An Algorithmic Analysis of Deliberation and Representation in Collective Behaviour

by

Grzegorz Lisowski

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy in Computer Science

Computer Science

October 2022

Contents

Acknowledgments

Completing this thesis was a long journey and it would not have been possible alone. First, I would like to thank my supervisor, Paolo Turrini. Thank you for your guidance and your attention to detail. I am grateful for constantly reminding me of the big picture of research and of the importance of communicating my findings, when I was stuck in technical details.

Throughout my PhD studies I had an honour to work with many brilliant minds. Dmitry Chistikov, thank you for your inspiring curiosity and passion for research. Thank you for tea breaks in your office and discussing problems just because they are interesting, even if they were not aimed at writing a paper. Sylvie Doutre, thank you for your warm welcome in Toulouse, your deep knowledge of the literature, and bright ideas. Lawqueen Kanesh, thank you for long hours at a whiteboard, which would always leave me optimistic, even if there were mistakes involved. Umberto Grandi, thank you for your continuous support, from my very first results as a MSc student, until now. I appreciate that a lot. Paul Harrenstein, thank you for showing me how to maintain precision and rigour while developing creative arguments. Mike Paterson, thank you for showing me how difficult proofs can be pretty and easy to understand. M.S Ramanujan, thank you so much for teaching me how to think about algorithms, which will be vital in my future work.

Not the least, I would like to thank all dear friends, who were always there for me, across countries, continents, and time-zones. Thank you for numerous visits in Phantom Coach (and other reasonably priced establishments), and all of the climbing, cycling, and hiking trips. Thanks for bumping into each other at conferences, sharing your research, and your great effort to nurture long standing friendships. Your warm welcome in Amsterdam, Detroit, Düsseldorf, London, Toulouse, Paris, and Warsaw means a lot to me. It would be so hard without you.

Thanks Mum and Dad!

Julia: 

Declarations

This work has been composed by myself and has not been submitted for any other degree or professional qualification.

- Work in Chapter 4 has been published in the Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS-2021), in collaboration with Paul Harrenstein, M.S Ramanujan, and Paolo Turrini.
- Work in Chapter 5 has been published in the Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS-2022), in collaboration with M.S Ramanujan and Paolo Turrini.
- Work in Chapter 6 has been published in the proceedings of the European Conference on Multi-Agent Systems (EUMAS-2022).
- Work in Chapter 7 has been published in published in the Proceedings of AAAI Conference on Artificial Intelligence (AAAI-2020), in collaboration with Dmitry Chistikov, Mike Paterson, and Paolo Turrini.
- Work in Chapter 8 has been published in the Proceedings of AAAI Conference on Artificial Intelligence (AAAI-2023). It was developed in collaboration with Umberto Grandi, Lawqueen Kanesh, M.S Ramanujan, and Paolo Turrini.

Abstract

The selection of a nominee by a group of players in the process of selecting a winner is present in many contexts. In sports, it is a major strategic problem to select the best team members. Crucially, in politics, this problem is essential for the process of primaries. There, factions decide which of their candidates should take part in the elections.

We study the strategic behaviour of coalitions from the game-theoretic perspective. More precisely, we analyse the existence of a pure Nash equilibrium in the games capturing the strategic nomination problem. First, we adapt the well-known Hotelling-Downs model, capturing the strategic behaviour of political parties in primaries. Subsequently, we explore this problem for tournament-based rules. There, winners are chosen based on the pairwise comparisons between candidates. First, we study the setting of knockout-tournaments. Next, we investigate tournaments, in which participants do not compete in rounds. For each of these mechanisms, we analyse the computational complexity of checking the existence of a pure Nash equilibrium.

Nominee selection can also be influenced by the deliberation between the voters. To account for that, we investigate the complexity of checking the convergence of a synchronous, threshold-based protocol. There, in every time step all agents update their opinion if the strict majority of their influencers disagrees with them. Furthermore, we explore computational aspects of majority illusion. This phenomenon occurs when a large number of agents in a network perceives the opinion, which is a minority view, as the one which is held by the majority of agents. We study the problem of checking the possibility of assigning opinions to agents, so that it holds for a large fraction of them. We further address the complexity of checking the possibility of eliminating the majority illusion by changing a small number of edges in a social network.

Abbreviations

- NE: Nash equilibrium.
- DSE: Dominant strategy equilibrium.
- CW: Condorcet winner rule.
- US: Uncovered set rule.
- SE: Single-elimination.
- s.b.a: Spanning binomial arborescence.
- SN: Social network.
- SCC: Strongly connected component.
- DAG: Directed acyclic graph.
- OD: Opinion diffusion.
- SU: Synchronous update.
- nop: No operation.
- ND: Neighbourhood diversity.
- FPT: Fixed parameter tractable.
- tw: Tree width.
- CNF: Conjunctive normal form.

- cw: Clique width.
- VC: Vertex cover number.

Chapter 1

Introduction

1.1 Motivation

The behaviour of a group of agents is one of the crucial aspects of the analysis of the mechanisms for collective decision making. Indeed, when considering a choice made by a large number of individuals based on their preferences, the impact of a single participant on the result is often marginal. One of the examples of such a situation is presidential elections. In such a case, it is not possible for an individual to strategise in order to change the outcome of a selection procedure. Nevertheless, it might be possible for a group of agents (such as a coalition of voters, or of candidates), to successfully influence the selection mechanism. In particular, a number of candidates, such as, for instance, a political party, can select one of its members to compete in the elections, in order to have their representative selected. Such considerations have been explored in recent multi-agent systems literature (see, e.g., Faliszewski et al. [2016]; Misra [2019]; Kondratev and Mazalov [2019]). One can also not underestimate the impact that deliberation between decision makers has on the outcome of a selection procedure. Additionally, the possibility of fluctuation in voters' preferences, based on their exchange of opinions, can have a strong impact on the chosen option (see, e.g., Faliszewski et al. [2022]; Corò et al. [2019]; Miller [1980]). This can result in the change of the optimal choice of a nominee of a group of candidates.

One of the contexts in which groups strategically select their nominee is the process of *primaries*, which constitute one of the main aspects of modern democracy. As we can observe in the case of the US elections, the selection of candidates by political parties not only has a strong impact on which of them has their nominee as a winner, but also on the way in which future policies are made. The importance and complexity of primaries has also become apparent in recent UK politics during the process of selecting a new prime minister. As such, the study of this process is crucial, both from political science and multi-agent systems perspectives (see, e.g., Borodin et al. [2019]).

While many factors play a part in primaries, the *strategic behaviour* of parties is especially interesting. In fact, winning an electoral competition, or attracting as many voters as possible, can be seen as the main goal of the parties participating in elections. As such, it is natural to assume that it is the factions' objective to nominate a candidate attracting the largest portion of the electorate. It is also worth mentioning, that especially in multi-party systems, the choices of goal-oriented parties depend on the nominations of their competitors. For instance, selecting a left-wing candidate, instead of a right-wing nominee, might have a different impact, depending on the orientation of candidates chosen by other parties.

As mentioned before, another aspect which might affect the strategic choices that parties make is the *deliberation* between voters. The phenomenon of the spread of opinions in the society, i.e., of *opinion diffusion*, is especially important in times of increasing social media impact, through platforms such as Facebook or Twitter, on peoples' views. In fact, the possibility for voters to convince each other to change their willingness to cast their vote for a specific candidate, or to change their political views, might mean that the optimal choice of a nominee changes over time. The importance of this factor became apparent, for instance, when the use of Facebook users' data was used to affect the US presidential campaign in 2016.

It is also important to note that the deliberation between voters is highly influenced by how they are *connected* in a social network. For instance, *homophily*, i.e., the tendency of people to form connections to those who are similar to themselves, might result in the reinforcement of their opinions. Hence, taking into account with whom voters communicate their opinions is of utmost importance when predicting how their preferences will change over time. Such considerations have implications on estimating who would be the best party nominee at the time of the elections. We note that these factors are closely connected to a number of misconceptions that voters might have about the way in which their peers think. One of such phenomena, which is present in a large number of social contexts, is the *majority illusion* (see, e.g., Lerman et al.). It occurs when a large number of voters hold the view that the majority of their peers have a different opinion than the one which is adopted by the overall majority. We say that a voter, for which this property holds, is under the *majority illusion*. This phenomenon has a strong influence on opinion diffusion, as it implies that a large number of voters might tend to adopt the minority view. For an overview of connections between social choice theory and social network analysis, see, e.g., Grandi [2017].

In this thesis, we study the problem of strategic nominee selection from the *game-theoretic* perspective. For a number of different social choice mechanisms we formulate the competition between parties as a strategic game, for which we address the problem of finding pure Nash equilibria. In particular, we study an extention of the well-known *Hotelling-Downs* model (Hotelling [1929]). There, voters and candidates are located in a metric space, which indicates their views. Subsequently, voters support a candidate, who is closest to them. Furthermore, we study the problem of finding pure Nash equlibria in the context of *tournament* based selection mechanisms, where the choice of a winner follows from pairwise comparisons between candidates, which we also call *tournament solutions.* In such a comparison one can take as fact that that the stronger candidate in a pair is preferred to the weaker one by the majority of voters. The problem of selecting candidates by groups participating in a tournament is also important from the perspective of sports, where teams, or coalitions of players select a player to participate in the competition. An interesting type of a tournament is a *knockout tournament*, where candidates compete in rounds. For an overview of connections between social choice theory and tournaments see, e.g., Williams and Moulin [2016] and Brandt et al. [2016a].

Subsequently, we study theoretical aspects of opinion diffusion, and of majority illusion. First, we consider the problem of determining whether an opinion diffusion process terminates in a given social network. We note that this problem is important, with respect to predicting voters' views after their deliberation. Subsequently, we investigate theoretical problems regarding majority illusion.

The focus of this thesis is on the study of nominee selection by competing groups, and of related phenomena occurring in social networks, from the *computational complexity* perspective. First, for games by which we model competitions between groups, we address the problem of whether it is possible to efficiently compute a pure Nash equilibrium, if it exists. This aspect is crucial with respect to determining the possibility of predicting the outcome with limited computational resources. Further, establishing the complexity of the proposed computational problems related to social networks can help to understand, if we can predict voters' opinions after deliberation in a short time. An overview of connections between social choice theory and computational complexity can be found in, e.g., Hemaspaandra [2018].

We note that, while our results are motivated by real-world applications, they are theoretical. Thus, they do not take into account all of the factors, which might influence the phenomena that we study. As such, they cannot be immediately used to predict outcomes of primaries, even though they provide insights useful towards the understanding of this process.

The motivation of the thesis is illustrated in the following example.

Motivating Example. In a University, the Chancellor is about to be elected. It is the duty of a committee, composed of 11 senior professors, to select a candidate among the nominees of four faculties: Social Science (A), Mathematics (B), Engineering (C), and Sciences (D). Social Science is considering to select candidate a_1 or a_2. Mathematics have two potential candidates, b_1 and b_2. Also, Engineering can choose between c_1 and c_2, while Sciences can select d_1 or d_2. The most pressing issue for the institution is how to allocate newly received substantial funding. All members of the committee agree that it is important from the perspective of the University's mission to support underfunded, theoretical projects. However, allocating more funding into cooperation with the industry would help the University receive even more funding in the future. It is known to all faculties, and to all committee members, to what extent the candidates agree with these options. It is also not a secret that some committee members are more theory-oriented than others.

The individuals' opinions can be presented on a line, on which every candidate, and every voter, is located. Such a line is shown in Figure 1.1. There, the willingness of a person to support theoretical projects is represented by how far to the left a person is located on the line. Correspondingly, their inclination to choose funding allocations preferential to industry collaborations is captured by how far to the right they are located. Below the line we show the opinion of a particular candidate, while above, how many committee members locate on a particular point on the spectrum.

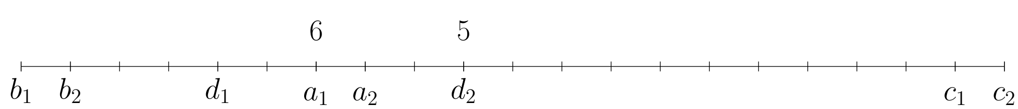

Figure 1.1: Positions of candidates of all faculties, i.e., A, B, C, and D, as well as the distribution of voters, with the number of committee members located at a point of the line indicated above the line.

Observe that the location of a voter on the spectrum determines their preferences over the candidates. To account for that we assume that given a pair of candidates, a voter finds the one which is closer to them more preferable. This observation gives us the following preference profile.

- **6 voters**: $a_1 \succ a_2 \succ d_1 \succ d_2 \succ b_2 \succ b_1 \succ c_1 \succ c_2$
- **5 voters:** $d_2 \succ a_2 \succ a_1 \succ d_1 \succ b_2 \succ b_1 \succ c_1 \succ c_2$

This profile gives rise to what we call a *tournament*, which is a directed graph over the set of candidates, where an edge from a candidate i, to another candidate j, indicates that the majority of voters prefer i to j. To account for the electoral competition between faculties, we consider what we call a *coalitional structure*, which is a tournament, as well as a partition of candidates into groups. The coalitional structure, based on the strategy profile, which we consider in this example, is shown in Figure 1.2. It is clear that a_1 wins against all other potential candidates in the majority contest, thus Social Science selects them as their nominee. Furthermore, believing that their candidate is going to lose in any case, Sciences select d_1, which is more popular among voters than d_2.

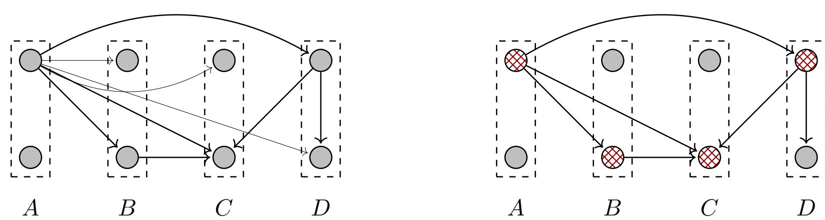

Figure 1.2: On the left, the coalitional structure with faculties A, B, C, and D, each with two members. On the right, the relation between faculties' nominees. Vertices in red represent the chosen candidates. For clarity, only selected edges in the tournament are presented.

However, without letting the faculties know, committee members let each other know who they think is their favourite candidate. Figure 1.3 represents among which pairs of voters the communication took place. There, an edge between two members indicates that they exchanged their opinions. Also, red vertices represent the voters, whose favourite candidate is d_2, while blue vertices correspond to those who believe that a_1 is the best candidate.

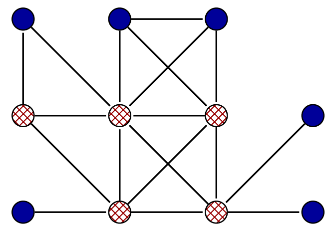

Figure 1.3: Distribution of opinions among voters before any communication. Red vertices correspond to voters whose preferred candidate is d_2, while the blue vertices correspond to those who would like to vote for a_1.

Voters revise their opinions based on their deliberation. Each of them decides to adopt a view if it is held by the strict majority of peers they communicate with. But then, as all a_1 supporters observe that their peers' preferred option is d_2, they also decide to vote for them. As they now agree that d_2 is the best candidate, they also revise their opinion regarding the key issue, agreeing with d_2. Figure 1.4 depicts the opinions of voters after the deliberation.

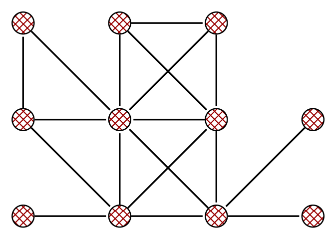

Figure 1.4: Connections between voters, and the distribution of opinions between voters after deliberation. Red vertices correspond to those whose favourite candidate is d_2.

Now we can see how the deliberation between voters affected their views. After changing their opinions, all of the committee members locate themselves at the position of d_2. The changed views of voters regarding the key issue is shown in Figure 1.5.

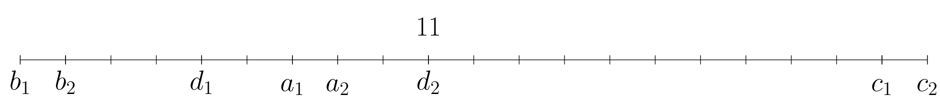

Figure 1.5: Positions of candidates of all faculties, as well as the distribution of voters, after their deliberation, with the number of committee members located at a point of the line indicated above the line.

The revised opinions of voters induce a radically different preference profile to the initial one.

- **11 voters:** $d_2 \succ a_2 \succ a_1 \succ d_1 \succ b_2 \succ b_1 \succ c_1 \succ c_2$

As d_1 and a_1 were nominated by the faculties, and a_1 is preferred to d_1 by all of the voters, ultimately a_1 is selected as the Chancellor. So, Social Science wins the election, as a_1 is their nominee. Notice, however, that in the revised profile, d_2 would win against a_1. Hence, if Sciences took the possibility of discussion among committee members into account and nominated the candidate d_2, they would have won the election.

1.2 Research Questions

Here, we outline our main research questions which motivate particular chapters of this thesis.

Hotelling-Downs Framework for Strategic Nominee Selection.

- *Under which circumstances do pure Nash equilibria exist in the Hotelling-Downs model, oriented at capturing nominee selection?* The solution concept of a Nash equilibrium is of high importance with respect to predicting agents' behaviour in a game, as it helps us to identify strategy profiles in which none of them has an incentive to change their choice. Thus, in the context of elections, checking its existence can help us predict which candidates will be nominated by parties.

- *What is the computational complexity of checking if a pure Nash equilibrium exist in particular elections?* As we show that, in certain elections, there are no pure Nash equilibria in the model we consider, it is natural to ask if they can be computed efficiently when they exist.

- *Is the problem of checking the existence of a pure Nash equilibrium easier, when only two parties participate in elections?* The case in which only two parties compete is important from the perspective of modern politics, for instance, with respect to US presidential elections. As such, establishing algorithmic results for this special case is of high interest.

These questions are addressed in Chapter 4.

Group Choices in Knockout Tournaments.

- *What is the computational complexity of checking the existence a pure Nash equilibrium in a given tournament?* The analysis of the properties of this decision problem is important, as argued in the context of our extension of Hotelling-Downs model.
- *Is there a difference between the complexity of checking the pure Nash equilibrium existence in games in which coalitions strive to win a tournament only, and those in which they aim at reaching a high round?* It is natural to assume that, depending on the nature of a competition, coalitions are incentivised not only to win, but also to perform well, even if their nominee is not the winner. As such, we ask whether the problem which we consider has a different computational complexity in such a case.
- *Is the case in which coalitions choose one candidate for the duration of the tournament, and the case in which they can choose a different nominee at every round, different from the algorithmic perspective?* As in a knockout tournament participants compete in rounds, coalitions might benefit from switching their nominee during the tournament. It is therefore natural to ask, if the algorithmic properties of competitions change when coalitions are allowed such changes.

These questions are addressed in Chapter 5.

Group Choices in Tournament Solutions.

- *Is there a difference between the computational complexity of checking the existence of a pure Nash equilibrium between knockout tournaments and tournament which are not played in rounds?* It is important to notice that knockout tournaments, studied in Chapter 5, are structured as a *binary tree*. This feature might help in designing algorithms for the problem of strategic nominee selection. It is therefore natural to ask whether this problem is more computationally complex in the context of tournaments which are not played in rounds.
- *Is the problem of checking the existence of an equilibrium different for particular rules determining the winners of a tournament?* Multiple methods of selecting a set of winners of a tournament have been proposed in the literature,

which differ with respect to their crucial properties. We are therefore interested in checking if the problem of strategic nominee selection differs between tournaments with different rules, with regards to computational complexity.

These questions are addressed in Chapter 6.

Convergence of Opinion Diffusion.

- *What is the complexity of checking if an opinion diffusion protocol terminates?* As we have noted, it is useful for a group to know what are voters' preferences *after deliberation*. Ideally, coalitions would prefer to base their choices on stable preferences held by voters. It is therefore natural to ask what is the complexity of checking if the opinion diffusion protocol converges.

- *Are there natural classes of networks for which the problem we consider is tractable?* Even though we show that the problem of convergence of the opinion diffusion protocol which we study is not tractable, we are interested in finding natural instances for which it can be computed efficiently.

These questions are addressed in Chapter 7.

Algorithmic Analysis of Majority Illusion.

- *What is the complexity of checking, for a given social network, if there is a distribution of opinions, such that a large number of voters is under majority illusion?* One can easily notice that there are social networks which allow for a distribution of opinions, such that all voters are under majority illusion (see Figure 1.3 for an example of such a network). On the other hand, in some cases, such as in networks in which all voters are disconnected, no voter is under illusion under any distribution of opinions. This motivates the question of how hard it is to check whether the structure of a social network allows for a large number of voters to be under illusion.

- *How hard is it to check if it is possible to sufficiently reduce the number of voters under majority illusion, by changing a small number of connections between them?* From the engineering point of view, it is interesting if it is possible to fix the problem of a large number of voters being under the majority illusion. We are particularly interested in the possibility of reducing the number of such voters by changing the set of their peers which they follow in order to amend their perception of the distribution of opinions.

These questions are addressed in Chapter 8.

1.3 Publications

The results of this thesis have been published, or are under review. Below we specify the details of the corresponding papers.

- "Convergence of Opinion Diffusion is **PSPACE**-Complete", coauthored with Dmitry Chistikov, Mike Paterson, and Paolo Turrini, was published in the Proceedings of AAAI Conference on Artificial Intelligence (AAAI-2020) Chistikov et al. [2020]. Results contained in that paper are presented in Chapter 7.
- "A Hotelling-Downs Framework for Party Nominees", coauthored with Paul Harrenstein, M.S. Ramanujan, and Paolo Turrini, was published in the Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS-2021) Harrenstein et al. [2021]. The results of this paper appear in Chapter 4.
- "Equilibrium Computation For Knockout Tournaments Played By Groups", coauthored with M.S. Ramanujan, and Paolo Turrini, was published in the Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS-2022) Lisowski et al. [2022]. The results of this paper appear in Chapter 5.
- "Strategic Nominee Selection in Tournament Solutions" was published in the proceedings of the European Conference on Multi-Agent Systems (EUMAS-2022). The results shown in this paper are presented in Chapter 6.
- "Identifying and Eliminating Majority Illusion in Social Networks", coauthored with Umberto Grandi, Lawqueen Kanesh, M.S. Ramanujan, and Paolo Turrini, was published in the AAAI Conference on Artificial Intelligence (AAAI-2023). The results of this manuscript are contained in Chapter 8.

