwinner Voting Rules. In *10th International Symposium of Algorithmic Game eory (SAGT)*, volume 10504, pages 80–92. Springer, 2017. (Cited on pages 4, 7, 39, 42, 43, and 47)

[BH11] Miguel A Ballester and Guillaume Haeringer. A characterization of the single-peaked domain. *Social Choice and Welfare*, pages 305–322, 2011. (Cited on page 5)

[BIT86] John Bartholdi III and Michael A Trick. Stable matching with preferences derived from a psychological model. *Operations Research Le ers*, pages 165–169, 1986. (Cited on pages 6, 60, 76, and 89)

[BJK06] Steven J Brams, Michael A Jones, and Christian Klamler. Better Ways to Cut a Cake. *Notices of the AMS*, 53(11):1314–1321, 2006. (Cited on page 114)

[BJK13] Steven J Brams, Michael A Jones, and Christian Klamler. N-Person Cake-Cutting: There May be No Perfect Division. *e American Mathematical Monthly*, 120(1):35–47, 2013. (Cited on page 114)

[BKS04] Piotr Berman, Marek Karpinski, and A Scott. Approximation Hardness of Short Symmetric Instances of MAX-3SAT. Technical report, 2004. (Cited on page 102)

[Bla48] Duncan Black. On the rationale of group decision-making. *e Journal of Political Economy*, pages 23–34, 1948. (Cited on page 5)

[Bla58] Duncan Black. *e eory of Commi ees and Elections*. Cambridge University Press, 1958. (Cited on page 3)

[BM11] Salvador Barberà and Bernardo Moreno. Top monotonicity: A common root for single peakedness, single crossing and the median voter result. *Games and Economic Behavior*, pages 345–359, 2011. (Cited on page 6)

[Bor81] J. C. Borda. *Mé moire sur les élections au scrutin*. Paris, 1781. (Cited on page 2)

[Bra08] Steven J Brams. Mathematics and democracy: Designing better voting and fair-division procedures. *Mathematical and Computer Modelling*, pages 1666–1670, 2008. (Cited on page 2)

[BSU13] Nadja Betzler, Arkadii Slinko, and Johannes Uhlmann. On the Computation of Fully Proportional Representation. *J. Artif. Intell. Res.*, pages 475–519, 2013. (Cited on pages 3, 6, 7, 20, 21, 22, 27, and 28)

[BT96] Steven J Brams and Alan D Taylor. *Fair Division: From cake-cu ing to dispute resolution*. Cambridge University Press, 1996. (Cited on page 4)

[BT00] Steven Brams and Alan D Taylor. The Win-Win Solution: Guaranteeing Fair Shares to Everybody (Japanese, Portuguese, and Spanish translations). 2000. (Cited on pages 4 and 5)

[BU09] Nadja Betzler and Johannes Uhlmann. Parameterized complexity of candidate control in elections and related digraph problems. *eoretical Computer Science*, pages 5425–5442, 2009. (Cited on page 6)

[Bud11] Eric Budish. The Combinatorial Assignment Problem: Approximate Competitive Equilibrium from Equal Incomes. *Journal of Political Economy*, pages 1061–1103, 2011. (Cited on pages 5 and 101)

[CC83] John R Chamberlin and Paul N Courant. Representative deliberations and representative decisions: Proportional representation and the Borda rule. *American Political Science Review*, pages 718–733, 1983. (Cited on pages 3, 7, 19, 22, and 42)

[CDP13] Katarına Cechlárová, Jozef Doboš, and Eva Pillárová. On the Existence of Equitable Cake Divisions. *Information Sciences*, 228:239–245, 2013. (Cited on pages 99, 113, 114, 115, and 117)

[CFK+15] Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*. Springer, 2015. (Cited on pages 15, 20, 21, and 46)

[CGS12] Denis Cornaz, Lucie Galand, and Olivier Spanjaard. Bounded Single-Peaked Width and Proportional Representation. In *Proceedings of the 20th European Conference on Arti cial Intelligence (ECAI).*, Frontiers in Artificial Intelligence and Applications, pages 270–275, 2012. (Cited on pages 6 and 21)