1.4 Thesis Outline

In Chapter 2, we begin the thesis with an overview of the literature related to particular parts. Then, in Chapter 3, we provide the definitions of basic concepts which we use in subsequent chapters.

Chapter 4 is devoted to the extension of the Hotelling-Downs model which we propose. We start with showing that there are instances of these games, which do not admit any pure Nash equilibrium. Moreover, in the general case, we demonstrate that checking if such an equilibrium exists is **NP**-complete. Further, we demonstrate several cases in which an equilibrium is guaranteed to exist. We further demonstrate that in elections in which only two parties participate, checking the existence of a pure Nash equilibrium is possible in linear time.

Subsequently, in Chapter 5, we study the problem of strategic nominee selection in the context of knockout tournaments. There, we study the complexity of checking the existence of a pure Nash equilibrium in a number of cases. We differentiate between the scenario in which groups only nominate one competitor, and an alternative scenario in which they select their representative to compete against a specific group. We further distinguish between competitions in which groups only strive to win and those in which they aim at reaching a high round of a tournament. For all of the cases, we show that the computational problem we consider is tractable, by providing quasi-polynomial or polynomial time algorithms.

Then, in Chapter 6, we contrast the results shown in the context of knockout tournaments and the tournaments which are not played in rounds. In particular, we study the selection method in which only the chosen player which beats all other selected candidates (if one exists) is a winner. Then, we investigate the properties of the *Uncovered Set* rule in the context of tournaments played by coalitions. For both of these selection methods, we show that checking the existence of a pure Nash equilibrium is **NP**-complete, while for the Uncovered Set rule it is intractable even to check if a coalition can win given some choices of other groups.

Further, in Chapter 7, we focus on the problem of convergence of the opinion diffusion protocol in which, in every time step, all agents change their (binary) opinion if the strict majority of the agents they communicate with holds a different view than themselves. We show that, in the general case, the problem of checking if this protocol converges for a given input is **PSPACE**-complete. However, in some restricted cases, such as in acyclic graphs, this problem is solvable in polynomial time. In this chapter we assume that networks are *directed*, i.e., that it is possible for an individual not to follow another voter, who follows them. This assumption is realistic in the context of social media, such as Twitter or Instagram. We also assume that agents are not connected to themselves in a network, which is the case in social media, such as Facebook.

In the following chapter, Chapter 8, we investigate the algorithmic aspects of social networks in which a given fraction of voters is under the majority illusion. First, we study the problem if, for a given network, there exits an assignment of opinions to agents such that at least a specified fraction of them is under the illusion. Subsequently, we analyse the problem of checking the possibility of changing the connections between agents in a limited way in order to ensure that less than a specified fraction of them is under illusion. We show that these problems are **NP**-

complete. We further provide several parametrised complexity results for both of these problems. In this chapter, we consider undirected networks. We note that our computational hardness results are immediately transferable to the case of directed networks. We aim at the study of parametrised complexity of such networks in future research.

Finally, in Chapter 9, we conclude and indicate avenues for further research.

Chapter 2

Literature Review

In this chapter we discuss the literature relevant to particular parts of this thesis. In Section 2.1, we start with discussing the results relevant to our extension of the *Hotelling-Downs model*. Subsequently, in Section 2.2, we provide an overview of literature on *tournaments*. Finally, in Section 2.3, we discuss literature related to selected aspects of *social network analysis* and *opinion diffusion*, as well as *majority illusion*.

2.1 Hotelling-Downs Model

Applications of the Hotelling-Downs Model. The Hotelling-Downs model has been widely studied and applied to various contexts. Apart from the natural application in the analysis of the political debate, or location of retail points, it has its application in problems such as *brand positioning* (for a broad overview, see, e.g., the highly impactful Stokes [1963] and Eiselt et al. [1993], as well as Eiselt [2011] for a survey). Furthermore, much research has been devoted to lifting the assumptions made in the original model. Such modifications include capturing scenarios with multiple players and voting rules (see, e.g., Eaton and Lipsey [1975]; Bilò et al. [2020]; Sengupta and Sengupta [2008]) or dimensions in the metric space (e.g., Veendorp and Majeed [1995], see also Eiselt [2011]). In the field of algorithmic game theory, Feldman et al. [2016b] have analysed the case in which candidates attract voters only in a limited range. In the context of voting, Brusco et al. [2012] have looked at an application of the model with employment of the plurality with the run-off rule. It is worth noting that the work on selections made by parties restricted to intervals (Sabato et al. [2017]) is closest to the extension of the Hotelling-Downs model which we present in this thesis, even though it presents important differences in terms of equilibrium existence and algorithmic analysis. We further discuss this reference in Chapter 4.

Voronoi Games. In algorithmic game theory, *Voronoi games* feature players selecting points in a given space, with their utility being equal to the number of points in the space for which their selection is the closest. Voronoi games have been studied as sequential decision problems (see, e.g., Ahn et al. [2004]; Bandyapadhyay et al. [2015]), where two players select their (potentially multiple) locations in rounds. In the simultaneous variant, which is closer to the setup which we study in this thesis, Durr and Thang [2007] show that checking if a Nash equilibrium exists is **NP**-complete, although studying games played on arbitrary graphs and using more complex computational gadgets. Furthermore, Mavronicolas et al. [2008] provide a characterisation of Nash equilibria in games played on cycle graphs. Also, Fournier [2019] considers a setting in which consumers (our voters) are distributed non-uniformly, but players are allowed to position themselves anywhere on a graph. In a related contribution Boppana et al. [2016] consider Voronoi games with restricted positioning, but on a different spatial domain, namely a k-dimensional unit torus. Similarly, Núñez and Scarsini, in a series of works (Núñez and Scarsini [2017, 2016]), study players with limited available positions from a finite set of locations. In contrast to our model, the action spaces are the same for all of the players. They then show that Nash equilibria exist in games with a large number of players.

Facility Location. *Facility location* is an important problem connected to our analysis of primaries, where a planner selects the location of various facilities to satisfy as many agents as possible, given their positioning. This setup, originating from Moulin [1980], has been extensively studied in the social choice literature. In particular, Feldman et al. [2016a] have considered how to locate facilities when participants can strategically misrepresent their position in order to benefit from the planner's decisions.

Strategic Candidacy. Our results closely linked to strategic voting, (see e.g., Meir [2018]), the emerging area of computational social choice where participants may misrepresent their preferences to potentially manipulate the result of an election and, in particular, strategic candidacy. In the typical settings of strategic candidacy (see, e.g., Brill and Conitzer [2015]; Dutta et al. [2001]; Eraslan and McLennan [2004]), candidates are equipped with preferences over their opponents, and are allowed to step down to let their favourite rival win. In some models (e.g., Obraztsova et al. [2015]; Elkind et al. [2015]) participation in the elections incurs a cost, in which case it might be beneficial for candidates to simply abstain if they cannot themselves win.

2.2 Algorithmic Analysis of Tournaments

Our results regarding nominee selection have links to multiple research lines in the computational analysis of tournaments. There, in the context of social choice theory, particular attention has been paid to stable solutions (Brandt et al. [2016b]), and to restricted subclasses displaying desirable properties (Brandt et al. [2018]). The existence of well-behaved solutions has also motivated the study of the complexity of their computation (as in e.g., Brandt and Fischer [2008]; Brandt et al. [2010, 2018]).

Along the lines of strategic candidacy, Kondratev and Mazalov [2019] recently studied, in the context of tournament solutions, how politicians can form a coalition so that a representative of their group is elected. Further, Faliszewski et al. [2016] conducted a complexity analysis of how parties can win elections based on the voters' preferences over the set of all potential candidates. There, political parties select their representatives to compete in the elections, which are based on the plurality rule. In their study, the investigation was limited to checking if a party has a necessary, or a possible winner, leaving the study of game theoretic solution concepts open in this scenario.

Further, our results are directly relevant to the problem of *tournament manipulation*, for example understanding how the seeding can be manipulated to force a particular winner, a problem extensively studied in the literature (by e.g., Vu et al. [2009]; Aziz et al. [2014]; Chatterjee et al. [2016]; Vassilevska Williams [2010]; Hazon et al. [2008]; Kim et al. [2017]; Aziz et al. [2018]; Konicki and Williams [2019]) or whether a pair of players can reverse their comparison in order to make one of them the winner of the tournament (see, e.g., Altman et al. [2009]).

Our research is also related to the analysis of possible and necessary winner problems in the context of partial tournaments Aziz et al. [2015]. There, the problem considered tournaments with a partial information about the results of pairwise contests, and whether an option is a winner in some or in all completions of the partial tournament.

2.3 Social Network Analysis

Here, we discuss the literature relevant to our analysis of the convergence of an opinion diffusion protocol, as well as of majority illusion.

Social Influence Models. The graph-like structure of social networks has attracted interest in computer science, with studies of the influence weight of nodes in the network (Kempe et al. [2005]) and the properties of the influence function (Grabisch and Rusinowska [2010]). Social influence has been widely analysed in the social sciences, from the point of view of strategic behaviour (Isbell [1958]) and its implications for consensus creation (de Groot [1974]) and cultural evolution (Axelrod [1997]).

If the social networks are modelled as undirected, rather than directed, graphs, it has long been known that convergence takes at most a polynomial number of steps under majority updates (Chacc et al. [1985]). In these models, **PSPACE**-hardness results have only been shown for more powerful *block sequential* update rules (Goles et al. [2016]).

It is worth noting that convergence is a **PSPACE**-complete property in various models related to the one studied in this thesis, notably directed discrete Hopfield networks (Orponen [1993]) and Boolean dynamical systems (see, e.g., Barrett et al. [2003, 2007]). Hardness in these results (and their strengthenings, as studied by Ogihara and Uchizawa [2017], Rosenkrantz et al. [2018], and Kawachi et al. [2019], crucially depends on the availability of functions that *identify* 0 and 1 (see the discussion of the ingredients for the hardness proofs in Chapter 7). In the protocol we consider, opinion diffusion is instead based on self-dual functions, where flipping all inputs to a self-dual function always leads to flipping its output. In other words, in the setting we consider the diffusion protocol is symmetric with respect to opinions held by agents.

Let us further discuss the relation between the **PSPACE**-hardness results on diffusion protocols known in the literature, and the protocol which we study in this thesis. Whilst Kosub [2008] shows the **NP**-completeness of deciding the existence of a fixed-point configuration if *all* self-dual functions are available, our update rule, in comparison, is monotone (i.e., it has no negation). Moreover, sparse graphs of bounded indegree — with each agent having up to six influencers — suffice for our proof of **PSPACE**-hardness. In the related model of cellular automata, known results show that majority is "arguably the most interesting" local update rule (Tosic [2017]).

Opinion Manipulation Models. Our work is directly related to computational models of social influence, notably the work of Auletta et al. [2020], where networks and initial distribution of opinions are identified, such that an opinion can become a consensus opinion following local majority updates. In the context of our research on majority illusion, it is important to observe that when all nodes are under majority illusion, a synchronous majoritarian update causes an initial minority to evolve into a consensus in just one step. Furthermore, Auletta et al. [2015, 2017] study the possibility of a minority colour being adopted by the majority of agents. We note, however, that the problem we investigate in Chapter 8, i.e., of whether it is possible for a given fraction of agents to be under majority illusion in a given network, for some colouring, is substantially different from these investigations. To see this, note that it is possible for the minority colour to be adopted by the majority in one step even if only a small fraction of agents is under majority illusion (e.g., in networks with an odd number of vertices, where three of them form a clique, while the rest has degree 0). Let us further notice that the problem studied by Auletta et al. [2015] is solvable in polynomial time for large minorities, which is not the case for the problem we consider.

Other notable models include the work of Doucette et al. [2019] who studied the propagation of possibly incorrect opinions with an objective truth value in a social network, and the stream of papers studying the computational aspects of exploiting (majoritarian) social influence via opinion transformation (Bredereck and Elkind [2017]; Auletta et al. [2020, 2021]; Castiglioni et al. [2020]). Further, control of collective decision-making (Faliszewski and Rothe [2016]) is an important topic in algorithmic mechanism design: the difficulty of establishing whether manipulation is a real threat is paramount for system security purposes.

Deliberative Democracy and Social Choice. Opinion diffusion underpins recent models of deliberative democracy, in terms of delegation (Dryzek and List [2003]), representation (Endriss and Grandi [2014]), and stability (Christoff and Grossi [2017]). Formal models of democratic representation build on an underlying consensus-reaching protocol de Groot [1974]; Brill [2018]. Social networks have also become of major interest to social choice theory, with propositional opinion diffusion Grandi et al. [2015] emerging as a framework for social choice on social networks Grandi [2017]. Our research aligns with the work in computational social choice, in particular strategic voting [Meir, 2018] and iterative voting (e.g., Meir et al. [2017]; Reijngoud and Endriss [2012]), where decision-making happens sequentially. Of relevance are also the recently found connections between iterative voting and social networks (Wilczynski; Baumeister et al. [2020]).

Network Manipulation. Another important research line, which is related to our study of majority illusion, has looked at how to transform a social network structure with applications in the voting domain. Wilder and Vorobeychik, e.g., studied how an external manipulator having a limited budget can select a set of agents to directly influence, to obtain a desired outcome of elections. In a similar setting, Faliszewski et al. [2018] studied "bribes" of voters' clusters.

Distortions in Social Networks. We further note that there are multiple paradoxical effects in social networks which are related to our work. For instance, the *friendship paradox*, according to which, on average, individuals are less well-connected than their friends (see, e.g., Hodas et al. [2013]; Alipourfard et al. [2020]). Exploiting a similar paradox, Santos et al. [2021] recently showed how false consensus leads to the lack of participation in team efforts.

Chapter 3

Preliminaries

In this chapter we present basic notions used throughout the thesis. We commence with providing definitions of fundamental concepts explored in *game theory*, including *pure Nash equilibrium*, which is crucial to our analysis of strategic behaviour in nominee selection. This is done in Section 3.1. Further, in Section 3.2 and in Section 3.3 we outline key notions relevant to our analysis of strategic nomination of candidates. Namely, in Section 3.2 we define the *Hotelling-Downs* model, and in Section 3.3 we discuss *tournaments*, including the description of *knockout tournaments*. Then, in Section 3.4 we provide concepts related to *social networks*, including the *opinion diffusion protocol* which we use in our investigations. There, we also define *majority illusion*. Finally, in Section 3.5 we discuss aspects of *computational complexity*, which we use throughout this thesis. In particular, we define *complexity classes*, which we use to determine the difficulty of computational problems we consider. We also present problems which we use for our proofs by reduction. Finally, we formally define *Turing machines* and *Boolean circuits*. Table 3.1 outlines the relevance of sections of this chapter to particular parts of this thesis.

Section 3.1	Game Theory	Chapters 4,5,6
Section 3.2	Hotelling-Downs Games	Chapter 4
Section 3.3	Tournament Solutions	Chapters 5,6
Section 3.4	Social Networks	Chapter 7,8
Section 3.5	Computational Complexity	Chapters 4,5,6,7,8

Table 3.1: Overview of sections of this chapter and of their relevance to particular parts of the thesis.

3.1 Game Theory

Let us define basic notions of a *strategic game* and of the (pure) *Nash equilibrium* which will be crucial for our analysis in Chapters 4, 5 and 6. An extensive introduction to game theory can be found in, e.g. Maschler et al. [2020].

A strategic game is a tuple $(N, \mathcal{C}, u)$, where $N = \{1, \ldots, n\}$ is a set of *agents* and $\mathcal{C} = A_1 \times \cdots \times A_n$ is a set of *strategy profiles*, with A_i being a set of *actions* available to agent i. Further, $u = (u_1, \ldots, u_n)$ is a tuple of *utility functions*, where $u_i : \mathcal{C} \to \mathbb{R}$ determines the utility of an agent i given a strategy profile.

Furthermore, given a strategy profile $\mathbf{c}$ (which is a member of A), an agent i, and an action $c_i \in A_i$, we denote as $(c_i', \mathbf{c}_{-i})$ the strategy profile in which the action selected by i is c_i' and all other players select the same action as in $\mathbf{c}$. Then, we say that a strategy profile $\mathbf{c}$ is a *pure Nash equilibrium* (NE), if for every player i, and for every action $c_i' \in A_i$, it holds that $u_i(\mathbf{c}) \geqslant u_i(c_i', \mathbf{c}_{-i})$. So, a strategy profile is a NE if no agent can improve their utility by changing their strategy unilaterally.

Furthermore, given an agent $i \in N$, an action $a \in A_i$, and a strategy profile $\mathbf{c}$, we have that a is a *best response* to $\mathbf{c}_{-i}$ if and only if $a \in \arg\max_{a' \in A_i} u_i(a', \mathbf{c}_{-i})$. Notice that $\mathbf{c}$ is an NE if and only if for all $i \in N$, c_i is a best response to $\mathbf{c}$.

3.2 Hotelling-Downs Games

Let us define the key concepts which we use in our analysis of our coalitional extension of the Hotelling-Downs model.

Parties, Voters, and Games. The games we are concerned with are played on a discrete *line* $\{0, 1, 2, \ldots, k\}$. For $x, y \in \mathbb{R}_0$ with $x \leq y$, we denote by $[x, y]$ the *segment* $\{\lfloor x \rfloor, \lfloor x \rfloor + 1, \ldots, \lceil y \rceil\}$. On this line, a positive number of *voters* are placed according to a distribution function $f : [0, k] \to \mathbb{N}_0$. So, every voter is assigned a place corresponding to their views. Observe that since we require the line to be discrete, it follows that the number of voters in the game is finite. Further, we denote as $V(f) = \sum_{i \in [0,k]} f(i)$ the total number of voters on a line $[0, k]$. We will also assume throughout that $V(f) > 0$. At times we will restrict attention to distribution functions f that are *uniform*, i.e., such that exactly one voter is assigned to each point of the line. So, in a uniform distribution, $f(i) = 1$ for each $i \in [0, k]$.

The players of the game are given by a set $P = \{P_1, P_2, \ldots, P_n\}$ of *parties*. Each party P_i is fully described by a set of points on the line, i.e., $P_i \subseteq [0, k]$. We also assume that parties are not empty. Intuitively, these points correspond to the positions of the candidates from which a party has to choose its nominee. Formally, they make up the party's strategies. We allow different parties to have candidates that occupy the same position, i.e., P_i and P_j need not be disjoint for distinct i and j.

Parties strategise over which candidate to select as their *nominee*. Thus, in this context, a strategy profile is a tuple $\mathbf{c} = (c_1, c_2, \ldots, c_n)$, where $c_i \in P_i$. With a slight abuse of notation, we will sometimes consider strategy profiles to be sets. For every party P_i and every strategy profile $\mathbf{c} = (c_1, \ldots, c_n)$, we associate a utility for P_i. Intuitively, it is the number of voters that are closer to c_i than every other party P_j's chosen nominee c_j, and who are therefore willing to vote for c_i. A voter that is just as far removed from nominees on the left as from nominees to the right will contribute a half to the utility of the former and half to the utility of the latter. Further, we assume that nominees that are in the same position share the number of voters they attract evenly. The following example illustrates the setup and introduces the intuition behind the concept of utility in the current context.

Example 3.1. *Figure 3.1 depicts the line* $[0, 5]$ *and two parties,* $P_1 = \{4\}$ *and* $P_2 = \{0, 2\}$*. The distribution* f *of the voters, indicated at the top of the line, is given by* $f(0) = f(1) = f(3) = f(5) = 0$*,* $f(2) = 4$*, and* $f(4) = 5$*. Below the line the figure displays the strategy profile* $\mathbf{c} = (c_1, c_2)$*, where* $c_1 = 4$ *and* $c_2 = 2$*. Given this strategy profile,* c_1 *attracts* 5 *voters, while* c_2 *attracts* 4 *voters.*

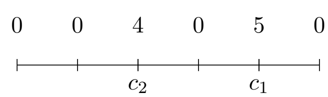

Figure 3.1: Game in Example 3.1. Numbers above the line indicate the number of voters at each position, while below the line we specify the positions of nominees of P_1 and P_2.

Let us define the utility function in the current context formally. Let P_i be a party and $\mathbf{c} = (c_1, \ldots, c_n)$ be a strategy profile. Then, we denote the highest player in $\mathbf{c}$ which is strictly smaller than c_i, whenever one exists, i.e., the immediate predecessor of c_i, as $L(c_i)$. Likewise, we use $R(c_i)$ for c_i's immediate successor in $\mathbf{c}$ on the line, whenever one exists. That is, $L(c_i) = \sup\{c_j \in \mathbf{c} \colon c_j < c_i\}$ and $R(c_i) = \inf\{c_j \in \mathbf{c} \colon c_j > c_i\}$, assuming that $\sup \emptyset = -\infty$ and that $\inf \emptyset = \infty$. To capture whether a voter is willing to support a given nominee, we associate with each candidate $c_i \in \mathbf{c}$ an indicator function $\sigma_i^{\mathbf{c}} \colon [0, k] \to \{0, \frac{1}{2}, 1\}$. This function assumes value 1 at m if $m \in [0, k]$ is strictly closer to c_i than to every other $c_j \in \mathbf{c}$.

Further, it has the value $\frac{1}{2}$ at m if m is equally close to c_i as some other $c_j \in \mathbf{c}$, with $c_i \neq c_j$, and no nominee is strictly closer to m than c_i. Finally, it has value 0 at m if m is strictly closer to some $c_j \in \mathbf{c}$ other than c_i. Hence, we have that

$$\sigma_i^{\mathbf{c}}(m) = \begin{cases} 1 & \text{if } |c_i - m| < \min\{|L(c_i) - m|, |R(c_i) - m|\}, \\ \frac{1}{2} & \text{if } |c_i - m| = \min\{|L(c_i) - m|, |R(c_i) - m|\}, \\ 0 & \text{Otherwise} \end{cases}$$

We note that associating the function we consider with a candidate, rather than just with a position on the line, will be helpful towards capturing their utilities. If $\mathbf{c}$ is clear from the context, we sometimes write simply σ_i. Further, the *range* of candidate c_i on line $[0, k]$ which we denote as $\mathit{range}_{c_i}(\mathbf{c})$, is the set $\{m \in [0, k] \colon \sigma_i^{\mathbf{c}}(m) > 0\}$. It is worth observing that the range of a candidate is an interval. We also denote as $\#S$ the cardinality of a set S. Moreover, $[c_i]$ is the set of candidates in $\mathbf{c}$ sharing the position c_i. I.e., $[c_i] = \{1 \leq j \leq n \colon c_i = c_j\}$. We are now ready to define the *utility* $u_i(\mathbf{c})$ of party P_i for a profile $\mathbf{c}$.

$$u_i(\mathbf{c}) = \frac{\sum_{m \in [0,k]} \sigma_i^{\mathbf{c}}(m) \cdot f(m)}{\#[c_i]}$$

Observe that in the current setting each voter is either attracted to one candidate, or contributes to the utility of multiple parties with the sum of its contributions being equal to 1. Hence, we get that for every strategy profile $\mathbf{c}$ it holds that $\sum\limits_{P_i \in P} u_i(\mathbf{c}) = V(f)$.

Input representation. As we are concerned with the computational complexity of decision problems within this framework, it is useful to clarify the input representation. In particular, it is important to notice that we can represent a game, with voters situated on the line $[0, k]$, a distribution given by a function $f : [0, k] \to \mathbb{N}_0$, and a set of parties $P = \{P_1, \ldots, P_n\}$ as a $(n+1) \times (k+1)$ table, where an entry $(1, i)$ specifies the number of voters at position $i - 1$ and for $j > 1$, entry (j, i) specifies if party P_{j-1} has a potential candidate at position $i - 1$. So, the representation of the game has size bounded by $(k + 1) \cdot n + (k + 1) \cdot \log \max_f$ bits, where $\max_f$ denotes the maximum number of voters at any point on line $[0, k]$.

3.3 Tournament Games Played by Coalitions

Let us provide the definitions of the concepts needed for our analysis of tournaments played by coalitions. An extensive overview of tournaments can be found in Brandt et al. [2016b], and in Williams and Moulin [2016].

Tournaments. A *tournament* is a directed graph (N, E), where N is the set of *players*, while E, which we also call a *beating relation*, is an irreflexive, and asymmetric relation over N. We further assume that for every pair of players $i \neq j$ in N, exactly one of (i, j) and (j, i) belongs to E. Further, as we assume that E is irreflexive, for every $i \in N$ we have that $(i, i) \notin E$. By this we capture that no player beats itself. If it holds that $(i, j) \in E$, then we say that i *beats* j in E. We will also say that i beats j, omitting E, when E is clear from the context. Given a tournament (N, E) and $i \in N$, we denote as $B(i)$ the set of players beaten by i. Formally, $B(i) = \{j \in N : i \text{ beats } j\}$. Also, for a graph $G = (N, E)$, we say that G is *k-partite*, if there is a partition $N_1, \ldots, N_k$ of N, such that for every N_i and every pair $j, j' \in N_i$, it holds that $(j, j') \notin E$.

Coalitions. We study the case in which players are partitioned into *coalitions*. For a given tournament (N, E), a coalition is a member of a partition $C = \{C_1, \ldots, C_m\}$ of N. We call such a partition a *set of coalitions*, and denote a tuple $T = (N, E, C)$ as a *coalitional structure*. So, for every player $i \in N$, we have that i is a member of exactly one $C_i \in C$. Further, in Chapter 6, we model the scenario in which all coalitions choose one of their players to participate in the competition. So, we assume that each of them selects exactly one player. Hence, a *strategy profile* in this context is a tuple $(c_1, \ldots, c_m)$, such that for every $i \leqslant m$ we have that $c_i \in C_i$. For convenience, for a coalitional structure (N, E, C) and $j \in N$, we denote as $\mathcal{C}(j)$ the coalition $C_i \in C$, such that $j \in C_i$. Further, given a coalitional structure (N, E, C), a pair of coalitions $C_i, C_j \in C$, and a player $c_i \in C_i$, we say that c_i *dominates* C_j if for every $c_j \in C_j$ we have that c_i beats c_j. We also say that C_i dominates C_j, if for every $c_i \in C_i$ we have that c_i dominates C_j.

To account for the competition between the selected players, given a coalitional structure $T = (N, E, C)$ and a strategy profile $\mathbf{c}$=$(c_1, \ldots, c_n)$, a *filtration of* T induced by $\mathbf{c}$, which we denote as $T_{\mathbf{c}}$, is the tournament $(\mathbf{c}, E')$, where for every pair of players $c_i, c_j \in \mathbf{c}$ it holds that c_i beats c_j in E' if and only if c_i beats c_j in E. In other words, $T_{\mathbf{c}}$ is a restriction of T to players in $\mathbf{c}$.

Example 3.2. *Let us illustrate the notions defined above. Consider coalitions* $A =$

$\{a_1, a_2\}$, $B = \{b_1, b_2\}$, $C = \{c_1, c_2\}$, *and* $D = \{d_1, d_2\}$. *Then, assume that* a_1 *beats all other players. Further, all members of* B *and* C *are beaten by all players in groups* A *and* D*, as depicted in the left side of Figure 3.2. The remaining edges in the tournament are chosen arbitrarily. Then, coalitions select players* a_1, b_2, c_2*, and* d_1*, i.e., the strategy profile* (a_1, b_2, c_2, d_1).

The left subfigure of Figure 3.2 presents the tournament based on the results of the beating relation, while the right depicts a filtration based on the strategy profile (a_1, b_2, c_2, d_1).

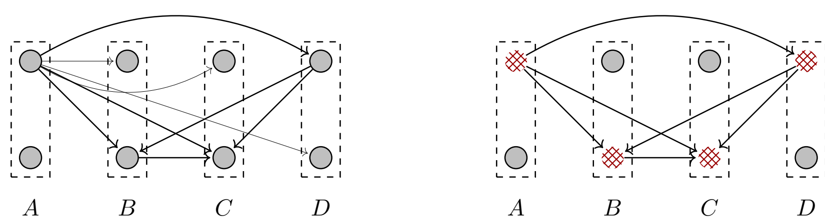

Figure 3.2: In the left subfigure, the coalitional structure with coalitions A, B, C, D, each with two members. In the right subfigure, filtration based on a strategy profile. Vertices in red represent the chosen players. For clarity, only selected edges in the tournament are presented.

Tournament Solutions. In the scenario we consider, coalitions are interested in optimizing their performance in a competition. To determine the set of winners of a tournament, in Chapter 6, we consider *tournament solutions*, i.e., rules for selecting a set of winners from a tournament. Let $\mathbf{T}_N$ be the set of all tournaments with the set of players N. Then, a tournament solution is a function $F : \mathbf{T}_N \to 2^N$. Further, given a coalitional structure $T = (N, E, C)$, a strategy profile $\mathbf{c}$ and a tournament solution F, we say that a coalition C_i is a *winner* of $T_\mathbf{c}$ under F if $c_i \in F(T_\mathbf{c})$. When clear from the context, we just say that C_i is a winner of $T_\mathbf{c}$, or that C_i is a winner under $\mathbf{c}$ (when the coalitional structure is clear from the context). We also say that $c_i \in C_i$ is a winner in the *initial tournament* if $c_i \in F(T)$, i.e., if c_i is a winner in the tournament defined on the set of all players in N.

Nash Equilibrium. We are interested in the game-theoretic study of the scenario we consider, focusing on NE. In the scenario studied in Chapter 6, in a strategy profile $\mathbf{c}$, which is a NE, no coalition would have their unilaterally changed nominee as a winner, if they did not in $\mathbf{c}$. So, for a coalitional structure $T = (N, E, C)$ and a tournament solution F, we say that $\mathbf{c}$ is a NE under F, if, for every coalition C_i, such that C_i is not a winner of $T_\mathbf{c}$, and for every $c'_i \in C_i$, it holds that C_i is not a winner of $T_{(c'_i, \mathbf{c}_{-i})}$ under F.

Knockout Tournaments. In Chapter 5, we will further consider tournaments in which coalitions compete in rounds. For a tournament $T = (N, E)$, a *knockout tournament*, or a *single-elimination tournament* on T is defined as a complete binary tree B with $\#N$ *leaves*. We will also refer to such structures as to *SE-tournaments*. We denote the set of leaves of a SE-tournament as $L(B)$. Further, we consider a bijective function $\pi : N \to L(B)$, which we call the *seeding*, which maps the $\#N$ players to the $\#N$ leaves of a SE-tournament. Then, the *winner of the knockout tournament*, corresponding to π, is determined recursively. First, the winner at a leaf l is the player j, with $l = \pi(j)$. Then, the winner of the subtree rooted at a vertex v, i.e., of a tree contained in T, is the winner of the match between the winners of the two subtournaments rooted at the children of v. There, the winner is decided by the orientation of the unique edge between these two players, say, i and j. So, i is the winner if and only if $(i, j) \in E$. We will only consider SE-tournaments based on perfect binary trees, implying that the number of players entering a knockout tournament is a power of 2.

We assume that the seeding is fixed, and known to all coalitions a priori. Hence, when possible, in order to keep the notation simple, we will refrain from explicitly referring to the seeding.

An example of a SE-tournament is depicted in Figure 3.3.

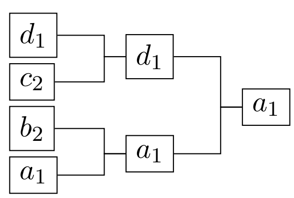

Figure 3.3: A SE-tournament with four players, a_1, b_2, c_2, d_1, with d_1 beating c_2, a_1 beating b_2, and a_1 beating d_1. There, a_1 and d_1 are winners in the first round of the tournament, since a_1 beats b_2, while c_2 beats d_1. In consequence, a_1 is the winner of the tournament.

Subtournaments. Consider now a set of players $[1, 2^m]$, for some natural number m. We denote by $SE_{\pi,[1,2^m]}$ the spanning binomial arborescence representing the SE-tournament played by the players in $[1, 2^m]$, following the seeding π. We call the root of $SE_{\pi,[1,2^m]}$ the *winner* of $SE_{\pi,[1,2^m]}$. Moreover, for each $r \in [1, m]$, we denote by $SE^r_{\pi,[1,2^m]}$ the binomial arborescence representing the subtournament of $SE_{\pi,[1,2^m]}$, played by the winners of all the r^{th} round matches of $SE_{\pi,[1,2^m]}$. Notice that there are exactly 2^{m-r} players who win at least r rounds. In other words, $SE^0_{\pi,[1,2^m]}$ is the same as the full tournament $SE_{\pi,[0,2^m]}$ and, for every $r \in [1, m]$, $SE^r_{\pi,[1,2^m]}$ denotes the subgraph of $SE^{r-1}_{\pi,[1,2^m]}$, obtained by deleting all of its leaves. Notice that $SE^m_{\pi,[1,2^m]}$ contains a single vertex, which corresponds to the winner of $SE_{\pi,[1,2^m]}$. Figure 3.4 depicts subtournaments for particular rounds of the SE-tournament shown in Figure 3.3.

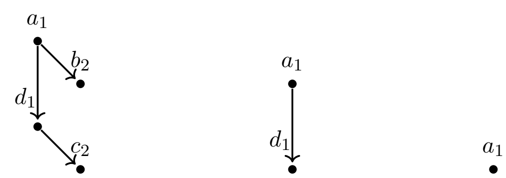

Figure 3.4: Subtournaments of the SE-tournament in Figure 3.3. The leftmost subfigure shows $SE^0_{\pi,\{a_1,b_2,c_2,d_1\}}$, the middle one, $SE^1_{\pi,\{a_1,b_2,c_2,d_1\}}$, and the rightmost, $SE^2_{\pi,\{a_1,b_2,c_2,d_1\}}$.

Input Representation. Let us establish the input representation and input size of the problems concerning tournament played by coalitions which we consider. Observe first that we can represent any graph (N, E), with $N = \{p_1, \ldots, p_n\}$, as a $n \times n$ table, where every entry (i, j) specifies if p_i beats p_j, or if p_j beats p_i. Similarly, we can represent a partition $C_1, \ldots, C_l$ of N as a $n \times l$ table, where each entry (i, j) specifies if p_i belongs to C_j. Given these observations, we can represent a coalitional structure (N, E, C) in space of size $\mathcal{O}(\#N^2 \log \#N) + \mathcal{O}(\#N \cdot \#C)$. Here, $\mathcal{O}(\#N^2 \log \#N)$ is a bound on the representation of (N, E), and $\mathcal{O}(\#N \cdot \#C)$ is the space to represent the partition of players into coalitions.

Furthermore, we can represent a seeding of a knockout tournament, assuming that l is a power of 2, as a l long list, where each entry j specifies which C_i is seeded at position j. Hence, $\mathcal{O}(\#C \log \#C)$ is the size of space needed to represent a seeding. As the number of coalitions $\#C$ is at most the number of players $\#N$, since we do not permit empty coalitions, the input has the overall bit-size $\mathcal{O}(\#N^2 \log \#N)$.

3.4 Social Networks

Let us define the concepts needed for our analysis of *social networks*.

Social Networks. Let N be a finite set of *agents* (also referred to as *vertices*), and E be a directed graph over N, i.e., a relation over the set of agents. We assume that E is irreflexive, by which we mean that for every $i \in N$, it holds that $(i, i) \notin E$.

In other words, an agent does not take their own opinion into account. We further call a tuple (N, E) a *social network.* The intuition behind this definition is that each agent is influenced by the incoming edges, and influences the outgoing ones. So, we interpret the fact that $(i, j) \in E$ as "i influences j". An example of a social network is shown in Figure 3.5.

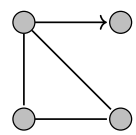

Figure 3.5: Example of a social network with four vertices. There, three of them are all connected to each other, while one vertex has only one influencer. A connection between a pair vertices which does not indicate its direction depicts the symmetric relation between them.

Then, for each $i \in N$, we define the set of agents influenced by i as $N(i) = \{j : (i, j) \in E\}$. We also call agents in $N(i)$ the *neighbours* of i. Similarly, we define the set $N^{-1}(i) = \{j : (j, i) \in E\}$ of the *influencers of* i. Observe that as we assume that social networks are irreflexive, it holds that $i \notin N^{-1}(i)$. Further, if $N(i) = \emptyset$, then we say that i is a *sink.* Also, if $N^{-1}(i) = \emptyset$, then we say that i is a *source.* Moreover, we say that a network (N, E) is an *extension* of (N, E'), if $E' \subseteq E$. Similarly, if $E \subseteq E'$, we say that (N, E) is a *subnetwork* of (N, E'). Also, we say that a set of agents $S \subseteq N$ is a *clique*, if for every $i, j \in S$, such that $i \neq j$, it holds that $(i, j) \in E$, following the assumption that social networks are irreflexive. Further, we say that S is an *independent set*, if for every $i, j \in S$, we have that $(i, j) \notin E$.

In our analysis we will study two natural classes of networks, i.e., *bipartite*, and *planar* networks. We say that a network is bipartite if it can be divided into two independent sets. This notion is important from the perspective of social network analysis. In particular, *affiliation networks* (see, e.g., Lattanzi and Sivakumar [2009]), in which each member of a group of, e.g., students, is assigned to a member of another group of, e.g., professors. We further call a bipartite network (N, E), which is divided into two independent sets A and B, such that for every $i \in A$ and $j \in B$, it holds that $(i, j), (j, i) \in E$, a *complete bipartite network.* We also denote such a network as $K_{\#A,\#B}$. Figure 3.6 depicts an example of a complete bipartite network, $K_{4,3}$.

We further say that a network (N, E) is *planar*, if it can be drawn on a plane, so that the edges in E do not cross. These networks are well-represented

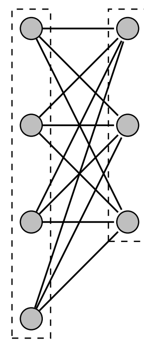

Figure 3.6: Example of a complete bipartite network with seven vertices ($K_{4,3}$).

in practice, for instance in the representation of motorways (see, e.g., Viana et al. [2013]). Further, we will make use of the fact that many natural networks, such as those containing a clique with more than four vertices, are not planar, in our computational complexity results. An example of a planar network is shown in Figure 3.7.

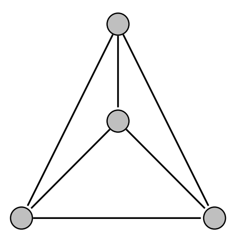

Figure 3.7: Example of a planar clique with four vertices.

We will also consider *strongly connected components* (*SCC*s) of networks, as well as *directed acyclic graphs (DAGs)*. For a network (N, E) and a sequence $S = (i_0, \ldots, i_n)$, such that each i_j in S belongs to N, we say that S is a *path* if for each $k < n$ it holds that $(i_k, i_{k+1}) \in E$. If there is a path from a vertex i to a vertex j, then we write that $i \rightarrow j$. We further say that (N, E) is a DAG, or that it is *acyclic*, if no agent i is reachable from itself, i.e., there is no path in E, which starts and ends at i. Observe that this implies that there are no infinite paths in (N, E).

Then, a network (N', E'), with the set of agents $N' \subseteq N$ and $E' \subseteq N'$ is a *strongly connected component (SCC)*, if for every pair $i, j \in N'$, such that $i \neq j$, there exist a a path from i to j in (N', E'), while N' is a maximal such set with respect to set inclusion.

As is well-known (see, e.g., Bollobás [1998]), each network $SN = (N, E)$ can be partitioned into SCCs, yielding a DAG $SCC_{SN} = (SCCs, E')$, where:

1. *SCCs* is the set of all SCCs of *SN*.

2. For every $SCC_u, SCC_v \in SCCs$, we have that $(SCC_u, SCC_v) \in E'$ if and only if for some $i \in SCC_u$, $j \in SCC_v$, we have that $j \in N(i)$.

An example of a partition of a network into SCCs is depicted in Figure 3.8.

Example 3.3. *Figure 3.8 presents the connections between agents. Further, each set of vertices in a dashed rectangle is a SCC of this network. Observe how these components form a DAG, in which the clique consisting of four vertices is a source, and the other components are sinks.*

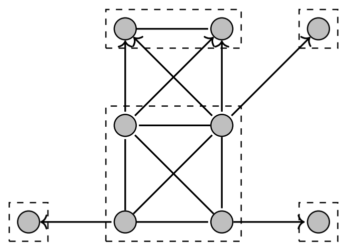

Figure 3.8: Example of a social network and its partition into SCCs.

We are interested in the opinions adopted by agents in social networks and, in particular, how they spread following the influence relation. For this we equip each agent with a single opinion. We call a social network, where every agent is assigned a view (or a *colour*) a *labelled social network*. Throughout the thesis we assume the binary set of opinions $\{b, r\}$, i.e, *blue* and *red*.

Definition 3.1 (Labelled Social Network)**.** *A* labelled social network *is a tuple* $SN = (N, E, f)$, *where:*

- (N, E) *is a social network,*
- $f : N \to \{b, r\}$ *is a* labelling *of each vertex.*

Given a labelling f of a social network (N, E), we denote the set of vertices labelled b, i.e., $\{i \in N : f(i) = b\}$, as B_f, and the set of vertices labelled r, i.e., $\{i \in N : f(i) = r\}$, as R_f. Moreover, for a set $S \subseteq N$, we say that B_f^S is the set of vertices labelled red in S, while R_f^S is the set of vertices labelled r in S. We omit f, if it is clear from the context. Also, the *blue surplus* of a vertex is the number of its blue neighbours minus the number of its red neighbours. For a vertex set X and a labelling $f : X \to \{b, r\}$, we define the *red neighbourhood of a vertex i under f* as the set of neighbours of i in X that are assigned the label r by f, and this set is denoted by $N_{f,r}^X(i)$. We drop the explicit reference to X or f in this notation, if it is clear from the context. The variation of this definition for blue neighbourhood is analogous. Examples of labelled social networks are provided later, in Example 3.4.

Opinion Change. We model opinion change as an update protocol on the network, where each agent i takes the opinion of their influencers, i.e., $E^{-1}(i)$, into account. We note that the protocol of opinion change, based on a fraction of infuencers disagreeing with an agent, is well studied in the literature (see, e.g., Granovetter [1978]).

For a given labelled social network (N, E, f) and an agent i, let us call $A(i) = \{j \in N^{-1}(i) : f(i) = f(j)\}$ the set of influencers, who agree with i's opinion. Further, let $D(i) = N^{-1}(i) \setminus A(i)$ be the set of i's influencers, who disagree with i. We further assume that agents change their opinion if the fraction of their influencers disagreeing with them is strictly higher than a half.

Definition 3.2 (Opinion Change). *Let* $SN = (N, E, f)$ *be a labelled social network, and let* $i \in N$ *be an agent. Then, the* opinion diffusion step *is the function* $OD : N \to \{b, r\}$, *such that:*

$$OD(SN, i) = \begin{cases} flip(f(i)) & \text{if } \#D(i) > \#A(i) \\ f(i) & \text{otherwise} \end{cases}$$

where $flip(k) = \{b, r\} \setminus f(k)$ *denotes the change from an original opinion to its opposite value.*

In particular, a vertex with $2k$ influencers always takes the opinion of the strict majority of the set $N^{-1}(i) \cup \{i\}$. To see that, consider a vertex i, with $\#N^{-1}(i) = 2k$, and observe that $OD(SN, i) \neq f(i)$ if and only at least $k+1$ of i's neighbours have a different opinion than i. But this implies that $OD(SN, i)$ is the colour of the strict majority in $E^{-1}(1) \cup \{i\}$, as the cardinality of this set is $2k+1$.

We are now ready to define the protocol for the opinion change in a labelled social network. Here we focus on the *synchronous update*, in which all agents modify their opinions at the same time.

Definition 3.3 (Synchronous Update). *Let* $SN = (N, E, f)$ *be a labelled social network. Then,* $SU(SN) = (N, E, f')$, *where for every* $i \in N$ *we have that* $f'(i) = OD(SN, i)$.