[Chè17] Guillaume Chèze. Existence of a Simple and Equitable Fair Division: A Short Proof. *Mathematical Social Sciences*, 87:92–93, 2017. (Cited on pages 5 and 114)

[Chè18] Guillaume Chèze. Abel and Galois Cannot Share a Cake in a Simple and Equitable Way. 2018. (Cited on page 114)

[Che19] Jiehua Chen. Computational Complexity of Stable Marriage and Stable Roommates and Their Variants. *arXiv preprint arXiv:1904.08196*, 2019. (Cited on page 60)

[CIM16] Ágnes Cseh, Robert W. Irving, and David F. Manlove. The Stable Roommates Problem with Short Lists. In *Algorithmic Game eory - 9th International Symposium, SAGT 2016, Liverpool, UK, September 19-21, 2016. Proceedings*, pages 207–219, 2016. (Cited on pages 61, 66, and 67)

[CKM+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 2016 ACM Conference on Economics and Computation (EC)*, pages 305–322, 2016. (Cited on pages 5 and 105)

[Cla06] Simon Clark. The uniqueness of stable matchings. *Contributions in eoretical Economics*, pages 1–28, 2006. (Cited on page 89)

[Coo50] Clyde H Coombs. Psychological scaling without a unit of measurement. *Psychological Review*, 1950. (Cited on page 5)

[CP12] Katarína Cechlárová and Eva Pillárová. On the Computability of Equitable Divisions. *Discrete Optimization*, 9(4):249–257, 2012. (Cited on pages 113 and 114)

[CS06] Yan Chen and Tayfun Sönmez. School choice: an experimental study. *Journal of Economic theory*, pages 202–231, 2006. (Cited on page 88)

[CS14] Peter Coles and Ran Shorrer. Optimal truncation in matching markets. *Games and Economic Behavior*, 87:591–615, 2014. (Cited on pages 4 and 61)

[CSS19] Jiehua Chen, Piotr Skowron, and Manuel Sorge. Matchings under Preferences: Strength of Stability and Trade-offs. *arXiv preprint arXiv:1902.10535*, 2019. (Cited on page 61)

[Dem12] Gabrielle Demange. Majority relation and median representative ordering. pages 95–109, 2012. (Cited on page 6)

[DF81] Lester E Dubins and David A Freedman. Machiavelli and the Gale-Shapley algorithm. *e American Mathematical Monthly*, pages 485–494, 1981. (Cited on page 66)

[DS61] Lester E Dubins and Edwin H Spanier. How to Cut a Cake Fairly. *e American Mathematical Monthly*, 68(1P1):1–17, 1961. (Cited on page 114)

[DST18] Yuan Deng, Weiran Shen, and Pingzhong Tang. Coalitional Permutation Manipulations in the Gale-Shapley Algorithm. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 928–936, 2018. (Cited on page 61)

[Eec00] Jan Eeckhout. On the uniqueness of stable marriage matchings. *Economics Le ers*, pages 1–8, 2000. (Cited on pages 77 and 89)

[EFS09] Edith Elkind, Piotr Faliszewski, and Arkadii M. Slinko. Swap Bribery. In *Proceedings of the Second International Symposium on Algorithmic Game eory, SAGT*, pages 299–310, 2009. (Cited on page 40)

[EFSS17] Edith Elkind, Piotr Faliszewski, Piotr Skowron, and Arkadii Slinko. Properties of multiwinner voting rules. *Social Choice and Welfare*, pages 599–632, 2017. (Cited on page 3)

[EL14] Edith Elkind and Martin Lackner. On Detecting Nearly Structured Preference Profiles. In *Proceedings of the Twenty-Eighth AAAI Conference on Arti cial Intelligence*, pages 661–667, 2014. (Cited on pages 7, 20, and 22)

[EL15] Edith Elkind and Martin Lackner. Structure in dichotomous preferences. In *Twenty-Fourth International Joint Conference on Arti cial Intelligence*, 2015. (Cited on pages 13 and 101)

[ELP13] Gábor Erdélyi, Martin Lackner, and Andreas Pfandler. Computational Aspects of Nearly Single-Peaked Electorates. In *Proceedings of the Twenty-Seventh AAAI Conference on Arti cial Intelligence*, 2013. (Cited on pages 6, 12, and 22)