It is important to notice that the synchronous update protocol defined above is deterministic. So, given a labelled social network, we can compute its unique labeling after any given number of synchronous updates. Further, we denote as the *update sequence* of a labelled social network SN the infinite sequence of states of SN, after successive synchronous updates.

Definition 3.4 (Update Sequence)**.** *Given a labelled social network $SN = (N, E, f)$, the update sequence generated by SN is the sequence of labelled social networks $SN_{us} = (SN_0, SN_1 \dots)$, such that $SN_0 = SN$, and for every $n \in \mathbb{N}$, we have that $SN_{n+1} = SU(SN_n)$.*

We call a labelled social network *SN stable*, if $SU(SN) = SN$. So, in a stable social network, for every agent i, it holds that at least a half of i's neighbours is labelled with the same colour as i. Further, a labelled social network is *convergent*, if its update sequence contains a stable social network, i.e., if it reaches a fixed point, which we also call its *limit network*. For convenience, we also say that a network *stabilises*, or *converges*, after k steps, if the k^{th} element of its update sequence is stable. Similarly, we say that an agent stabilises after k steps, if it is labelled with the same colour in every $SN_{k'}$, with $k' \geqslant k$.

Example 3.4. *Let us illustrate the notions of a labelled social network and of opinion diffusion. Subfigures of Figure 3.9 present two different labellings of the same social network. Observe that in both of them blue vertices constitute the majority. However, in the right subfigure (which we denote as SN), after one step of opinion diffusion, all vertices are labelled red, as each vertex has the strict majority of its neighbours labelled r. In other words, in the update sequence $SN_{us} = (SN_0, SN_1 \dots)$, every network SN_i, with $i > 0$, has all of the vertices labelled r. On the contrary, in the left subfigure (denoted as SN'), the opinion diffusion protocol will never terminate, as the members of the central clique will switch their opinion in every opinion diffusion step. In other words, the update sequence SN'$_{us}$ does not have a fixed point.*

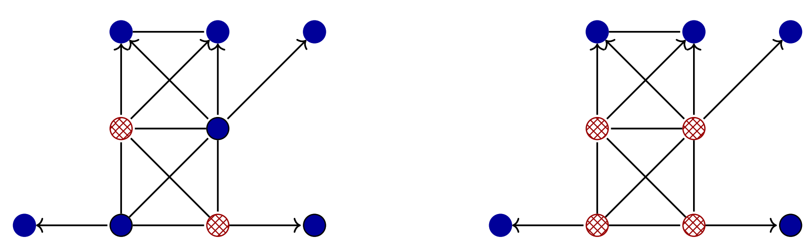

Figure 3.9: In the left subfigure, a labelling of a network which never converges. In the right, a labelling for which the protocol converges after one opinion diffusion step.

Majority Illusion. Here, we provide basic definitions, which we will use in our analysis of majority illusion. There, we assume that, for each social network (N, E) we consider, E is symmetric. I.e., we assume that social networks are *undirected*. Note that in this case, for every agent i, we have that $N^{-1}(i) = N(i)$.

Then, we say that a colour $k \in \{b, r\}$ is a *strict majority winner* in a labelled social network (N, E, f), if there are strictly more agents coloured with k, than with $\{b, r\} \setminus \{k\}$. Further, we use $W_{(N,E,f)}$ to denote such a winner, whenever it exists. Similarly, for an agent $i \in N$, we say that a colour $k \in \{b, r\}$ is a *strict majority winner in i's neighbourhood*, if the strict majority of i's neighbours is labelled k. We further use $W^i_{(N,E,f)}$ to denote such a winner, whenever it exists.

In order to define majority illusion, we say that an agent $i \in N$ is *under illusion* if they have a wrong perception of the majority winner. In other words, for an agent i to be under illusion in a social network (N, E) with labelling f, we must have that $W_{(N,E,f)}$ and $W^i_{(N,E,f)}$ exist, while $W^i_{(N,E,f)} \neq W_{(N,E,f)}$. In our analysis of majority illusion we are concerned with the *proportion* of agents in a network, which are under illusion. To account for that, we define the concept of *q-majority illusion*, by which we mean that at least the fraction q of agents is under illusion.

Definition 3.5 (q-majority illusion)**.** *For a given social network* (N, E)*, fraction* q*, and labelling* $f : N \to \{b, r\}$*, we say that* f induces *q-majority illusion, if at least* $q{\cdot}\#N$ *agents are under illusion in* (N, E, f)*.*

If there exists a labelling of a network (N, E), which induces q-majority illusion, then we say that (N, E) *admits* q-majority illusion. In Chapter 8 we further assume that the strict majority colour is blue, whenever one exists. Also, for a network (N, E) and agents $i, i' \in N$, such that $N(i) = \{i'\}$, we say that i is a *dependent* of i'. Then, we also say that i' has a dependent. Let us observe now that if a labelling f induces 1-majority illusion for a network (N, E), and i is a dependent of i', then $f(i') = r$. Finally, for a labelled network (N, E, f) and an agent $i \in N$, we define the *margin of victory* for i as $\#N_{f,b}(i) - \#N_{f,r}(i)$. We also define the margin of victory, for a labelled social network, in a natural way. Further, for a pair of labelled social networks (N, E, f), (N, E, f'), and $i \in N$, we say that i is *pushed into illusion* in (N, E', f), if i is under illusion in (N, E, f), but not in (N, E, f). Symmetrically, we say that illusion is *eliminated* from i in (N, E', f), if i is under illusion in (N, E, f), but not in (N, E', f).

Graph Parameters. In our analysis of computational properties of majority illusion we will use a number of graph properties. In particular, we consider the *neighbourhood diversity* (see Lampis [2012]), which captures the number of "twin classes" in the graph. We say vertices u and v are *twins*, if they have the same neighbours, i.e., $N(u) \setminus \{v\} = N(v) \setminus \{u\}$. Given a graph G, let further $V(G)$ denote the set of vertices in G.

Definition 3.6. *The* neighbourhood diversity *(ND) of a graph G, which we also denote as $nd(G)$, is the minimum number w, such that $V(G)$ can be partitioned into w sets of twin vertices. Each set of twins, called a* module*, is either a clique, or an independent set. We call these sets* clique modules*, and* independent modules*, respectively.*

We will further consider a property of a social networks that has gained importance in recent years, i.e., the *c-closure* (see, e.g., Fox et al. [2018, 2020]; Koana et al. [2020]). For a natural number c, we say that a network is *c-closed*, if every pair of vertices in this network, which have at least c neighbours in common, is adjacent. This concept was introduced in an attempt to capture the spirit of "social-network-like" graphs, without relying on probabilistic models. Note that c-closure generalises one of the most agreed-upon properties of social networks, i.e., *triadic closure*. This property is that when two members of a social network have a friend in common, they are are friends themselves. It is important to note that Fox et al. [Fox et al., 2020, Table A.1], and later Koana et al. [Koana et al., 2020, Table 1], showed that several social networks and biological networks are indeed c-closed for rather small values of c.

Input Representation. Let us establish the input size for the key notions regarding social networks, which we defined above. As argued in the case of tournaments, a network (N, E), with $N = \{v_1, \ldots, v_n\}$, can be represented as a $n \times n$ table, where every entry (i, j) specifies if (i, j), (j, i), or $\{(i, j), (j, i)\}$ belongs to E. Hence, a network can be represented in size $\mathcal{O}(\#N^2)$. Furthermore, a labelling $f : N \to \{b, r\}$ can be represented as a vector of length n, where every position i indicates if v_i is coloured blue or red. So, f can be represented in size $\mathcal{O}(n)$.

3.5 Computational Complexity

Here, we define key concepts related to *computational complexity*, which we will use in the analysis of the problems we consider in this thesis. We will study *computational problems*, in which we are interested in determining whether an answer is positive, or negative, for a given instance. We define computational problems in terms of sets of *binary strings*, i.e., strings composed of 0 and 1, which can be used to encode complex objects. So, we denote the set of strings of length n, with $n \in \mathbb{N}$, as $\{0, 1\}^n$. Then, for a function f from the set of binary strings to $\{0, 1\}$, the *language* of f is the set $L_f = \{x : f(x) = 1\}$. Then, a computational problem of deciding f is the problem of whether a binary string x (which we refer to as an *input*) belongs to L_f.

We will further examine the *time* and *space* complexity of algorithms for such problems. Intuitively, the time complexity is a measurement of a number of steps needed to compute the answer to a problem by a specific algorithm, whereas the space complexity indicates how many bits of space are needed for such a computation. To account for the time measurement, let us introduce big $\mathcal{O}$ and big Ω notation. Intuitively, big $\mathcal{O}$ indicates the *upper-bound* of a function. Formally, for a pair of functions $f, g : \mathbb{N} \to \mathbb{N}$, we say that $f \in \mathcal{O}(g)$, if for some constants $n_0 \in \mathbb{N}$ and $c > 0$, it holds that, for every $n \geqslant n_0$, we have that $f(n) \leqslant g(n) \cdot c$. We note that every polynomial function $f(n)$ is in $n^{\mathcal{O}(1)}$. Symmetrically, big Ω notation indicates the *lower-bound* of a function. Formally, we say that $f(n) \in \Omega(g)$, if there are constant numbers n_0, c, such that for every $n \geqslant n_0$, we have that $f(n) \geqslant g(n) \cdot c$.

Complexity Classes. When studying the computational hardness of the problems we consider, we will be referring to the notion of *complexity classes*, i.e., collections of problems, which are defined in terms of their computational complexity. Here, we define the classes which we will be mentioning in the thesis.

It is customary to assume that a problem is tractable, if it can be solved in time bounded by some polynomial of input size. So, we say that a problem R, with an input of size n, is solvable in *polynomial time* (or that it is in **P**), if there exists an algorithm deciding R, which is computed in $n^{\mathcal{O}(1)}$ steps.

Furthermore, we say that a problem R with an input n is solvable in *linear time* (or that it is in **L**), if there exists an algorithm deciding R, which is computed in $\mathcal{O}(n)$ steps.

Further, we say, intuitively, that a problem R is in **NP**, if an answer to R can be verified in polynomial time. The formal definition of this notion is given in a later part of this chapter, in Definition 3.8. Note that it follows directly by definition that all problems in **P** are also in **NP**. It is also believed that **NP** is larger than **P**.

Another interesting class which we will study contains problems which might not be solvable in polynomial time, although there are algorithms deciding them, whose running time is bounded by a function growing slower than the exponential function. We say that a problem P, with an input n, is solvable in *quasi-polynomial* time (or that it is in **QP**), if there exists an algorithm deciding R, which is computed in $2^{\mathcal{O}(\log^c n)}$, for some fixed constant c. It is believed that there are no quasi-polynomial algorithms, which decide an **NP**-complete problem (see, e.g., Impagliazzo and Paturi [2001]).

Finally, we say that a problem is in **PSPACE**, if it is computable while using no more than $n^{\mathcal{O}(1)}$ bits of space. It is important to note that **PSPACE** contains all of the problems in **P** or **NP**.

Crucially, we get the following relations between the discussed classes (see Theorem 7.4 in Papadimitriou [1994]).

$$\mathbf{L} \subseteq \mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{PSPACE}$$

Reductions. To account for the comparison of computational complexity of problems, let us define the notion of a *many-one reduction*. For a problem R and a problem R', we say that R is *reducible* to R', if there is a function f, such that, for each input n of R, the answer for $f(n)$ to R' is positive if and only if the answer to R is positive for n. Then, intuitively, we have that R' is at least as hard as R.

Crucially, we say that a problem R is **NP**-hard, if every problem in **NP** is reducible to R, using a function computable in polynomial time. Observe that it follows immediately that if R is an **NP**-hard problem, and R is reducible to R', then R' is also **NP**-hard. Then, we say that a problem R is **NP**-*complete*, if it is in **NP** and is **NP**-hard. The definition of **PSPACE**-hardness and **PSPACE**-completeness is analogous.

Turing Machines. One of the fundamental models of computation, which we will use in our proofs, is of a *Turing machine*. Let us provide the formal definition of a Turing machine, following Definition 2.1 in Papadimitriou [1994].

Definition 3.7 (Turing Machine)**.** *A* Turing machine *is a quadruple* $M = (K, \Sigma, \delta, s)$*. Here,* K *is a finite set of* states*, and* $s \in K$ *is the* initial state*.* Σ *is a finite set of* symbols *(we say that* Σ *is the* alphabet *of* M*). We assume that* K *and* Σ *are disjoint sets.* Σ *always contains the special symbols* $\sqcup$ *and* $\triangleright$*, i.e., the* blank *and the* first *symbol. Finally,* δ *is a* transition function*, which maps* $K \times \Sigma$ *to* $(K \cup \{h, \text{"yes"}, \text{"no"}\}) \times \Sigma \times \{\leftarrow, \rightarrow, -\}$*. We assume that* h *(the* halting state*), "yes", "no" and the coursor directions,* $\leftarrow$ *for "left",* $\rightarrow$ *for "right", and* $-$ *for "stay", are not in* $K \cup \Sigma$*.*

Then, for a binary string x, which we call an *input* a Turing machine M, is initiated on a finite string of symbols, which starts with $\triangleright$, and followed by $x \in (\Sigma \cup \{\sqcup\})^*$. We call this string an *input* of M. The function δ indicates further the change that M makes, when encountering a given symbol in some state. So, $\delta(q, \sigma)$ specifies, intuitively, the next state, a symbol by which σ is replaced, and the next symbol on the tape, which should be read. Here, the symbol $\triangleright$ corresponds to the initial point in which coursor is located. Furthermore, we assume that if it gets to the position outside of the input, it reads the symbol $\sqcup$.

To account for the computation of M, let us introduce a notion of a *configuration*, by which we denote a tuple (q, u, v), where $q \in K$ is a state, while u and v are strings in Σ^*. Intuitively, q corresponds to a state in which a machine is. Also, u, v correspond to the symbols on the tape, on the left and on right side of the coursor, respectively. Let now σ denote the last symbol in u, and let $\delta(q, \sigma) = (p, \rho, D)$. Further, we say that a configuration $c = (q, u, v)$ *yields in one step* a configuration $c' = (q', u', v')$ if and only if the following hold:

1. $p = q'$.

2. If $D =\rightarrow$, then u' is given by u, with last symbol replaced by ρ, and the last symbol of v appended. Further, v' is equal to v, with the first symbol of v removed.

3. If $D =\leftarrow$, then u' is equal to u, with σ removed. Also, v' is equal to v, with ρ added at the start.

4. If $D = -$, then u' is equal to u, with σ replaced by ρ. Also, $v' = v$.

Then, we say that a configuration c_1 *yields* (or *reaches*) a configuration c_n, if and only if there is a sequence of configurations $(c_1, \ldots, c_n)$, such that for every $i < n$, we have that c_i yields c_{i+1} in one step. Furthermore, a *run* of M on input x is the sequence of subsequent configurations reached by $(\triangleright, s, x)$. Then, we say that M *halts*, or *converges*, on x, if the run of M on x reaches one of the three states: "yes", "no" or h. Otherwise, we say that it *diverges*. Further, if it reaches the state "yes", then we say that M *accepts* x, and if it reaches "no" that it *rejects* x.

We are now ready to formally define the class **NP**. Given a language L, we denote as a *polynomial decider machine for L* a Turing machine M_L, such that $L = \{w : M_l \text{ accepts } w \cdot c \text{ for some string } c\}$, with $\#c \in \mathcal{O}(1)$. We further say that c is a *certificate* for w.

Definition 3.8. *A problem R is in the class* **NP** *if there exists a polynomial decider machine for L_R.*

Problems Used in Computational Complexity Proofs. Here, we provide the definitions of computational problems, which we later use in proofs. We say that a *literal* is a propositional variable, or its negation. Also, we call a disjunction of literals a *clause*. Then, we say that a propositional formula φ is in *conjunctive normal form* (CNF), if φ is a conjunction of clauses. Furthermore, we say that φ is in 3-CNF, if it is in CNF, and each of its clauses is formed by at most three literals.

The well-known **NP**-complete problem 3-SAT (see, e.g., Karp [1972]) concerns checking if there exists an assignment of binary truth values to the variables in a formula φ in 3-CNF, under which φ is true, i.e., whether it is *satisfiable*.

3-SAT:
Input: 3-CNF formula φ.
Question: Is φ satisfiable?

In some of our proofs, we use an **NP**-complete variation of 3-SAT, in which the number of occurrences of each literal in a formula is limited. We say that a propositional formula φ is in 2P2N-3-CNF, if it is in 3-CNF, and for each variable x in φ, it holds that X appears in φ twice in the positive form and twice in the negative form. The **NP**-complete problem 2P2N-3-SAT concerns checking if a formula in 2P2N-3-CNF is satisfiable (see Berman et al. [2004]).

2P2N-3-SAT:
Input: 2P2N-3-CNF formula φ.
Question: Is φ satisfiable?

Another variation of 3-SAT which we will use is the **NP**-complete problem Planar 3-SAT (see, e.g., Lichtenstein [1982]). For a formula φ, we define an *incidence graph* of φ, in which a variable x is adjacent to a clause C, if and only if x, or $\neg x$, appears in φ. Then, in Planar 3-SAT, it is determined whether a formula φ with a planar incidence graph is satisfiable (see Lichtenstein [1982]).

Planar 3-SAT:
Input: φ in 3-CNF with a planar incidence graph.
Question: Is φ satisfiable?

Another useful **NP**-complete problem which we use in our proofs concerns finding cliques of a given size in a graph. So, k-Clique is the problem of checking whether there exists a clique of size k in an input graph (see, e.g., Papadimitriou [1994]).

k-Clique:
Input: Graph G, $k \in \mathbb{N}$.
Question: Is there a clique of size k in G?

Similarly, k-Independent Set is the problem of checking if there is an independent set of size k in an input graph. This problem has also been shown to be **NP**-complete (see, e.g., Papadimitriou [1994]).

k-Independent Set:
Input: Graph G, $k \in \mathbb{N}$.
Question: Is there an independent set of size k in G?

We will also use the **NP**-complete problem ILP-Feasibility. There, an input is a matrix A and a vector b. Then, it is checked if there exists a vector $\bar{x}$, which satisfies all inequalities given by A and b (see, e.g., Lenstra [1983]).

ILP-Feasibility:
Input: Matrix $A \in \mathbb{Z}^{m \times p}$ and a vector $b \in \mathbb{Z}^{m \times 1}$.
Question: Is there a vector $\bar{x} \in \mathbb{Z}^{p \times 1}$ satisfying the m inequalities given by A, i.e, $A \cdot \bar{x} \leqslant b$?

We will further use the following **PSPACE**-complete problem (see, e.g., Theorem 19.9 in Papadimitriou [1994]), in which it is decided whether a Turing machine accepts an input using limited space.

In-place Acceptance:
Input: Turing machine M, input x.
Question: Does M accept x without leaving $\#x + 1$ first symbols of its string?

Parametrised Complexity. Here, we introduce basic notions relevant to the *parametrised complexity* of computational problems. This type of complexity measurement is especially relevant to the analysis of problems which are hard to compute in the general case. Intuitively, we are interested in establishing how difficult is a problem, if some parameter of the input is small. For an extensive overview of parametrised complexity see, e.g., Cygan et al. [2015].

To account for the fact that a problem can be solved efficiently, i.e., in polynomial time, if a parameter is small, we employ the notion of *fixed-parameter tractability.* We say that a problem with an input n is fixed-parameter tractable (FPT), or that it is in the class **FPT**, for a parameter k, if it is solvable in time $\mathcal{O}(f(k)) \cdot \#n^{\mathcal{O}(1)}$, for some computable function f.

It is important to note that ILP-Feasibility is in **FPT**, when parametrised by the number of variables.

Proposition 3.1 (Lenstra [1983]; Kannan [1987]; Frank and Tardos [1987]). ILP-Feasibility *can be solved using* $\mathcal{O}(p^{\mathcal{O}(p)} \cdot L)$ *arithmetic operations and space polynomial in* L*, where* L *is the number of bits in the input and* p *is the number of variables.*

Moreover, we say that a problem is in **XP** for a parameter k, if there exists an algorithm solving this problem, which runs in time $n^{f(k)}$ (called an **XP**-*algorithm*), where f is some computable function. Note that **FPT**$\subseteq$ **XP**.

We are further interested in problems which are not in **FPT** for a specific parameter. The **W**-hierarchy defines a series of complexity classes extending **XP**. We note that showing that a problem is hard for a class $W[i]$ in this hierarchy, with $i \geqslant 1$, is an evidence that the problem is unlikely to be in **FPT**. In the context of this thesis, we say that a problem R is **W**[1]-hard, when parametrised by r, if there is a many-one reduction from the k-Clique problem computable in time $f(k) \cdot \#n^{\mathcal{O}(1)}$, where n is the instance of k-Clique, with parameter r not larger than $g(k)$, for some function g. We note that both k-Clique and k-Independent Set are **W**[1]-hard. Finally, we say that a problem is *para*-**NP**-*hard*, if it is already **NP**-complete for a constant value of the parameter r.

Tree Decomposition. Tree width is a fundamental graph parameter, useful for the design of parametrised algorithms, which will play an important role in our parameterised complexity analysis. Intuitively, this measurement indicates how "close" a graph is to a tree. Then, an **FPT** algorithm for a problem parametrised by the tree width implies a polynomial-time algorithm on "tree-like" graphs. Given a graph G, let $E(G)$ denote the edge set of G. For a rooted tree T and a non-root vertex $t \in V(T)$, by *parent*(t) we denote the parent of t in the tree T. Similarly, *child*(t) is a set of vertices such that t is a parent of each of its members. For vertices $u, t \in T$, we say that u is a *descendant* of t, denoted $u \preceq t$, if t lies on the unique path from u to the root. Note that every vertex is its own descendant. If $u \preceq t$ and $u \neq t$, then we write $u \prec t$. Likewise, u is an *ancestor* of t if t is a descendant of u.

Definition 3.9. *A* tree decomposition *of a graph* G *is a pair* (T, β) *of a tree* T *(whose vertices are called* nodes*) and a function* $\beta : V(T) \to 2^{V(G)}$*, such that:*

1. $\bigcup_{t \in V(T)} \beta(t) = V(G)$.
2. *For every edge* $e \in E(G)$*, there exists a node* $t \in V(T)$*, such that both endpoints of* e *belong to* $\beta(t)$*.*
3. *For every vertex* $v \in V(G)$*, the subgraph of* T *induced by the set* $T_v = \{t \in V(T) : v \in \beta(t)\}$ *is a connected tree.*

We say that the width *of* (T, β) *is* $\max_{v \in V(T)} \#\beta(v) - 1$. *Then, the* tree width *of* G*, which we also refer to as* $tw(G)$*, is the minimum width of a tree decomposition of* G.

Let now (T, β) be a tree decomposition of a graph G. We always assume that T is a rooted tree and so, we have a natural parent-child and ancestor-descendant relationship among vertices in T. We call the set $\beta(t)$ the *bag* at node t. Then, for a node $t \in V(T)$, by V_t, we denote the set $\bigcup_{t' \preceq t} \beta(t')$, i.e., the set of all the vertices in the bags in the subtree of T rooted at t.

When designing algorithms using tree decompositions, it is generally helpful to work with a special kind of tree decomposition, i.e., a *nice tree decomposition.*

Definition 3.10. Let (T, β) be a tree decomposition of a graph G, where r is the root of T. The tree decomposition (T, β) is called a *nice tree decomposition*, if the following conditions are satisfied.

1. $\beta(r) = \emptyset$ and $\beta(\ell) = \emptyset$ for every leaf node ℓ of T.

2. Every non-leaf node (including the root node) t of T is of one of the following types:

    - **Introduce node:** The node t has exactly one child t' in T and $\beta(t) = \beta(t') \cup \{v\}$, where $v \notin \beta(t')$.
    - **Forget node:** The node t has exactly one child t' in T and $\beta(t) = \beta(t') \setminus \{v\}$, where $v \in \beta(t')$.
    - **Join node:** The node t has exactly two children t_1 and t_2 in T, where $\beta(t) = \beta(t_1) = \beta(t_2)$.

We note that, using a well-known, polynomial-time algorithm, we can convert any given tree decomposition to a nice tree decomposition of the same width (Cygan et al. [2015]). We note that graphs of bounded tree width are sparse. That is, the number of edges in a graph with n vertices, and of tree width k, is $\mathcal{O}(k^2)$. On the other hand, graphs of bounded neighbourhood diversity can be dense. For instance, a complete graph has a ND of 1, but has n^2 edges. Moreover, note that ND is "incomparable" with tree width. That is, there are graphs of constant ND with unbounded tree width (e.g., a clique) and graphs of constant tree width with unbounded ND (e.g., a path).

Clique Width. Another important parameter which we use in the thesis, is the *clique width.* This notion is a generalisation of tree width. See Dabrowski et al. [2019]

for an overview of key results regarding this notion. First, we define the *disjoint union* of two graphs, G_1 and G_2, as a graph $(V(G_1) \cup V(G_2), E(G_1) \cup E(G_2))$. We denote it as $G_1 \oplus G_2$. In words, a disjoint of G_1 and G_2 contains these two graphs, with no edges between them. Then, we define the clique width of a graph G, as the *minimum number of labels* sufficient to construct G using the following operations:

1. Construct a graph with a single vertex v, which is labelled i.
2. Construct a disjoint union of two labelled graph, G_1 and G_2.
3. For a pair of distinct labels i and j, construct an edge between every pair of vertices v_i, and v_j, such that v_i is labelled i, while v_j is labelled j.
4. Relabel each vertex with a label i with a label j.

In Example 3.5 we show how a graph can be constructed using those operations.

Example 3.5. *Let us show how a pair of vertices can be constructed using the operations used in the definition of clique width. First, we create a graph with vertex labelled b. Subsequently, we construct a graph with a vertex labelled r, and create a disjoint union of these structures. As a final step, we link the vertex labelled r to the vertex labelled b. Figure 3.10 depicts this construction.*

Figure 3.10: Example of construction of a graph using operations following from the definition of clique width.

Boolean Circuits. Let us define Boolean circuits, basing on Definition 4.4 in Papadimitriou [1994]. We start with what we call the *syntax* of a circuit. A Boolean circuit is a directed graph $C = (V, E)$, where the vertices in $V = \{1, \ldots, n\}$ are called *gates* of C. We further assume that C is acyclic. Also, all vertices in the graph have indegree $0, 1$ or 2. Then, each gate $i \in V$ has a *sort* $s(i)$, where $s(i)$ belongs to $\{\text{TRUE}, \text{FALSE}, \text{AND}, \text{OR}, \text{NOT}\} \cup \{x_0, x_1, \ldots\}$.

If $s(i) \in \{\text{TRUE}, \text{FALSE}\} \cup \{x_0, x_1, \ldots\}$, then the indegree of i is 0. Gates with indegree 0 are called *inputs* of C. Further, if $s(i) = \text{NOT}$, then the indegree of i is 1. Also, if $s(i) \in \{\text{AND}, \text{OR}\}$, then the indegree of i is 2. Finally, a vertex n with no outgoing edges is called an *output* of C.

Figure 3.11 depicts an example of a Boolean circuit with three inputs. Observe that the syntax of this circuit corresponds to the formula $(x_0 \vee (x_1 \wedge \neg x_2)) \wedge \neg x_2$.

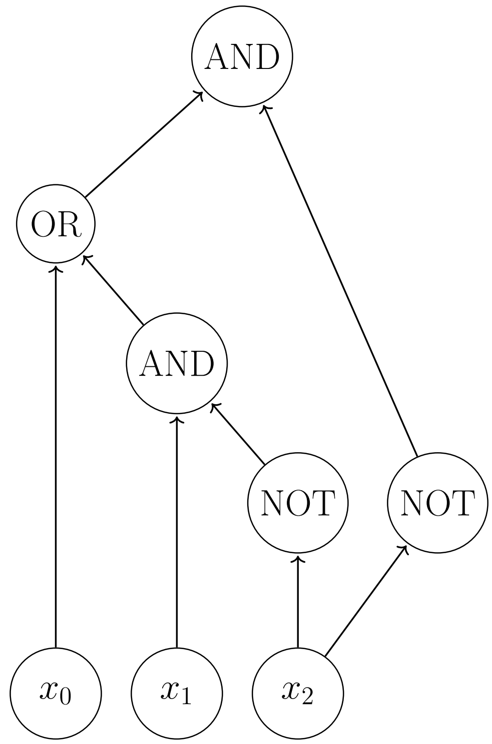

Figure 3.11: A Boolean circuit corresponding to the formula $(x_0 \vee (x_1 \wedge \neg x_2)) \wedge \neg x_2$.

Let us now define the *semantics* of a circuit. We will specify exactly one truth value, i.e., TRUE or FALSE, to each gate of a circuit, for each valuation of propositional variables. We denote as $X(C)$ the set of all propositional variables in C. So, $X(C) = \{x \in X : x = s(i) \text{ for some } i \in V\}$. Given a valuation T over $X(C)$, we define the *truth value* of a gate $i \in V$ $(T(i))$, by induction on i. If $s(i) = \text{TRUE}$, then $T(i)$ is TRUE, and if $s(i) = \text{FALSE}$, then $T(i)$ is FALSE. Further, if $s(i) \in X(C)$, then, if i is true in T, we have that $T(s(i)) = \text{TRUE}$, and $T(s(i)) = \text{FALSE}$ otherwise. Let us now observe that, if $s(i) = \text{NOT}$, then there is a unique gate j, such that $(j, i) \in E$, i.e., there is an edge incoming to i from j. Then, $T(i) = \text{TRUE}$, if $T(j) = \text{FALSE}$, and $T(i) = \text{FALSE}$ otherwise. Now, if $s(i) = \text{OR}$, then there are two edges, (j, i) and (j', i), incoming to i. Then, $T(i) = \text{TRUE}$ if and only if $T(j) = \text{TRUE}$ or $T(j') = \text{TRUE}$. Finally, if $s(i) = \text{AND}$, while (j, i) and (j', i) are the edges incoming to i, then $T(i) = \text{TRUE}$ if and only if $T(j) = \text{TRUE}$ and $T(j') = \text{TRUE}$. Observe how, for the circuit presented in Figure 3.11, assigning the truth value FALSE to the gate x_2, and TRUE to both x_0 and x_1, results in all the other gates being assigned the value TRUE.

Chapter 4

Nominee Selection in Hotelling-Downs Spaces

4.1 Introduction

The Hotelling-Downs model, introduced by Hotelling [1929], is perhaps the most impactful and well-established framework to study strategic positioning of self-interested players on a spacial dimension, e.g., candidates on a political spectrum. In Hotelling and Downs's original setup, two self-interested ice cream vendors strategically place themselves on a beach so as to attract as many customers as possible, in the knowledge that the relaxed beachgoers will always opt for the one closer to them. This game has a unique NE, in which both agents choose the most central location. The simplicity and depth of this observation has led to applications in corporate strategy, strategic candidacy, and spatial design (see, e.g., Eiselt [2011]). Downs [1957] himself mentions the potential of this framework to predict how parties will set their agendas and how they will position themselves in the political spectrum, suggesting that party politics will tend to more moderate choices. Notice that this framework is naturally suited to the study of strategic nominee selection, where the policy of a party corresponds to political views of their chosen candidate.

It is fair to say that Downs's observation relies on severely restrictive assumptions. First and foremost, his model involves only two agents. Indeed, it has been shown that, if the number of agents is increased, there are cases without Nash equilibria (Eaton and Lipsey [1975]). Second, and perhaps more importantly, agents are allowed unrestricted movement. While this might be a reasonable assumption for ice cream vendors on a beach, this is certainly not the case for political parties, which can only count on a few potential nominees, typically tied up to relatively fixed political stances. Similarly, from an economics perspective, producers might be limited to a fixed number of products which can potentially be released.

To address this gap we analyse the extension of the Hotelling-Downs model, in which parties' can select their position on the line only from a finite set of positions, corresponding to the views of their potential nominees. Importantly, we do not presuppose any restrictions on the locations of candidates within a party.

Even though primaries were recently studied from the perspective of multi-agent systems in Borodin et al. [2019], surprisingly, variations of the Hotelling-Downs model involving multiple participants with restricted options – and which could thus capture the mathematics behind real-world situations – have been largely overlooked. A notable exception is the work by Sabato et al. [2017] on *real candidacy games*, where competing candidates select intervals on a line and are then chosen based on a given social choice rule. Notwithstanding the similarities of this work with our framework, it also displays some important technical differences. In particular, their restricted action spaces, in our view are not suitable to model nominee selection, and they can force equilibrium existence. Furthermore, the mentioned paper does not consider computational complexity.

Our Contribution. In this chapter we provide a game-theoretic analysis of nominee selection, where parties choose independently and simultaneously from their respective pools of potential candidates. We assume that both voters and party candidates occupy a fixed position on a line, with voters always voting for the (nominated) candidate that is closest to them. We carry out an algorithmic analysis of verifying the existence of pure Nash equilibria, while focusing on the differences between two-party electoral competitions and those in which an arbitrary number of parties participates. Specifically, we show that if there are only two parties, then the problem of establishing whether a NE exists can be achieved in linear time (Theorem 4.1). By contrast, finding a NE is **NP**-complete in the multi-party case (Theorem 4.2). We also look at some natural restrictions, such as having parties with non-overlapping political spectra, and provide equilibrium existence results for these, as well.

Structure of the Chapter. In Section 4.2 we study the structural and algorithmic properties of Nash equilibria in games limited to two parties. Further, in Section 4.3 we analyse the case in which an arbitrary number of parties competes. Section 4.4 concludes with a discussion of our main findings and of some interesting directions for future research.

4.2 Games with Two Parties

In this section we carry out an analysis of Nash equilibria in games with two parties only. First, we study conditions under which a NE exists. Then, we focus on the complexity of checking if there is an equilibrium profile in a given game.

Existence. Our framework is a generalisation of the discrete version of Hotelling-Downs model where, as argued earlier, a NE is guaranteed to exist. To see this in our model, consider a $[0,k]$ line and two parties, P_1 and P_2. It is well known that if $P_1 = P_2 = [0,k]$, k is even and $f : [0,k] \to \mathbb{N}_0$ is uniform, then the game has a unique NE (see, e.g., Eaton and Lipsey [1975]). Moreover, such an equilibrium can be computed in $k-1$ rounds of iterated elimination of strictly dominated strategies. In this equilibrium both parties choose the central position, getting utility of $\frac{V(f)}{2}$ each. When k is odd instead, then every outcome in which parties select one of the central positions is a NE. With voters that are potentially non-uniformly distributed this fact is still true, provided that the notion of central position is replaced by that of *median* position, which we will define next.

A position $\mathbf{m} \in [0,k]$ is called a median, if $f(\mathbf{m}) > 0$, $\sum_{n \leqslant \mathbf{m}} f(n) \geqslant \frac{V(f)}{2}$, and $\sum_{n \geqslant \mathbf{m}} f(n) \geqslant \frac{V(f)}{2}$. In words, a median is a non-empty position, such that half of the voters is located there or on the left of it, and a half there or on the right of it. Intuitively, this is a position at which a median voter is located. Given a distribution of voters on a line, we denote as $\mathbf{m}_L$ its smallest median position. So, $\mathbf{m}_L$ is the leftmost position on the line, which is a median position. Then $\mathbf{m}_R$ is the largest, i.e., rightmost, median position. Furthermore, for simplicity, if the median position is unique, we simply refer to it as $\mathbf{m}$. It is worth noting that median positions always exist, as we assume that $V(f) > 0$. However, they do not need to be unique. Moreover, there are cases in which median positions do not come consecutively. To see this, consider a $[0,4]$ line and the distribution of voters $f : [0,k] \to \mathbb{N}_0$, such that $f(0) = f(4) = 1$ and for each $i \in [1,3]$, we have that $f(i) = 0$, as shown in Figure 4.1.

Then, both 0 and 4 are median positions, while they are not consecutive. Note also that we immediately have that in every election there are at most two median positions and that given positions $\mathbf{m}_L$ and $\mathbf{m}_R$, if $n \in [\mathbf{m}_L + 1, \mathbf{m}_R - 1]$, then $f(n) = 0$. In other words, all positions between $\mathbf{m}_L$ and $\mathbf{m}_R$ do not have any voters located there.

We now use the notion of a median position to show that a NE is guaranteed

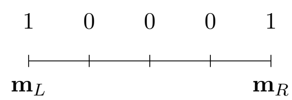

Figure 4.1: Example of a game with two median positions. Numbers above the line specify the number of voters at each position. Observe that positions $\mathbf{m}_L$ and $\mathbf{m}_R$, indicated below the line, are median positions, as for each of them exactly a half of voters is located there.

to exist, if the parties' choices are intervals, i.e., if, given a line $[0,k]$, for each party P_i, there are $p_l, p_r \in [0,k]$,such that $P = [p_l, p_r]$. Incidentally, this encodes the action space as studied in Sabato et al. [2017], under a very basic "voting" rule, in which the candidate which attracts the highest number of voters is the winner.

Furthermore, we fix the following definition. For candidates $p_1, p'_1 \in P_1$ and $p_2 \in P_2$, we say that p'_1 *is strictly closer to* p_2 *than* p_1, whenever either $p_1 < p'_1 < p_2$ or $p_2 < p'_1 < p_1$. Now we show the following useful lemmata. First, we show that for every strategy profile, in which parties do not choose the same position, it holds that changing a nominee to a one that is strictly closer to the opponent, does not lower the utility of a party.

Lemma 4.1. *Let $\mathbf{c} = (c_1, c_2)$ be a strategy profile with $c_1 \in P_1$ and $c_2 \in P_2$. Then, if there exists $c'_1 \in P_1$, such that c'_1 is strictly closer to c_2 than c_1, then it holds that $u_1(c'_1, c_2) \geqslant u_1(c_1, c_2)$.*

Proof. Take a line $[0,k]$, two parties P_1 and P_2, as well as a distribution of voters $f : [0,k] \to \mathbb{N}_0$. Also, take a strategy profile $\mathbf{c}$=(c_1, c_2), with $c_1 \in P_1$ and $c_2 \in P_2$. Further, let $c'_1 \in P_1$ be strictly closer to c_2 than c_1. Without loss of generality, we assume that $c_1 < c_2$ (the case of $c_1 > c_2$ is similar and $c_1 = c_2$ is impossible). Then, we have that c'_1 is such that $c'_1 > c_1$ and that $c'_1 < c_2$. Then, observe that $\sigma_{c_1}(m)$, i.e., an indication of the number of voters at position m, which are attracted by c_1, is at least as large, as $\sigma_{c'_1}(m)$, for all $m \in [0,k]$, fixing the choice of c_2. This implies that $u_1(c'_1, c_2) \geqslant u_1(c_1, c_2)$. □

Further, we show that if both parties nominate candidates located on the same side of a median position m, then the one which is closer to m attracts at least half of the voters. This holds, as by the definition of a median position, we have that at least a half of the voters are located at each side of m.

Lemma 4.2. *For every line $[0,k]$, distribution of voters $f : [0,k] \to \mathbb{N}_0$, set of parties $P = \{P_1, P_2\}$, and strategy profile $\mathbf{c} = (c_1, c_2)$, it holds that: (1) if for some*

$c_i, c_j \in \boldsymbol{c}$, it holds that $c_j \leqslant c_i \leqslant \boldsymbol{m}_R$, then $u_{\mathcal{C}(c_i)}(c_1, c_2) \geqslant \frac{V(f)}{2}$, and (2) if for some $c_i, c_j \in \boldsymbol{c}$, we have that $\boldsymbol{m}_L \leqslant c_i \leqslant c_j$, then $u_{\mathcal{C}(c_i)}(c_1, c_2) \geqslant \frac{V(f)}{2}$.

Proof. Take a line $[0, k]$, distribution of voters $f : [0, k] \to \mathbb{N}_0$, set of parties $P = \{P_1, P_2\}$, and a strategy profile $\mathbf{c} = (c_1, c_2)$. Notice that if $c_1 = c_2$, then the claim follows immediately. Then, without loss of generality, suppose that $c_1 > c_2$. Suppose further, that (1) is the case. Observe that for every $n \in [\mathbf{m}_R, k]$, we have that $\sigma_{c_i}(n) = 1$. Also, we know that $\sum_{i \in [\mathbf{m}_R, k]} f(i) \geqslant \frac{V(f)}{2}$, and so the claim follows. The reasoning showing the claim for case (2) is symmetric. □

Now we are ready to show the existence of NE in interval models.

Proposition 4.1. *Let $f : [0, k] \to \mathbb{N}_0$ be the distribution of voters and P_1, P_2 be parties. If P_1 and P_2 are intervals, then there are $c_1 \in P_1$ and $c_2 \in P_2$, such that (c_1, c_2) is a NE.*

Proof. Take a line $[0, k]$, distribution of voters $f : [0, k] \to \mathbb{N}_0$ and parties P_1, P_2, such that P_1 and P_2 are intervals. We will show that there exists a NE profile $\mathbf{c}$=(c_1, c_2). Recall that $\mathbf{m}_L$ and $\mathbf{m}_R$ denote the median positions. If there is a single median position, then $\mathbf{m} = \mathbf{m}_L = \mathbf{m}_R$. Let us consider the following, exhaustive cases.

Case 1: $\{\mathbf{m}_L, \mathbf{m}_R\} \cap P_1 \neq \emptyset$ and $\{\mathbf{m}_L, \mathbf{m}_R\} \cap P_2 \neq \emptyset$. Then, take $\mathbf{m}_1 \in \{\mathbf{m}_L, \mathbf{m}_R\}$, such that $\mathbf{m}_1 \in P_1$ and $\mathbf{m}_2 \in \{\mathbf{m}_L, \mathbf{m}_R\}$, such that $\mathbf{m}_2 \in P_2$. Then, let us show that $(\mathbf{m}_1, \mathbf{m}_2)$ is a NE. Suppose it is not and assume without loss of generality that there is a candidate $c_1' \in P_1$, such that $u_1(c_1', \mathbf{m}_2) > u_1(\mathbf{m}_1, \mathbf{m}_2)$. Observe that, by Lemma 4.2, it holds that $u_1(\mathbf{m}_1, \mathbf{m}_2) \geqslant \frac{V(f)}{2}$. But then it also holds that $u_2(\mathbf{m}_1, \mathbf{m}_2) \geqslant \frac{V(f)}{2}$, which implies that $u_2(\mathbf{m}_1, \mathbf{m}_2) \leqslant \frac{V(f)}{2}$. So, $u_1(\mathbf{m}_1, \mathbf{m}_2) = u_2(\mathbf{m}_1, \mathbf{m}_2) = \frac{V(f)}{2}$. Then, by Lemma 4.2, for every value of c_1', we have that $u_2(c_1', \mathbf{m}_2) \geqslant \frac{V(f)}{2}$. So, $u_1(c_1', \mathbf{m}_2) \leqslant \frac{V(f)}{2}$. Hence, c_1' is not a profitable deviation, which contradicts the assumptions.

Case 2: $\{\mathbf{m}_L, \mathbf{m}_R\} \cap P_1 = \emptyset$ or $\{\mathbf{m}_L, \mathbf{m}_R\} \cap P_2 = \emptyset$. Without loss of generality, let $\{\mathbf{m}_L, \mathbf{m}_R\} \cap P_1 = \emptyset$. Notice that as P_1 and P_2 are intervals, it holds that either (1) $P_1 \cap [\mathbf{m}_L + 1, \mathbf{m}_R - 1] = \emptyset$ and $P_1 \cap [\mathbf{m}_L, \mathbf{m}_R] = \emptyset$, (2) $P_1 \subseteq [\mathbf{m}_L, \mathbf{m}_R]$, while $P_2 \cap [\mathbf{m}_L, \mathbf{m}_R] \neq \emptyset$ or $P_2 \subseteq [\mathbf{m}_L, \mathbf{m}_R]$, while $P_1 \cap [\mathbf{m}_L, \mathbf{m}_R] \neq \emptyset$, or (3) for all $p \in P_1 \cup P_2$, it is the case that $p \in [\mathbf{m}_L, \mathbf{m}_R]$.