[End17] Ulle Endriss. *Trends in Computational Social Choice*. lulu.com, 2017. (Cited on page 41)

[FHH14] Piotr Faliszewski, Edith Hemaspaandra, and Lane A. Hemaspaandra. The complexity of manipulative attacks in nearly single-peaked electorates. *Artif. Intell*, pages 69–99, 2014. (Cited on page 20)

[FSS10] Herbert Fleischner, Gert Sabidussi, and Vladimir I. Sarvanov. Maximum independent sets in 3- and 4-regular Hamiltonian graphs. *Discrete Mathematics*, 310(20):2742 – 2749, 2010. Graph Theory — Dedicated to Carsten Thomassen on his 60th Birthday. (Cited on page 48)

[G$^+$73] Allan Gibbard et al. Manipulation of voting schemes: a general result. *Econometrica*, pages 587–601, 1973. (Cited on page 61)

[GI89] Dan Gusfield and Robert W. Irving. *e Stable marriage problem - structure and algorithms*. Foundations of computing series. 1989. (Cited on page 1)

[GMT14] Laurent Gourvès, Jérôme Monnot, and Lydia Tlilane. Near Fairness in Matroids. In *Proceedings of the Twenty-First European Conference on Arti cial Intelligence*, pages 393–398, 2014. (Cited on page 113)

[GS62] David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *e American Mathematical Monthly*, pages 9–15, 1962. (Cited on pages 3, 59, and 65)

[GS85] David Gale and Marilda Sotomayor. Ms. Machiavelli and the stable matching problem. *e American Mathematical Monthly*, pages 261–268, 1985. (Cited on page 61)

[GSOS17] Begum Genc, Mohamed Siala, Barry O'Sullivan, and Gilles Simonin. Finding robust solutions to stable marriage. *arXiv preprint arXiv:1705.09218*, 2017. (Cited on pages 8, 61, 65, and 81)

[GSSO19] Begum Genc, Mohamed Siala, Gilles Simonin, and Barry O'Sullivan. Complexity Study for the Robust Stable Marriage Problem. *eoretical Computer Science*, 2019. (Cited on page 61)

[Har75] John C Harsanyi. Can the maximin principle serve as a basis for morality? A critique of John Rawls's theory. *American political science review*, pages 594–606, 1975. (Cited on page 21)

[ILG87] RW Irving, P Leather, and D Gusfield. An efficient algorithm for the "Optimal" stable marriage. *Journal of the Association for Computing Machinery*, pages 532–543, 1987. (Cited on pages 4 and 65)

[IM08] Kazuo Iwama and Shuichi Miyazaki. A survey of the stable marriage problem and its variants. In *International conference on informatics education and research for knowledge-circulating society (ICKS 2008)*, pages 131–136, 2008. (Cited on pages 3 and 60)

[IMMM99] Kazuo Iwama, Shuichi Miyazaki, Yasufumi Morita, and David Manlove. Stable marriage with incomplete lists and ties. In *International Colloquium on Automata, Languages, and Programming*, pages 443–452, 1999. (Cited on page 60)

[Ina69] Ken-ichi Inada. The simple majority decision rule. *Econometrica: Journal of the Econometric Society*, pages 490–506, 1969. (Cited on page 6)

[IP19] Ayumi Igarashi and Dominik Peters. Pareto Optimal Allocation of Indivisible Goods with Connectivity Constraints. In *Proceedings of the irty-ird AAAI Conference on Arti cial Intelligence (AAAI), 2019*, 2019. (Cited on pages 8, 102, 105, 106, 115, and 125)

[Irv85] Robert W. Irving. An Efficient Algorithm for the "Stable Roommates" Problem. *J. Algorithms*, 6(4):577–595, 1985. (Cited on pages 3 and 59)

[Irv94] Robert W Irving. Stable marriage and indifference. *Discrete Applied Mathematics*, pages 261–272, 1994. (Cited on page 60)

[Irv07] Robert W Irving. The cycle roommates problem: a hard case of kidney exchange. *Information Processing Le ers*, pages 1–4, 2007. (Cited on page 59)

[Kar72] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. 1972. (Cited on page 14)