Case 2.1: $P_1 \cap [\mathbf{m}_L, \mathbf{m}_R] = \emptyset$ and $P_1 \cap [\mathbf{m}_L, \mathbf{m}_R] = \emptyset$. Without loss of generality, we assume that, for all $p \in P_1$, $p < \mathbf{m}_L$. Let us consider three exhaustive cases: (1) for some $p \in P_2$, $p > \max(P_1)$, (2) for all $p \in P_2$, $p < \max(P_1)$, (3) $\max(P_1) = \max(P_2)$.

Case 2.1.1: for some $p \in P_2$, $p > \max(P_1)$. Take the smallest such $p \in P_2$ (denote it as s_2). Notice that $\max(P_1)$ is a best response to s_2 by Lemma 4.1. Also, if $\min(P_2) > \max(P_1)$, then s_2 is a best response to $\max(P_1)$, by Lemma 4.1. Moreover, notice that by Lemma 4.2 we have that if $\min(P_2) \leqslant \max(P_1)$, then $u_2(\max(P_1), s_2) \geqslant \frac{V(f)}{2}$, as $\max(P_1) < \mathbf{m}_L$ and $\max(P_1) < s_2 \leqslant \mathbf{m}_L$ (as P_1 and P_2 are intervals). Also, by Lemma 4.2, $u_2(\max(P_1), p_2) \leqslant \frac{V(f)}{2}$, for every $p_2 \in P_2$, such that $p_2 \leqslant s_2$, as $\max(P_1) < \mathbf{m}_L$ and $p_2 \leqslant \max(P_1)$. So, $(\max(P_1), s_2)$ is a NE.

Case 2.1.2: for all $p \in P_2$, $p < \max(P_1)$. Take the smallest $p \in P_1$, such that $p > \max(P_2)$ (denote it s_1). Notice that the profile $(s_1, \max(P_2)$ is a NE, by Lemma 4.2 and by Lemma 4.1, similarly to how we argued above.

Case 2.1.3: $\max(P_1) = \max(P_2)$**.** Then, the profile $(\max(P_1), \max(P_2))$ is a NE, by Lemma 4.2, as $\max(P_1) < \mathbf{m}_L$, and for every $p_2 \in P_2$, $p_2 \leqslant \max(P_1)$.

Case 2.2: $P_1 \subseteq [\mathbf{m}_L, \mathbf{m}_R]$, while $P_2 \cap [\mathbf{m}_L, \mathbf{m}_R] \neq \emptyset$ or $P_2 \subseteq [\mathbf{m}_L, \mathbf{m}_R]$, while $P_1 \cap [\mathbf{m}_L, \mathbf{m}_R] \neq \emptyset$. Let us assume without loss of generality that $P_1 \cap [\mathbf{m}_L, \mathbf{m}_R] = \emptyset$, while $P_2 \cap [\mathbf{m}_L, \mathbf{m}_R] \neq \emptyset$. Let us consider the following cases, similar to the case 2.1.

Case 2.2.1: for some $p \in P_2$, it holds that $p > \max(P_1)$. Then, take the smallest such p and call it s_2. Further, observe that $\max(P_1)$ is a best response to s_2, by Lemma 4.1. Now, we show that $(\max(P_1), s_2)$ is a NE, by demonstrating that s_2 is a best response to $\max(P_1)$. To see that let us first observe that if $\min(P_2) > \max(P_1)$, then s_2 is a best response to $\max(P_1)$, by Lemma 4.1. Also, if $\min(P_2) \leqslant \max(P_1)$, then the claim follows by Lemma 4.2, as $\max(P_1) < \mathbf{m}_R$.

Case 2.2.2: $\max(P_2) < \max(P_1)$**.** Then, take the smallest $s_1 \in P_1$, such that $s_1 > \max(P_2)$. Observe that by assumption it holds that $s_1 < \mathbf{m}_R$. Hence, $u_1(s_1, \max(P_2)) \geqslant \frac{V(f)}{2}$, by Lemma 4.2. Now notice that since P_1 is an interval, for every $p \in P_1$, such that $p < s_1$, it holds that $p \leqslant \max(P_2)$. So, by Lemma 4.2, $u_2(\max(P_1), p) \leqslant \frac{V(f)}{2}$, as $\max(P_1) \leqslant \mathbf{m}_R$. Finally, for every $p \in P_1$, such that $p > s_1$, we have that $u_1(p, \max(P_2)) \leqslant u_2(\max(P_1), s_2)$, by Lemma 4.1. It follows now that s_1 is a best response to $\max(P_2)$. It follows by Lemma 4.1 that $(s_1, \max(P_2))$ is a NE.

Case 2.2.3: $\max(P_1) = \max(P_2)$**.** Then, $(\max(P_1), \max(P_2))$ is a NE by Lemma 4.2, since $\max(P_1) > \mathbf{m}_L$ and $\max(P_1) < \mathbf{m}_R$.

Case 2.2.3: for all $p \in P_1 \cup P_2$, it is the case that $p \in [\mathbf{m}_L + 1, \mathbf{m}_R - 1]$. Take an arbitrary profile (c_1, c_2). Observe that then, by Lemma 4.2, we have that $u_1(c_1, c_2) = u_1(c_1, c_2) = \frac{(V(f)}{2}$, while for each c_1', c_2' it holds that $u_1(c_1', c_2) \leqslant \frac{(V(f)}{2}$ and $u_1(c_1, c_2') \leqslant \frac{(V(f)}{2}$. So, (c_1, c_2) is a NE. □

Critically, the existence of Nash equilibria is no longer guaranteed in the general setting, where parties are not necessarily given by intervals, which is arguably a more realistic representation of nomination processes.

Proposition 4.2. *There are games with two parties and no Nash equilibria, even when the distribution of voters is uniform.*

Example 4.1. *We will provide an example showing that Proposition 4.2 holds, i.e., that for some games with two parties competing and with uniform distribution of voters, in which no strategy profile is a NE. Consider the line* $[0, 8]$ *of uniformly distributed voters and two parties* P_1, P_2*, with* $P_1 = \{1, 7\}$*,* $P_2 = \{2, 6\}$*, as depicted in Figure 4.2.*

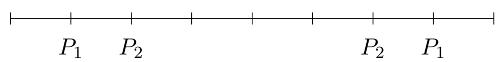

Figure 4.2: Game used in the Example 4.1, where voters are uniformly distributed. Below the line positions of candidates from parties P_1 and P_2.

For simplicity, we model the instance as the normal form game in Figure 4.3. Then, by examining the representation of the game, we get that this game has no NE.

	2	6
1	2 \ 7	4 \ 5
7	4 \ 5	2 \ 7

Figure 4.3: Normal form representation, with rows representing P_1's choices and column P_2's. The matrix entries encode the utilities as a function of the length of the line $[0, k]$ and the distribution of voters $f : [0, k] \to \mathbb{N}_0$.

Notice that following Proposition 4.2 the existence of a NE is not guaranteed in our framework even in the simplest case of games with two parties and a uniform distribution of voters. This motivates the need for an algorithmic analysis of deciding the existence of Nash equilibria in our framework.

Computation

We now move to study of the complexity of checking whether a NE exists or not, in a given game in which only two parties compete. While it is straightforward to see that a polynomial-time algorithm exists for this problem (simply try all possible profiles and check if any is a NE), we will provide a linear-time algorithm for this case. Moreover, the procedure we present will also return an equilibrium profile, whenever one exists. Before presenting it and proving its soundness, we show that if elections with two parties admit a NE, then they also admit a NE in which one of the parties selects a candidate close to the median position. This observation will constitute the core of our algorithm.

We start with defining the concept of *most central candidates*. Consider a party P_i on a line [0,k], with a distribution of voters $f : [0,k] \to \mathbb{N}_0$. We denote the set $C_i = \{p : p \in \{L_i^L, L_i^R, R_i^L, R_i^R\}\}$, where $L_i^L, L_i^R, R_i^L, R_i^R \in P_i$ and

$$\begin{aligned}
L_i^L &= \operatorname*{argmin}_{\{p \in P_i:\ p \leqslant \mathbf{m}_L\}} |\mathbf{m}_L - p| \\
L_i^R &= \operatorname*{argmin}_{\{p \in P_i:\ p \in [\mathbf{m}_L+1, \mathbf{m}_R-1]\}} |p - \mathbf{m}_L| \\
R_i^L &= \operatorname*{argmin}_{\{p \in P_i:\ p \in [\mathbf{m}_L+1, \mathbf{m}_R-1]\}} |\mathbf{m}_R - p| \\
R_i^R &= \operatorname*{argmin}_{\{p \in P_i:\ p \geqslant \mathbf{m}_R\}} |p - \mathbf{m}_R|
\end{aligned}$$

as P_i's most central candidates.

In words, P_i's most central candidates are those that are closest to the left (i.e., L_i^L and L_i^R) and the right (i.e., R_i^L and R_i^R) median voter positions. Notice that for every party P, we have that the set of their most central candidates has the cardinality of at least 1, as we assume that parties are not empty. However, the cardinality of C_i may vary. For instance, if the median voter position is unique and a party has a candidate that is exactly there, then the set of most central candidates of a party is a singleton.

Figure 4.4 illustrates the central candidates of a party P_1 made up by candidates p_1, p_2, p_3 and p_4 on a line [0,6].

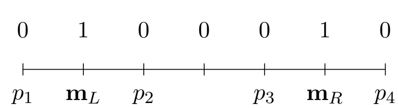

Figure 4.4: Example of a game with a party P_1, depicting its central candidates. Observe that p_1, p_2, p_3 and p_4 from P_1 are the L_1^L, L_1^R, R_1^L, and R_1^R candidates respectively, as $\mathbf{m}_L = 1$ and $\mathbf{m}_R = 5$.

We show now that if a NE exists, then there is a NE in which at least one of the two parties selects a most central candidate. This allows us to show that a strategy profile is a NE, if both parties select candidates located between median positions. First, we show the following lemma.

Lemma 4.3. *Let $[0,k]$ be a line, $f:[0,k]\to\mathbb{N}_0$ be a distribution of voters, P_1, P_2 be parties, and (c_1, c_2) be a strategy profile, such that $c_1, c_2 \in [\boldsymbol{m}_L, \boldsymbol{m}_R]$. Then, (c_1, c_2) is a NE.*

Proof. Take a line $[0,k]$, distribution of voters $f:[0,k]\to\mathbb{N}_0$, parties P_1, P_2, and a strategy profile (c_1, c_2), such that $c_1, c_2 \in [\mathbf{m}_L, \mathbf{m}_R]$. Notice that by Lemma 4.2, for every profile (c_1', c_2'), such that $c_1', c_2' \in [\mathbf{m}_L, \mathbf{m}_R]$, it holds that $u_1(c_1', c_2') = u_2(c_1', c_2') = \frac{V(f)}{2}$. Further, assume without loss of generality that $c_1 \leqslant c_2$. We now show that c_1 is a best response to c_2, i.e., that for every $c_1' \in P_1$, we have that $u_1(c_1', c_2) \leqslant u_1(c_1, c_2)$. Indeed, if $c_1' < \mathbf{m}_L < c_1$, then the claim holds by Lemma 4.1 and Lemma 4.2. Also, one can verify that, by Lemma 4.2, if $c_1' \in [\mathbf{m}_L, \mathbf{m}_R]$, then $u_1(c_1', c_2) = u_2(c_1, c_2) = \frac{V(f)}{2}$. Finally, if $c_1' > \mathbf{m}_R$, then, also by Lemma 4.2, we have that $u_1(c_1', c_2) \leqslant \frac{V(f)}{2}$, so the claim follows. Analogously it can be shown that c_2 is a best response to c_1. □

We will further show that, if there is a NE in a given election, then there is a NE, in which at least one of the parties selects a most central candidate.

Lemma 4.4. *For every line $[0,k]$, distribution of voters $f:[0,k]\to\mathbb{N}_0$ and parties P_1, P_2, if there is a NE profile (c_1, c_2), then there is a NE profile (c_1', c_2'), such that $c_1' \in C_1$ or $c_2' \in C_2$.*

Proof. Take the line $[0,k]$, distribution of voters $f:[0,k]\to\mathbb{N}_0$, and parties P_1, P_2. Suppose that there is a NE profile $\mathbf{c} = (c_1, c_2)$. Now we show that there is a NE profile (c_1', c_2'), such that $c_1' \in C_1$ or $c_2' \in C_2$. Without loss of generality, we assume that $c_1 \leqslant c_2$.

Case 1: $c_1 \in C_1$ or $c_2 \in C_2$. The claim follows immediately.

Case 2: $c_1 \notin C_1$ and $c_2 \notin C_2$. We will consider the following cases:

1. $c_1, c_2 \in [\mathbf{m}_L + 1, \mathbf{m}_R - 1]$.
2. $c_1, c_2 < \mathbf{m}_L$.
3. $c_1, c_2 > \mathbf{m}_R$.
4. $c_1 < \mathbf{m}_L < c_2$.
5. $c_1 < \mathbf{m}_R < c_2$

Note that they are exhaustive, as we assume that $c_1 < c_2$, and as by assumption that they are not most central candidates, they are not located at a median position. Then, indeed, we either have that both c_1 and c_2 are located between $\mathbf{m}_L$ and $\mathbf{m}_R$ (case 1), that they are strictly smaller than $\mathbf{m}_L$ (case 2) or greater than $\mathbf{m}_R$ (case 3) or that they are on the opposite sides of a median position (cases 4 and 5).

Case 2.1: $c_1, c_2 \in [\mathbf{m}_L + 1, \mathbf{m}_R - 1]$**.** Then, (L_1^R, R_2^L) is a NE by Lemma 4.3.

Case 2.2: $c_1, c_2 < \mathbf{m}_L$**.** Assume first that $c_1 = c_2$, and notice that $u_1(c_1, c_2) = u_2(c_1, c_2) = \frac{V(f)}{2}$. Further, consider the position L_2^L. Notice that it exists, since $c_2 < \mathbf{m}_L$, while $c_2 \neq L_2^L$. We will now show that (c_1, L_2^L) is a NE. Notice that as $\mathbf{c}$ is a NE, $u_2(c_1, L_2^L) \leqslant u_2(c_1, c_2)$. So, by Lemma 4.2, $u_2(c_1, L_2^L) = u_2(c_1, c_2) = \frac{V(f)}{2}$. Therefore, as c_2 is a best response to c_1, so is L_2^L. To show that c_1 is a best response to L_2^L as well, assume towards contradiction that $u_1(c_1', L_2^L) > u_1(c_1, L_2^L)$, for some $c_1' \in P_1$. Observe that $u_1(c_1, L_2^L) = \frac{V(f)}{2}$, since (c_1, c_2) is a NE. So, as $c_1 < L_2^L \leq \mathbf{m}_L$, by Lemma 4.2 it holds that $u_1(c_1', L_2^L) \leqslant \frac{V(f)}{2}$, for every $c_1' \in P_1$, such that $c_1' < c_1$. Hence, as we know that $u_1(c_1', L_2^L) > \frac{V(f)}{2}$, we get that $c_1' > L_2^L$. But then, by Lemma 4.1 we get that $u_2(c_1', L_2^L) \geqslant u_2(c_1', c_2)$, and thus $u_1(c_1', L_2^L) \leqslant u_1(c_1', c_2)$. Hence, $u_1(c_1', c_2) > u_1(c_1, c_2)$, as $u_1(c_1, c_2) = \frac{V(f)}{2}$ and $u_1(c_1', L_2^L) > \frac{V(f)}{2}$. This is however impossible, as (c_1, c_2) is a NE.

Assume further that $c_1 \neq c_2$. Recall that, by assumption, $c_1 < c_2$. Then, consider position L_1^L and the profile (L_1^L, c_2). Notice that L_1^L exists, since $c_1 < \mathbf{m}_L$ and $c_1 \notin C_1$. We further show that, as c_1 is a best response to c_2, so is L_1^L. Indeed, if $c_1 < L_1^L < c_2$, then it holds that L_1^L is also a best response by Lemma 4.1. Observe that in such a case, c_2 is also a best response to L_1^L, as otherwise (c_1, c_2) would not be a NE. Also notice that if $c_1 < c_2 < \mathbf{m}_L$, then by Lemma 4.2 we have that $u_1(c_1, c_2) \leqslant \frac{V(f)}{2}$. Observe further that by Lemma 4.2 it holds that if $L_1^L \geqslant c_2$, then $u_1(L_1^L, c_2) \geqslant \frac{V(f)}{2}$. It then follows from Lemma 4.1 that L_1^L is a best response to c_2. But then we have that by Lemma 4.2, $u_2(c_1, c_2) \geqslant \frac{V(f)}{2}$. Hence, $u_2(L_1^L, c_2) \geqslant \frac{V(f)}{2}$, since (c_1, c_2) is a NE. But then it follows from Lemma 4.2 that c_2 is a best response to L_1^L.

Case 2.3: $c_1, c_2 > \mathbf{m}_R$**.** Reasoning is symmetric to case 2.2.

Case 2.4: $c_1 < \mathbf{m}_L < c_2$**.** Without loss of generality, let $\mathbf{m}_L \in \mathit{range}_{c_1}(\mathbf{c})$. Observe that then $u_1(\mathbf{c}) \geqslant \frac{V(f)}{2}$, and thus $u_1(\mathbf{c}) \geqslant u_2(\mathbf{c})$. Then, consider the position L_2^R, if it exists, and R_2^R otherwise. Observe that one of them exists, since $c_2 > \mathbf{m}_L$ and $c_2 \notin C_2$. Without loss of generality, we assume that L_2^R exists, and show that (c_1, L_2^R) is a NE. Observe that, by Lemma 4.1, we have that $u_2(c_1, c_2) \leqslant u_2(c_1, L_2^R)$. So, as (c_1, c_2) is a NE, we have that $u_2(c_1, c_2) = u_2(c_1, L_2^R) \leqslant \frac{V(f)}{2}$. Moreover, we know that $u_1(c_1, L_2^R) \geqslant \frac{V(f)}{2}$. Further, as c_2 is a best response to c_1, so is L_2^R, by Lemma 4.1. To show that c_1 is a best response to L_2^R as well, suppose towards contradiction that there is a $c_1' \in P_1$, such that $u_1(c_1', L_2^R) > u_1(c_1, L_2^R)$. Notice now that by Lemma 4.2 and Lemma 4.1 it holds that $u_1(c_1', L_2^R) \leq u_1(c_1, L_2^R)$, if $c_1' < c_1$ or $c_1' \geq L_2^R$. Hence, $c_1 < c_1' < L_2^R$. But then $u_1(c_1', c_2) > u_1(c_1, c_2)$, again by Lemma 4.1. This, however, leads to a contradiction, since (c_1, c_2) had been assumed to be a NE.

Case 2.5 $c_1 < \mathbf{m}_R < c_2$**.** Reasoning is symmetric to case 2.4.

□

We can now show that checking if a NE exists can be done in linear time. We first show the following structural lemma.

Lemma 4.5. *Take a line* $[0, k]$*, distribution of voters* $f : [0, k] \to \mathbb{N}_0$*, set of parties* P*,* $c_1 \in P_1$*, and* $c_2 \in P_2$*. Then, there is a best response* c_2 *to* c_1*, such that one of the following holds: (1)* $c_2 = c_1$*, (2)* $c_1 < c_2$ *and for every* $c_2' \in P_2$*, such that* $c_2' > c_1$*,* $|c_2' - c_1| > |c_2 - c_1|$*, or (3)* $c_1 > c_2$ *and for every* $c_2' \in P_2$*, such that* $c_1 > c_2'$*, we have that* $|c_1 - c_2'| > |c_1 - c_2|$*.*

Proof. Follows as a consequence of Lemma 4.1. □

By Lemma 4.5 it holds that, given a choice of one of two parties, we only need to check three choices of the second to find its best response. This, and the previously shown facts, allows us to provide a procedure for checking if a NE exists and, if it does, to construct a profile witnessing it.

Theorem 4.1. *If only two parties are present, then checking if a NE exists is linear-time solvable.*

Proof. Consider the line $[0, k]$, distribution of voters $f : [0, k] \to \mathbb{N}_0$, and two parties P_1, P_2. Then, compute the sets of most central candidates of P_1 and P_2, namely C_1 and C_2. Notice that we can do it in linear time, having computed the median positions, which is also possible in linear time. Also, by Lemma 4.4, we know that if there is a NE in the game we consider, then there is a profile $\mathbf{c} = (c_1, c_2)$ which is a NE, while $c_1 \in C_1$ or $c_2 \in C_2$.

Given an i in C_1, let $S(i)$ be the set of candidates in P_2, such that for every $c_2 \in S(i)$ either (1) $c_2 = i$, (2) $i < c_2$ and for every $c_2' \in P_2$, such that $c_2' > i$, it holds that $c_2' - i > c_2 - i$, or (3) $i > c_2$, while for every $c_2' \in P_2$, such that $i > c_2$, we have that $i - c_2' > i - c_2$. Notice that $\#S(i) \leqslant 3$, and that by Lemma 4.5 there is a $c_2 \in \arg\max_{p \in P_2} u_2(i, p)$, such that $c_2 \in S(i)$. Then, we can compute $\arg\max_{p \in P_2} u_2(i, p)$

by examining at most three strategy profiles. We can now check symmetrically, if for some $c_2 \in \arg\max_{p \in P_2} u_2(i,p)$, it holds that $i \in \arg\max_{p \in P_1} u_1(p, c_2)$. If yes, we found a NE. Repeat this procedure for all i in $C_1 \cup C_2$. As $\#C_1 + \#C_2$ is bounded by 8, we can find a NE, if it exists, by examining at most 24 strategy profiles. This is because, by Lemma 4.4, and by Lemma 4.5, we know that if there is at least NE in a given game, then in some of them one of most central candidates is chosen, and the location of their opponent is limited to one of 3 positions. Notice that then the computation terminates in linear time. So, our algorithm computes the sets of most central candidates and subsequently checks, for each member of these sets, whether they can be extended to a NE profile. Notice that its correctness relies on Lemma 4.4 and Lemma 4.5. □

4.3 Games with Many Parties

We first observe that there are instances without NE for games with an arbitrary number of parties present, even when the setting is restricted to a uniform distribution of voters. This constitutes a major difference between the studied framework and the classical Hotelling-Downs model, where, for some numbers of players, the existence of NE is guaranteed (see, e.g., Eaton and Lipsey [1975]). In that framework, for instance, a NE equilibrium exists with four agents competing, but not with three.

Proposition 4.3. *For every $n \geqslant 2$, there is a game with the uniform distribution of voters and n parties, that has no NE.*

Proof. Suppose that $n > 2$. Then, take the line [0,20] with the uniform distribution of voters, and a set of parties $P = \{P_1, P_2, \ldots, P_n\}$. Then, let $P_1 = \{4, 7\}$, $P_2 = \{6, 8\}$. Also, for every P_i such that $i > 2$ and $i \leqslant n$, let $P_i = \{5\}$, as depicted in Figure 4.5.

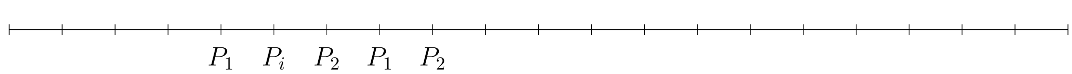

Figure 4.5: Positions of candidates of all parties in the game used in the proof of Proposition 4.3, which are indicated below the line. The position specified P_i corresponds to the candidate of any party other than P_1 or P_2.

Note that in all strategy profiles parties other than P_1 and P_2 select 5. Further, consider utilities of parties P_1 and P_2 in all strategy profiles. The utilities of these parties are shown in Figure 4.6.

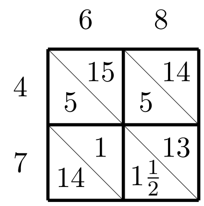

Figure 4.6: Representation of the game used in the proof of Proposition 4.3.

It is routine to check that this game has no NE. Finally, notice that by Proposition 4.2 there are instances without NE also for the two party case. □

Interestingly, there are cases without Nash equilibria even if there is a party that is guaranteed to get the majority of votes. This would be impossible, however, if parties were only concerned with winning the elections (i.e., receiving more votes than other competitors), rather than with attracting as many voters as possible.

Proposition 4.4. *There are games with no Nash equilibria, such that for some party P_i and for every strategy profile $\mathbf{c}$ it holds that $u_i(\mathbf{c}) > \frac{V(f)}{2}$.*

Example 4.2. *Now we show that Proposition 4.4 holds, by providing an example of a game, in which there is no NE, while one of the parties is guaranteed to attract the strict majority of voters. Consider the elections with voters uniformly distributed on the line $[0, 100]$. Also, consider parties $P_1 = \{70\}, P_2 = \{73, 89\}, P_3 = \{88, 90\}, P_4 = \{88, 90\}, P_5 = \{75\}, P_6 = \{100\}$. Notice that, by construction, under every strategy profile in this game, P_1 receives at least 70 out of 101 votes. We will now demonstrate that there is no NE in this game. As actions of P_1, P_5 and P_6 are fixed, we focus on the utilities of parties P_2, P_3 and P_4. Table 1 gives the utilities of these parties in all strategy profiles.*

	(73,88,88)	(89,88,88)	(89,88,90)	(73,88,90)	(89,90,88)	(73,90,88)	(73,90,90)	(89,90,90)
u_2	$2\frac{1}{2}$	6	1	$2\frac{1}{2}$	1	$2\frac{1}{2}$	$2\frac{1}{2}$	$7\frac{1}{2}$
u_3	$6\frac{1}{4}$	$3\frac{1}{2}$	7	$7\frac{1}{2}$	$5\frac{1}{2}$	6	$6\frac{1}{4}$	$2\frac{3}{4}$
u_4	$6\frac{1}{4}$	$3\frac{1}{2}$	$5\frac{1}{2}$	6	7	$7\frac{1}{2}$	$6\frac{1}{4}$	$2\frac{3}{4}$

Table 4.1: Utillity of parties P_2, P_3, P_4 in all strategy profiles for the game used in Example 4.2. Given a profile (c_2, c_3, c_4), c_2 is the choice of party P_2, c_3 is the choice of P_3 and c_4 is the choice of P_4.

It can then be checked that there is no NE in this game. Below we present which party has a profitable deviation for every strategy profile.

- $(73, 88, 88) \rightarrow_{P_2} (89, 88, 88)$
- $(89, 88, 88) \rightarrow_{P_4} (89, 88, 90)$
- $(89, 88, 90) \rightarrow_{P_2} (73, 88, 90)$
- $(73, 88, 90) \rightarrow_{P_4} (73, 88, 88)$
- $(89, 90, 88) \rightarrow_{P_2} (73, 90, 88)$
- $(73, 90, 88) \rightarrow_{P_3} (73, 88, 88)$
- $(73, 90, 90) \rightarrow_{P_2} (89, 90, 90)$
- $(89, 90, 90) \rightarrow_{P_4} (89, 90, 88)$

We will also consider a natural class of games, where parties' candidates are located within non-overlapping intervals (which we also refer to as parties' *sectors*), which we call a *sector structure*. Intuitively, this is a scenario in which each two parties form disjoint intervals. Formally, for a line $[0, k]$ with a set of parties $P = \{P_1, \dots, P_n\}$, we say that P has the *sector structure* if $i \neq j$ implies that $[\min(P_i), \max(P_i)] \cap [\min(P_j), \max(P_j)] = \emptyset$. Without loss of generality, we will assume that given a set of parties $P = \{P_1, \dots, P_n\}$ with a sector structure and parties P_i, P_j, such that $i > j$, we have that $c_i > c_j$ for each $c_i \in P_i$ and $c_j \in P_j$. So, $P_1, \dots, P_n$ encodes the strict ordering of parties on a line. Figure 4.7 shows an example of a set of parties with the sector structure.

Example 4.3. *Consider sectors of parties $P_1 = \{0\}, P_2 = \{1\}, P_3 = \{3, 4\}$ on the line $[0, 4]$, where voters are uniformly distributed. This example is depicted in Figure 4.7. Here, a sector of P_1 is $[0, 0]$ (marked as S_1), a sector of P_2 is $[1, 1]$ (marked as S_2), and a sector of P_3 is $[3, 4]$ (marked as S_3).*

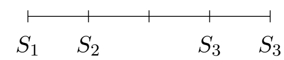

Figure 4.7: Example game with the set of parties with the sector structure and a uniform distribution of voters. Below the line we specify the candidates of all of the parties.

It turns out that there are instances without NE, even if parties have the sector structure.

Proposition 4.5. *There are games where the set of parties P has the sector structure, but which have no Nash equilibria.*

Example 4.4. *To show that Proposition 4.5 holds, let us provide an example of a game in which no strategy profile is a NE, while the set of parties has the sector structure. Take a line* $[0, 36]$ *and distribution of voters such that* $f(11) = 5$*,* $f(21) = 6$*,* $f(35) = 3$*, and for every* $i \in [0, 36] \setminus \{11, 21, 35\}$*, we have that* $f(i) = 0$*. Also, let* $P = \{P_1, P_2, P_3, P_4\}$*, with* $P_1 = \{15, 20\}$*,* $P_2 = \{25, 35\}$*,* $P_3 = \{5\}$ *and* $P_4 = \{36\}$*. Notice that* P *has the sector structure (the sector of* P_1 *is* $[15, 20]$*, of* P_2 *is* $[25, 35]$*, of* P_3 *is* $[5, 5]$*, and of* P_4 *is* $[36, 36]$*).*

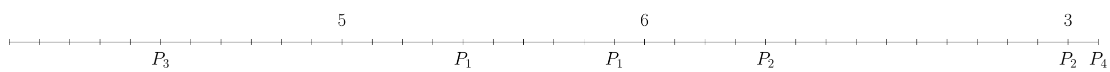

Figure 4.8: Location of voters and positions of candidates of all parties in the game used in Example 4.5. Numbers above the line indicate the number of voters at each position, while positions of particular parties are indicated below the line.

The representation of the strategic game between P_1 *and* P_2 *is depicted in Figure 4.9.*

	25	35
15	5 \ 6	11 \ 3
20	6 \ 0	6 \ 3

Figure 4.9: Representation of the game in Example 4.5

It is routine to check that this game has no NE, as P_3 *and* P_4 *have only one action available.*

Interestingly, this observation does not hold if the distribution of voters is uniform. This is due to the fact that if a nominee of a party is located between two neighbours, then selecting an alternative, which is also situated between these neighbours, is not profitable. This implies the existence of a NE, as in games with the sector structure we have that for every party P_i, and every strategy profile, the only possible deviation of P_i is to a candidate with the same neighbours, as their initial nominee.

Given a strategy profile $\mathbf{c}$ and a uniform distribution of voters, candidates $c_i \in P_i$ in $\mathbf{c}$ and $c'_i \in P_i$, we say that c_i and c'_i *have the same neighbourhood* if the following conditions hold: (1) both $L(c_i)$ and $R(c_i)$ exist, (2) $L(c_i) < c'_i < R(c_i)$, and (3) c_i and c'_i do not share their position with any other party's candidate in $\mathbf{c}$. So, $c_i \neq c_j$ and $c'_i \neq c_j$, for all c_j in $\mathbf{c}$ with $j \neq i$. Notice that this notion is only defined for candidates which are not leftmost or rightmost in a strategy profile. An instance of a game, in which two candidates have the same neighbourhood, is shown in Figure 4.10.

Example 4.5. *Consider the game in Figure 4.10. This is an example of a game where p_2^1 and p_2^2 have the same neighbourhood.*

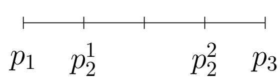

Figure 4.10: A game with the line of uniformly distributed voters of length 4 and parties $P_1 = \{0\}$, $P_2 = \{1, 3\}$ and $P_3 = \{4\}$. Observe that p_1^1 and p_1^2 have the same neighbourhood, as they are both located between p_1 and p_2, which is not the case for any member of a party different than P_1.

We further state the following lemma. While it is not difficult to see that it holds, we include its proof, as it will be crucial in our further arguments.

Lemma 4.6. *When voters are uniformly distributed, then, for every strategy profile $\mathbf{c}$ and a party P_i, with $c_i, c'_i \in P_i$, we have that, if c_i and c'_i have the same neighbourhood, then $u_i(\mathbf{c}) = u_i(c'_i, \mathbf{c}_{-i})$.*

Proof. Take a strategy profile $\mathbf{c}$, party P_i, and a candidate $c'_i \in P_i$, such that $c'_i > L(c_i)$ and $c'_i < R(c_i)$. Without loss of generality, let $c_i < c'_i$. Then notice that from the definition of utilities, we get that $u_i(\mathbf{c}) = \frac{R(c_i)-c_i}{2} + \frac{c_i-L(c_i)}{2} = \frac{R(c_i)-L(c_i)}{2}$. But then, by a symmetric calculation, it holds that $u_i(c'_i, \mathbf{c}_{-i}) = \frac{R(c_i)-c'_i}{2} + \frac{c'_i-c_i}{2} + \frac{c_i-L(c_i)}{2} = \frac{R(c_i)-c_i}{2} + \frac{c_i-L(c_i)}{2}$. Hence, $u_i(\mathbf{c}) = u_i(c'_i, \mathbf{c}_{-i}) = \frac{R(c_i)-L(c_i)}{2}$. □

This leads us to the existence of NE in games with parties with the sector structure and uniform distribution of voters.

Proposition 4.6. *For every line* $[0, k]$*, distribution of voters* $f : [0, k] \to \mathbb{N}_0$*, and a set of parties* $P = \{P_1, \ldots, P_n\}$*, if* f *is uniform and* P *has the sector structure, then there exists a NE.*

Proof. Take a line $[0, k]$ with the uniform distribution of voters, and a set of parties $P = \{P_1, \ldots, P_n\}$ with the sector structure. Consider any strategy profile $\mathbf{c}$, such that $c_1 = \max(P_1)$ and $c_n = \min(P_n)$. Notice that, as c_1 is the leftmost position in $\mathbf{c}$, and c_n is the rightmost position in $\mathbf{c}$, by Lemma 4.1, we have that P_1 and P_n cannot improve their utilities unilaterally. But also, as P has the sector structure, for every other party P_i and a pair $c_i, c_i' \in P_i$, it holds that c_i and c_i' have the same neighbourhood. So, by Lemma 4.6, no other party $P_i \in P$ can improve their utility. Hence, $\mathbf{c}$ is a NE. □

Given that there are instances without NE, even in elections with a relatively simple structure, it is natural to study the complexity of checking whether there is a NE in a given game. Observe that the problem is solvable in polynomial time, when the number of parties is bounded by a constant. Indeed, if there are k parties in an electoral competition, then the number of all possible strategy profiles is upper-bounded by m^k, where m is the size of the largest party. Hence, we can check the existence of an equilibrium in polynomial time by determining, for each of those profiles, if some party can deviate profitably. However, we find that the general case is **NP**-complete by reduction from 3-SAT.

NE-existence:
Input: Line $[0, k]$, set of parties $P = \{P_1, \ldots, P_n\}$, distribution function $f \colon [0, k] \to \mathbb{N}_0$.
Question: Is there a strategy profile in the game, given $[0, k]$, P, and f, which is a NE?

In our reduction we construct, for a formula φ in 3-CNF, an instance $\mathcal{I}$ of NE-existence (which is a game), such that there exists a NE in $\mathcal{I}$ if and only if φ is satisfiable. The line in our instance is composed of *variable* and *clause* segments, one for each variable and each clause, respectively. Then, for each variable x_i, we construct a *variable party*, with two candidates (corresponding to x_i and $\neg x_i$), which are located in the variable segment for x_i. Hence, every strategy profile in the game we construct corresponds to some valuation V over the set of variables. Further, for each clause C_j, we construct a corresponding party, with a candidate in each segment corresponding to a literal occurring in C_i, and two candidates in the clause segment for C_i. By creating a singleton party located in each variable segment, we get that the party corresponding to C_i, which chooses a candidate in a variable segment corresponding to a literal L in C_i, obtains a strictly positive utility only if L is true in V. Further, we construct, for each clause segment, an additional party, with candidates located in that segment. This ensures that a profile, in which a clause party chooses a nominee in the clause segment, is not a NE. This allows us to show, using a "potential" argument, that there is a NE in $\mathcal{I}$ if and only if φ is satisfiable.

Theorem 4.2. NE-EXISTENCE *is* **NP***-complete, even if the party size is bounded by a constant not smaller than* 5*.*

Proof. Notice first that the problem we consider is in **NP**. Indeed, given a strategy profile $\mathbf{c} = (c_1, \ldots, c_n)$, we can check in polynomial time whether $u_i(\mathbf{c}) \geq u_i(c'_i, \mathbf{c}_{-i})$, for every party P_i and every candidate $c'_i \in P_i$.

We prove **NP**-hardness by a reduction from 3-SAT. Take a 3-CNF formula φ, with the set $C = \{C_0, \ldots, C_m\}$ of clauses. Without loss of generality, we assume that for each $0 \leq k < \#C$, clause C_k is given by a set of three distinct literals $\{L_0^k, L_1^k, L_2^k\}$ over a set of variables X. Let C'_k denote a copy of C_k and x'_i a copy of x_i. We may assume that the literals are defined over a set of variables $X = \{x_0, \ldots, x_n\}$, and, without loss of generality, also that in every clause C_j at most one of the literals x or $\neg x$ occurs at most once. We may also assume that for every variable x, both literals x and $\neg x$ occur in φ.

Further, we construct the game on the line $[0, 9(\#X + \#C) - 1]$, which we can conveniently think of as being composed of $\#X + \#C$ segments of length 9. Figures 4.11 and 4.12 illustrate our construction. Thus, for each variable x_i ($0 \leq i < \#X$) we construct a *variable segment* $[9i, 9i{+}8]$, and for each clause C_k ($0 \leq k < \#C$) a *clause segment* $[9(\#X{+}k), 9(\#X{+}k){+}8]$. Hence, all positions $n < 9\#X$ are located in variable segment, whereas all positions $n \geq 9\#X$ lie in a clause segment.

Furthermore, we define the distribution function f, such that, for every $0 \leq k < 9(\#X + \#C)$

$$f(k) = \begin{cases} 6\#C & \text{if} \quad k < 9\#X \quad \text{and} \quad k \bmod 9 \in \{3, 5\}, \\ 1 & \text{if} \quad k \geq 9\#X \quad \text{and} \quad k \bmod 9 \in \{2, 3, 5, 6\}, \\ 0 & \text{otherwise} \end{cases}$$

As parties, we have, for every variable x_i ($0 \leq i < \#X$), and for every clause C_k and its copy C'_k ($0 \leq k < \#C$) that

$$\begin{aligned} P_{x_i} &= \{9i+3, 9i+5\} \\ P_{x'_i} &= \{9i+4\} \\ P_{C_k} &= \{9i+6\colon x_i \in C_k\} \cup \{9i+2\colon \neg x_i \in C_k\} \cup \\ &\quad\ \{9(\#X+k)+2, 9(\#X+k)+6\} \\ P_{C'_k} &= \{9(\#X+k)+3, 9(\#X+k)+5\} \end{aligned}$$

Observe now that the distribution function has been chosen in such a way that a party can only attract voters from the segment, within which its nominee is positioned. Also notice that the size of each party P_{x_i} and each party $P_{C'_k}$ is 2, whereas the size of each party P_{C_k} is 5. Variable segments are presented in Figure 4.11.

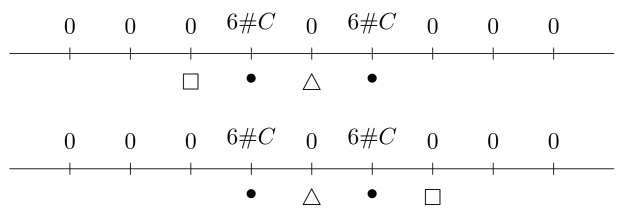

Figure 4.11: Variable segments $[9i, 9i+8]$ (above) and $[9j, 9j+8]$ (below), for a variables x_i and x_j, such that $\neg x_i \in C_k$ and $x_j \in C_k$. Variable parties P_{x_i} and P_{x_j} are indicated by the bullets in respectively the top and bottom segment. Choosing the left candidate corresponds to setting variable x_i, respectively x_j, to true, and choosing the right candidate corresponds to setting variable x_i, respectively x_j, to false. The clause party P_{C_k} has candidates at the positions indicated by the boxes. If neither x_m nor $\neg x_m$ occurs in C_k, then party P_{C_k} has no candidates in segment $[9m, 9m+8]$. The triangles denotes the solitary candidates of parties $P_{x'_i}$ and $P_{x'_j}$.

Further, a clause segment is illustrated in Figure 4.12.

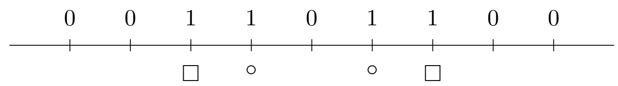

Figure 4.12: Clause segment $[9(\#X+k), 9(\#X+k)+8]$, for a clause C_k. Party P_{C_k} has candidates at the locations indicated by the boxes, but has no candidates in any other clause segments. Party $P_{C'_k}$ has two candidates at the locations indicated by the circles.

We show now that this game has a NE if and only if φ is satisfiable. First assume that φ is satisfiable and let V be a satisfying assignment over X. That is, V satisfies at least one literal in each clause. Given assignment V, we consider profiles $\mathbf{c} = (c_{x_0}, \ldots, c_{C'_K})$, which we will refer to as *proto-equilibria*. They are such that for every variable x_i, with $0 \leq i < \#X$, it holds that

$$c_{x_i} = \begin{cases} 9i+3 & \text{if } x_i \text{ is true in } V \\ 9i+5 & \text{if } x_i \text{ is false in } V \end{cases}$$

Moreover, for a profile to qualify as a proto-equilibrium, for every clause C_k there has to be some literal L in C_k that is satisfied by V, such that

$$c_{C_k} = \begin{cases} 9j+6 & \text{if } x_i \text{ is true in } V \text{ and } L = x_j \\ 9j+2 & \text{if } x_i \text{ is false in } V \text{ and } L = \neg x_j \end{cases}$$

We furthermore require that $c_{C'_k} = 9(\#X+k)+3$, for all $0 \leq k < \#C$. Notice that then $c_{x'_i} = 9i+4$, for $0 \leq i < \#X$.

By means of the following potential argument, we now show that among the proto-equilibria for V, there must be at least one NE. Towards this end, let $\lambda_i^{\mathbf{c}}$, for each proto-NE $\mathbf{c}$ for V, and for each $0 \leq i < \#X$, be the number of clause parties that choose their nominee from the variable segment $[9i, 9i+8]$ under $\mathbf{c}$. So

$$\lambda_i^{\mathbf{c}} = \#\{C_k \in C \colon c_{C_k} \in [9i, 9i+8]\}$$

Let now $\lambda^{\mathbf{c}} = (\lambda_{i_0}^{\mathbf{c}}, \ldots, \lambda_{i_{\#X-1}}^{\mathbf{c}})$ be a sequence of the values $\lambda_0^{\mathbf{c}}, \ldots, \lambda_{\#X-1}^{\mathbf{c}}$, ordered in a non-decreasing order. We will argue that every proto-NE $\mathbf{c}$, for which the sequence $\lambda^{\mathbf{c}}$ is *lexicographically maximal*, is also a NE. Here, we use the lexicographic order with respect to the standard relation $\leq$ on the integers. For instance, $(0,1,3,4,7,9)$ is lexicographically greater than $(0,1,2,7,8,8)$.

Further, let $\mathbf{c}^*$ be a proto-equilibrium, for which $\lambda^{\mathbf{c}^*}$ is lexicographically maximal. Then, for every variable party P_{x_i}, it holds that $u_{x_i}(9i+3, \mathbf{c}^*_{-x_i}) = u_{x_i}(9i+5, \mathbf{c}^*_{-x_i}) = 6\#C$, and it follows that P_{x_i} does not want to deviate from $\mathbf{c}^*$. We further observe that the singleton parties $P_{x'_i}$ $(0 \leq i < \#X)$ cannot profitably deviate from $\mathbf{c}^*$ either.

Moreover, for every party $P_{C'_k}$, we have that

$$u_{C'_k}(9(\#X+k)+3, \mathbf{c}^*_{-C'_k}) = u_{C'_k}(9(\#X+k)+5, \mathbf{c}^*_{-C'_k}) = 4$$

because $\mathbf{c}^*_{C_k} \notin [9(\#X+k), 9(\#X+k)+9]$. Therefore, $P_{C'_k}$ does not want to deviate from $\mathbf{c}^*$ either.