[Kar19] Alexander Karpov. A necessary and sufficient condition for uniqueness consistency in the stable marriage matching problem. *Economics Le ers*, 178, 2019. (Cited on page 89)

[KM10] Hirotatsu Kobayashi and Tomomi Matsui. Cheating strategies for the Gale-Shapley algorithm with complete preference lists. *Algorithmica*, pages 151–169, 2010. (Cited on page 61)

[Kno10] Vicki Knoblauch. Recognizing one-dimensional Euclidean preference profiles. *Journal of Mathematical Economics*, pages 1–5, 2010. (Cited on page 95)

[Knu97] Donald Knuth. *Stable marriage and its relation to other combinatorial problems : an introduction to the mathematical analysis of algorithms*, volume 10. American Mathematical Society, 1997. (Cited on pages 66 and 85)

[LB11] Tyler Lu and Craig Boutilier. Budgeted social choice: From consensus to personalized decision making. In *Twenty-Second International Joint Conference on Arti cial Intelligence*, 2011. (Cited on pages 3 and 20)

[LMMS04] Richard J Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM conference on Electronic commerce*, pages 125–131, 2004. (Cited on page 5)

[LS18] Martin Lackner and Piotr Skowron. Consistent approval-based multi-winner rules. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 47–48. ACM, 2018. (Cited on page 42)

[MI14] Eric McDermid and Robert W Irving. Sex-equal stable matchings: Complexity and exact algorithms. *Algorithmica*, pages 545–570, 2014. (Cited on pages 62, 69, 70, 85, 86, and 87)

[ML18] Vijay Menon and Kate Larson. Robust and approximately stable marriages under partial information. In *International Conference on Web and Internet Economics*, pages 341–355, 2018. (Cited on pages 4, 61, and 91)

[Mon95] Burt L Monroe. Fully proportional representation. *American Political Science Review*, pages 925–940, 1995. (Cited on page 3)

[Mou80] Hervé Moulin. On strategy-proofness and single peakedness. *Public Choice*, pages 437–455, 1980. (Cited on page 6)

[MS10] Dániel Marx and Ildikó Schlotter. Parameterized complexity and local search approaches for the stable marriage problem with ties. *Algorithmica*, pages 170–187, 2010. (Cited on pages 4, 61, 62, 64, 71, 72, 73, and 81)

[MS19] Neeldhara Misra and Chinmay Sonar. Robustness Radius for Chamberlin-Courant on Restricted Domains. In *SOFSEM 2019: eory and Practice of Computer Science - 45th International Conference on Current Trends in eory and Practice of Computer Science, Nový Smokovec, Slovakia, January 27-30, 2019, Proceedings*, pages 341–353, 2019. (Cited on pages 7 and 39)

[MSV17] Neeldhara Misra, Chinmay Sonar, and P. R. Vaidyanathan. On the Complexity of Chamberlin-Courant on Almost Structured Profiles. In *Proceedings of the 5th International Conference on Algorithmic Decision eory ADT*, pages 124–138, 2017. (Cited on pages 7, 13, and 19)

[NR09] Muriel Niederle and Alvin E Roth. Market culture: How rules governing exploding offers affect market performance. *American Economic Journal: Microeconomics*, pages 199–219, 2009. (Cited on page 88)

[Opa79] Jaroslav Opatrny. Total ordering problem. *SIAM Journal on Computing*, pages 111–114, 1979. (Cited on page 94)

[Pet17] Dominik Peters. Recognising multidimensional Euclidean preferences. In *irty-First AAAI Conference on Arti cial Intelligence*, 2017. (Cited on page 6)

[PRZ08] Ariel D Procaccia, Jeffrey S Rosenschein, and Aviv Zohar. On the complexity of achieving proportional representation. *Social Choice and Welfare*, pages 353–362, 2008. (Cited on pages 3 and 20)

[PW17] Ariel D Procaccia and Junxing Wang. A Lower Bound for Equitable Cake Cutting. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 479–495, 2017. (Cited on page 113)

[RBN17] Ugo Paavo Finnendahl Robert Bredereck, Jiehua Chen and Rolf Niedermeier. Stable roommate with narcissistic, single-peaked, and single-crossing preferences. In *Algorithmic Decision eory - 5th International Conference, ADT 2017, Luxembourg, Luxembourg, October 25-27, 2017, Proceedings*, pages 315–330, 2017. (Cited on pages 60, 61, 64, and 68)

[Rob77] Kevin WS Roberts. Voting over income tax schedules. *Journal of public Economics*, pages 329–340, 1977. (Cited on page 5)

[Rot82] Alvin E Roth. The economics of matching: Stability and incentives. *Mathematics of operations research*, pages 617–628, 1982. (Cited on pages 4, 61, and 76)

[Rot84] Alvin E Roth. The evolution of the labor market for medical interns and residents: a case study in game theory. *Journal of political Economy*, pages 991–1016, 1984. (Cited on pages 3 and 59)

[Rot08] Alvin E Roth. Deferred acceptance algorithms: History, theory, practice, and open questions. *international Journal of game Theory*, pages 537–569, 2008. (Cited on page 3)

[RR99] Alvin E Roth and Uriel G Rothblum. Truncation strategies in matching markets—in search of advice for participants. *Econometrica*, pages 21–43, 1999. (Cited on pages 4 and 61)

[RV90] Alvin E Roth and John H Vande Vate. Random paths to stability in two-sided matching. *Econometrica: Journal of the Econometric Society*, pages 1475–1480, 1990. (Cited on page 89)

[RW86] Daniel Ratner and Manfred K. Warmuth. Finding a Shortest Solution for the N × N Extension of the 15-PUZZLE Is Intractable. In *Proceedings of the 5th National Conference on Artificial Intelligence. Philadelphia, PA, USA, August 11-15, 1986. Volume 1: Science.*, pages 168–172, 1986. (Cited on page 67)

[Sen87] Amartya Sen. *Social Choice*. London, 1987. (Cited on page 1)

[SHS18] Erel Segal-Halevi and Warut Suksompong. Democratic Fair Allocation of Indivisible Goods. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*, pages 482–488, 2018. (Cited on page 104)

[Sip] Michael Sipser. *Introduction to the Theory of Computation*, volume 2. (Cited on page 13)

[Su99] Francis Edward Su. Rental Harmony: Sperner's Lemma in Fair Division. *The American Mathematical Monthly*, pages 930–942, 1999. (Cited on pages 104 and 105)

[Suk19] Warut Suksompong. Fairly Allocating Contiguous Blocks of Indivisible Items. *Discrete Applied Mathematics*, 2019. (Cited on pages 104, 105, and 114)

[SYE13] Dmitry Shiryaev, Lan Yu, and Edith Elkind. On elections with robust winners. In *Proceedings of the International conference on Autonomous Agents and Multi-Agent Systems, (AAMAS)*, pages 415–422, 2013. (Cited on page 40)

[SYFE15] Piotr Skowron, Lan Yu, Piotr Faliszewski, and Edith Elkind. The complexity of fully proportional representation for single-crossing electorates. *Theor. Comput. Sci*, pages 43–57, 2015. (Cited on pages 6, 7, 19, 20, 21, 25, 26, and 27)

[Tam93] Akihisa Tamura. Transformation from arbitrary matchings to stable matchings. *Journal of Combinatorial Theory, Series A*, pages 310–323, 1993. (Cited on page 89)

[Tay04] Alan D Taylor. The moulin, fair division and collective welfare. *Public Choice*, pages 468–470, 2004. (Cited on page 1)

[Tho11] William Thomson. Fair allocation rules. In *Handbook of social choice and welfare*, pages 393–506. 2011. (Cited on page 4)

[TST01] Chung-Piaw Teo, Jay Sethuraman, and Wee-Peng Tan. Gale-Shapley stable marriage problem revisited: Strategic issues and applications. *Management Science*, pages 1252–1267, 2001. (Cited on pages 3, 59, 61, and 74)

[VG17] Rohit Vaish and Dinesh Garg. Manipulating Gale-Shapley Algorithm: Preserving Stability and Remaining Inconspicuous. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI*, pages 437–443, 2017. (Cited on pages 8, 61, 62, 74, and 78)

[Xia12] Lirong Xia. Computing the margin of victory for various voting rules. In *Proceedings of the ACM Conference on Electronic Commerce, (EC)*, pages 982–999, 2012. (Cited on page 40)