Now, consider an arbitrary clause party P_{C_k}. As $\mathbf{c}^*$ is a proto-equilibrium, we note that there is some $0 \leq i < \#X$, such that $c_{C_k} = 9i+2$, if $c_{x_i} = 9i+5$, and $c_{C_k} = 9i+6$, if $c_{x_i} = 9i+3$. In either case, $u_{C_k}(\mathbf{c}^*) = \frac{6\#C}{\lambda_i^{\mathbf{c}^*}}$. As $\lambda_i^{\mathbf{c}^*} \leq \#C$, it follows that $u_{C_k}(\mathbf{c}^*) \geq 6$. Observe now that if P_{C_k} were to deviate and choose its nominee in another variable segment $[9j, 9j+8]$, such that either $c'_{C_k} = 9j+2$ and $\mathbf{c}^*_{x_j} = 9j+3$, or $c'_{C_k} = 9j+6$ and $\mathbf{c}^*_{x_j} = 9j+5$, then $u_{C_k}(\mathbf{c}^*) = 0$. Moreover, if P_{C_k} were to deviate to a position in clause segment $[9(\#X+k), 9(\#X+k)+8]$, then both $u_{C_k}(9(\#X+k)+2, \mathbf{c}^*_{-C_k}) \leq 2$ and $u_{C_k}(9(\#X+k)+6, \mathbf{c}^*_{-C_k}) \leq 2$. Again, party P_{C_k} does not want to deviate from $\mathbf{c}^*$. Finally, assume towards contradiction that P_{C_k} would profit from deviating to a position c'_{C_k} in a variable segment $[9j, 9j+8]$, with $0 \leq j \leq \#X$ different from $[9i, 9i+8]$, such that either $c'_{C_k} = 9j+2$ and $\mathbf{c}_{x_j} = 9j+5$, or $c'_{C_k} = 9j+6$ and $\mathbf{c}^*_{x_j} = 9j+3$. Let $\mathbf{c}^{**} = (c'_{C_k}, \mathbf{c}_{-C_k})$. Notice that $\mathbf{c}^{**}$ is a proto-equilibrium. Moreover, $u_{C_k}(\mathbf{c}^{**}) > u_{C_k}(\mathbf{c}^*)$, i.e., $\frac{6\#C}{\lambda_j^{\mathbf{c}^{**}}} > \frac{6\#C}{\lambda_j^{\mathbf{c}^*}}$. Hence, $\lambda_j^{\mathbf{c}^{**}} < \lambda_i^{\mathbf{c}^*}$. Observing that $\lambda_i^{\mathbf{c}^{**}} = \lambda_i^{\mathbf{c}^*} - 1$, and that $\lambda_j^{\mathbf{c}^{**}} = \lambda_j^{\mathbf{c}^*} + 1$, we find that $\lambda_j^{\mathbf{c}^*} < \lambda_i^{\mathbf{c}^*}$, $\lambda_j^{\mathbf{c}^{**}} \leq \lambda_i^{\mathbf{c}^{**}}$, and that $\lambda_k^{\mathbf{c}^*} = \lambda_k^{\mathbf{c}^{**}}$, for all $k \neq i, j$. It follows that $\lambda^{c^{**}}$ is lexicographically greater than λ^{c^*}, which contradicts the assumptions.

For the opposite direction, assume that φ is not satisfiable. Consider an arbitrary profile $\mathbf{c} = (c_{x_0}, \ldots, c_{C'_K})$, and assume towards contradiction that $\mathbf{c}$ is a NE. Let $V_{\mathbf{c}}$ be the assignment, such that for every $0 \leq i < \#X$, it holds that x_i is true in $V_{\mathbf{c}}$ if $c_{x_i} = 9i+3$, and that x_i is false if $c_{x_i} = 9i+5$.

Then, there is some clause C_k, with $0 \leq k < \#C$, such that $V_{\mathbf{c}}$ evaluates every literal in C_k to false. Accordingly, if c_{C_k} is in a variable segment $[9i, 9i+8]$ with $0 \leq i < \#X$, then either both $c_{C_k} = 9i+2$ and $c_{x_i} = 9i+3$, or both $c_{C_k} = 9i+6$ and $c_{x_i} = 9i+5$. In either case $u_{C_k}(\mathbf{c}) = 0$. Now, consider $d_{C_k} = 9(\#X+k)+2$. Then, $u_{C_k}(d_{C_k}, \mathbf{c}_{-C_k}) \geq 1$. Hence, $\mathbf{c}$ is not a NE, which contradicts the assumptions.

To conclude, assume that c_{C_k} is in the segment $[9(\#X+k), 9(\#X+k)+8]$. Observe that if $c_{C_k} = 9(\#X+k)+2$ and $c_{C'_k} = 9(\#X+k)+3$, then party P_{C_k} would deviate to $d_{C_k} = 9(\#X + k) + 6$. If $c_{C_k} = 9(\#X + k) + 6$ and $c_{C'_k} = 9(\#X + k) + 3$, party $P_{C'_k}$ deviates to $d_{C'_k} = 9(\#X + k) + 5$, and, if $c_{C_k} = 9(\#X + k) + 6$ and $c_{C'_k} = 9(\#X + k) + 5$, then party P_{C_k} deviates to $d_{C_k} = 9(\#X + k) + 2$. Finally, if $c_{C'_k} = 9(\#X + k) + 5$ and $c_{C_k} = 9(\#X + k) + 2$, then party $P_{C'_k}$ deviates to $d_{C'_k} = 9(\#X + k) + 3$.

To illustrate that let us depict the utilities of parties P_{C_k} and $P_{C'_k}$ in this scenario in Figure 4.13. There, $c^1_{C_k} = 9(\#X + k) + 2$, $c^2_{C_k} = 9(\#X + k) + 6$, $c^1_{C'_k} = 9(\#X + k) + 3$, and $c^2_{C'_k} = 9(\#X + k) + 5$. It is now routine to check that there is no NE under these circumstances.

	$c^1_{C'_k}$	$c^2_{C'_k}$
$c^1_{C_k}$	1 \ 3	2 \ 2
$c^2_{C_k}$	2 \ 2	1 \ 3

Figure 4.13: Representation of the utilities of P_{C_k} and $P_{C'_k}$.

It follows that **c** is not a NE, which contradicts the assumptions. We conclude that the game does not allow for any NE. □

4.4 Conclusion

In this chapter we studied an extension of the Hotelling-Downs model, where political parties compete for voters located on a left-to-right political spectrum, by selecting a nominee within their pool of potential nominees, who in turn have fixed political stances and attract the closer voters. Observe that in this approach the parties' goals exceed winning the elections. Notice that even though we assume that voters are located on a discrete line, our framework can be directly generalised in various ways, for example to scenarios where finitely many voters are placed on real intervals, preserving utilities.

Summary of Contributions. Our results indicate that predicting nominee selection can be a computationally hard problem. In particular, we have shown games without NE even with two parties (Proposition 4.2). Also, we have established that NE computation is **NP**-complete with more than two parties (Theorem 4.2). However, computing NE becomes easy in two-party systems (Theorem 4.1).

Future Research. Our contribution suggests a number of directions for future research. Let us name a few particularly interesting problems.

- In this chapter we limited ourselves to checking the existence of an equilibrium state. It is interesting, however, to study the properties of Nash equilibria in the current context. For instance, it would be natural to explore the price of anarchy in the games we considered.

- Even though the **NP**-hardness of NE existence shows the difficulty of this problem in the general case, it is natural to study classes of elections, more complex than those explored in this chapter, in which this is tractable.

- Even if restricting ourselves to a finite number of positions, the framework can be directly generalised in various ways, for example to scenarios where finitely many voters are placed on real intervals, preserving utilities.

- Establishing the parametrised complexity of checking the existence of NE is an important follow up. We saw that if the number of parties is fixed at a constant, then the problem can actually be solved in polynomial time. Indeed, if the line is given by $[0, k]$ and the number of parties is n, then one can enumerate the at most k^n possible strategies, and for each one of them, check whether it is a NE in polynomial time. This corresponds to an **XP** algorithm parameterised by the number of parties. However, obtaining a fixed-parameter algorithm parameterised by the number of parties, i.e., an $f(n)k^c$ time algorithm, where c is a constant independent of k and n, appears to be a challenging problem.

- In this chapter, we only considered games played on a line. It is natural, however, to consider electoral competitions defined on different spaces. A particularly interesting extension is the one, in which parties choose between candidates located in different dimensions, corresponding, e.g., to opinions on several issues.

- Exploration of possible strategic behaviour from the side of voters would link the results of this chapter to the classical approach in the study of strategic behaviour in voting.

- Another interesting direction involves the modelling assumptions, starting with the role of information, e.g., taking into account the uncertainty of voters participating in the election.

- We assumed that each party only selects one candidate, leaving open the problem of parties choosing sets of candidates instead. Moreover, it might be useful to also consider the case, in which parties can form coalitions, i.e., they decide to select a common candidate and share the joint payoff. The development of heuristics for these more complex problems is also a natural follow up.

- It would also be worthwhile to study solution concepts other than NE, such as *dominant strategy equilibria* (DSE). This concept is especially interesting from the perspective of predicting parties' actions. Finding a dominant strategy for a given party strongly suggests their choice, regardless of other parties' selections. We believe that in the setting studied in the current paper checking the existence of a DSE is algorithmically easier than verifying the existence of a NE.

- In this chapter we only studied the existence of some equilibrium profile. From the perspective of predicting parties' choices it would be interesting, however, to study whether a game admits more than one NE.

Chapter 5

Nominee Selection in Knockout Tournaments

5.1 Introduction

When a winner is to be selected from a set of players, it is natural to base the decision on *pairwise comparisons* between them, which allows for a neat representation of the extent to which they are supported. Apart from the natural application of tournaments, e.g., in sports, in voting this comparison can be determined by checking which of the players in a pair is preferred to the other by the majority of voters. To account for that, *tournaments*, understood as directed graphs over players (see, e.g. Moon [1968], Laffond et al. [1993], Fisher and Ryan [1995], Laslier [1997]), were introduced in the game theory and social choice research, and have gained attention in computer science for their well-behaved computational properties (see, e.g., Brandt et al. [2016b]). From the perspective of strategic choices made by groups, which are relevant, e.g., for primaries, it is natural to study tournaments played by group (or party) nominees.

In this chapter we focus on what are possibly the simplest and best-known tournaments, i.e., knockout (or single-elimination) tournaments. There, players are initially associated with the leafs of a full binary tree, and the winner of the match played between the players at a pair of sibling vertices proceeds to the next stage, i.e., the parent of these two vertices.

Our Contribution. In this chapter we extend standard single-elimination tournaments, to account for the strategic behaviour of coalitions. Before the tournament starts, we allow each of them to make an independent choice of the best player to put forward. We study the equilibrium behaviour of coalitions in such tournaments from an algorithmic point of view. Our analysis spans three axes. A) Whether coalitions choose their players for the entire tournament. If that is the case, then we call such competitions *one-shot* tournaments. Otherwise, we refer to them as *dynamic* tournaments. B) Whether only winning matters for the coalitions. If so, we call such tournaments *win-lose*. On the contrary, if tournament progression is of importance to the coalition, then we call this case *beyond win-lose*. C) Whether we focus on computing equilibria, or on verifying a given one. Despite the complex tournament structure, we show polynomial time, or quasi-polynomial time algorithms for all of these cases. See Table 5.2 for an overview of our results in this chapter.

Structure of the Chapter. In Section 5.2, we analyse one-shot tournaments, providing an algorithm for computing their equilibria. Further, in Section 5.3, we focus on algorithmic aspects of dynamic tournaments. Finally, in Section 5.4, we conclude and provide directions for future research.

5.2 One-Shot Knockout Tournaments

In this section, we study tournaments, in which coalitions choose a representative once and for all, before the competition starts. We are interested in equilibrium strategies – specifically, pure Nash equilibria – and the complexity of their computation, as well as verification.

Starting from a set of coalitions $C = \{C_1, \ldots, C_n\}$, in the current context, a strategy profile is a tuple $\mathbf{c} = (c_1, \ldots, c_n)$, such that for each coalition C_i, we have that $c_i \in C_i$. Also, whenever the seeding π is clear from the context, we write $SE_{\mathbf{c}}$ to denote the tournament between players in $\mathbf{c}$ following π. Furthermore, we say that a coalition C_i *wins* a tournament $SE_{\mathbf{c}}$ if the winner of $SE_{\mathbf{c}}$ belongs to C_i.

5.2.1 Win-Lose Games

The simplest type of one-shot tournaments we look at, are the ones where only winning matters (i.e., *win-lose games*). When picking a player to put forward in a win-lose game, the goal of each coalition is to win the tournament, given the choices of their opponents. Let us define Nash equilibrium for such games.

Definition 5.1 (Nash equilibrium)**.** *A profile $\boldsymbol{c} = (c_1, \ldots, c_n)$ is a NE if for all i, and for all $c'_i \in C_i$ it holds that if C_i wins $SE_{(c'_i, \boldsymbol{c}_{-i})}$, then C_i wins $SE_{\boldsymbol{c}}$.*

It is worth noting that a NE does not need to exist, which opens an important algorithmic questions regarding their existence and computation. An example of a coalitional structure which does not admit a NE is depicted in Figure 5.1.

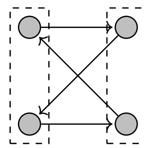

Figure 5.1: Coalitional structure without a NE. Observe that for every strategy profile in this coalitional structure, one of the coalitions can switch their representative to ensure that they win a tournament.

Structural and algorithmic properties of NE

We start with the analysis of Nash equilibria.

Equilibrium Existence. We start by addressing the question of deciding whether a given strategy profile is a NE, what we call the problem of *recognising* a NE. Let us show that computation of this problem is possible in sub-quadratic time.

Proposition 5.1. *Recognising a NE is solvable in polynomial time. Take a coalitional structure $T = (N, E, C)$. Further, let $C_i \in C$ be such that c_i is the winner of $SE_{\boldsymbol{c}}$.*

Proof. In the procedure we consider, for every coalition $C_j \in C$ such that $C_j \neq C_i$, and every player $c'_j \in C_j$, we check if C_j wins $SE_{(c'_j, \mathbf{c}_{-j})}$. Observe that if it is the case, then $\mathbf{c}$ is not a NE. Notice now that there is a set W_j of players in $\mathbf{c}$, with $\#W = \log \#C$, such that beating all members of W_j is necessary and sufficient for c'_j to win $SE_{(c'_j, \mathbf{c}_{-j})}$. We can therefore check if the condition we consider holds for c'_j in $\mathcal{O}(\log \#\mathcal{C})$ time. The algorithm for checking if $\mathbf{c}$ is a NE in $\mathcal{O}(\#N \log \#C)$ time follows. □

Equilibrium Computation. We are further interested in computing a NE if it exists, which is a more complex problem than recognising it. Surprisingly, we show that this can still be done in quasi-polynomial time.

More precisely, in the main result of this section, Theorem 5.1, we show, for a coalitional structure (N, E, C), the existence of an $\#N^{\mathcal{O}(\log \#C)}$-time algorithm for computing a NE. To obtain this result, we will make use of a key lemma, i.e., Lemma

5.2. There, we establish that a NE if one exists can be obtained by composing specific types of strategies for various subtournaments. This lemma effectively implies that we can compute a NE if one exists, by examining only at most $\#N^{\mathcal{O}(\log \#C)}$ out of the set of possibly $(\frac{\#N}{\#C})^{\#C}$ many strategy profiles.

Let us start with providing several useful notions.

Arborescences. We will often use the technical notion of *binomial arborescence*, following Vassilevska Williams [2010], which allows for a succinct formulation of the structural properties of SE-tournaments. This is because it captures the notion of a SE-tournament as a graph. An *arborescence* is a rooted directed tree, such that all the edges are directed away from the root. Intuitively, an edge between a pair of players captures which one of them won in the direct competition between them in a SE-tournament, with the fact that a vertex v is a root of a subarborescence meaning that it is a winner of a subtournament with the number of rounds equal to its degree.

Definition 5.2 (Vassilevska Williams [2010]). *Let $T = (N, E)$ be a tournament. The set of binomial arborescences over T is recursively defined as follows:*

- *Each $a \in N$ is a binomial arborescence rooted at a.*
- *If, for some $r > 0$, B_a and B_b are 2^{r-1}-vertex binomial arborescences, rooted at a and b respectively, then the tree B resulting from adding an edge from a to b is the 2^r-vertex binomial arborescence rooted at a.*

An example of binomial arborescence is shown in Figure 5.2. Observe how the relation in this arborescence corresponds to the structure of the SE-tournament depicted in Figure 3.3, with a_1 winning the tournament by beating b_2 in the first round and d_1 in the second. Also, we have that d_1 beats c_2 in the first round.

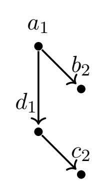

Figure 5.2: Example of a binomal arborescence with four players, a_1, b_2, d_1, and c_2. There, a_1 is the root of an 2^2-vertex binomial arborescence. Also, d_1 is the winner of a 2^1-vertex arborescence, while b_2 and c_2 are roots of 2^0-vertex arborescences.

If a binomial arborescence B is such that $V(B) = N$, then we say that B is a *spanning binomial arborescence* (s.b.a.) of $T = (N, E)$. Intuitively, a s.b.a. can be used to compactly encode how an SE-tournament will evolve, following the beating relation E. As shown by Vassilevska Williams [2010], there is a formal connection between binomial arborescences and knockout tournaments.

Proposition 5.2 (Vassilevska Williams [2010])**.** *Let (N, E) be a tournament, and let $v^\star \in N$. Then, there is a seeding of N, such that the resulting knockout tournament is won by $v^\star$ if and only if (N, E) has a s.b.a., rooted at $v^\star$.*

As a result of this proposition, we will interchangeably use the terms binomial arborescence and SE-tournament, when these are clear from the context. Regarding SE-tournaments, we will mainly work with binomial arborescences, as this allows for neater proofs and procedures.

We will further use the following notion.

Definition 5.3. *Let $T = (N, E, C)$ be a coalitional structure, $\mathbf{c}$ be a strategy profile, and let $C_i \in C$ be such that c_i is the root of the s.b.a. $SE_{\mathbf{c}}$. Further, take a coalition $C_j \in C$, and consider the c_i – c_j path H in $SE_{\mathbf{c}}$. Then, for every player p on this path, we denote by $\mathsf{Opp}_H[p, c_j]$ the set defined as follows:*

- *If $p = c_j$, then $\mathsf{Opp}_H[p, c_j]$ is the set of children of c_j in $SE_{\mathbf{c}}$.*
- *Otherwise, $\mathsf{Opp}_H[p, c_j]$ is the set $\{p\} \cup \{v : \ v \text{ is a child of } p \text{ and } v \succcurlyeq_p p'\}$, where p' denotes the unique child of p contained in H.*

The intuitive meaning of Definition 5.3 is that it formalises the set of future opponents, which the coalition C_j would have to face if it were to replace c_j with a different player $c_j' \in C_j$ in the profile $\mathbf{c}$, which we denote as $\bigcup_{p \in V(H)} \mathsf{Opp}_H[p, c_j]$. Note that this set is not larger than $\log \#C$, since that is the maximum number of opponents faced by any player in the tournament. We drop the explicit reference to H in this notation, when the root c_i (the winner) is clear from the context. See Example 5.1 for a visual illustration of Definition 5.3.

Example 5.1. *Here, we provide an example of a strategy profile which is not a NE, in which players* 12 *and* 33 *belong to the same coalition, which we call coalition 12, for the sake of convenience. Then, all of the other coalitions consist of one player only. We assume that the index of each of them is the same as the index of their unique player, e.g., player 14 plays for coalition 14. There are 33 players and 32 coalitions in total. The seeding pairs up coalition $2i + 1$ and coalition $2i + 2$ in the first round, for each $i \in [0, 15]$, and the beating relation corresponds to the arborescence shown in the figure. Consider now the path $1 \to 9 \to 11 \to 12$, which we call H. Notice that* $\mathsf{Opp}[12, 12] = \emptyset, \mathsf{Opp}[11, 12] = \{11\}, \mathsf{Opp}[9, 12] = \{9, 13\}$, *while* $\mathsf{Opp}[1, 12] = \{1, 17\}$*. Therefore, if coalition 12 wanted to win the tournament by replacing its chosen player, i.e., player 12 with an alternative, i.e., player 33, then it would have to beat the players chosen by coalitions* $\{11, 9, 13, 1, 17\}$*. Observe that these are precisely the coalitions indexed by* $\bigcup_{p \in V(H)} \mathsf{Opp}_H[p, 12]$*. Moreover, observe that fixing the choices of all coalitions, player* 33 *improves upon player* 12*, beating all the potential future opponents, namely players* $\{11, 9, 13, 1, 17\}$.

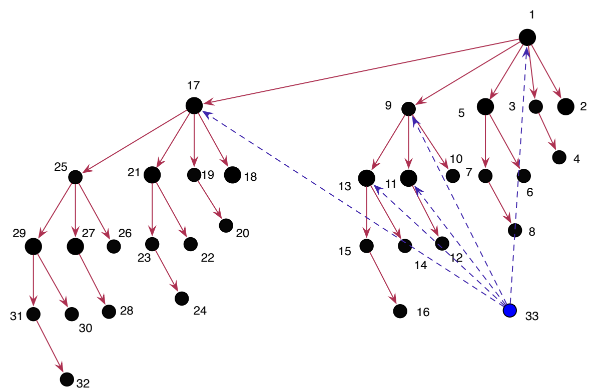

Figure 5.3: An example of a tournament, based on a strategy profile, which is not a NE.

The following structural lemma forms the crux of our algorithmic results. Informally, we use the fact that a profile is a NE if and only if no coalition C_i which is not a winner of the tournament can switch their representative c_i to a player c'_i, which beats all players beaten by c_i, as well as every future opponent, while the properties of the technical notion in Definition 5.3 are satisfied. Here, by a future opponent we mean a player that c_i would have faced if it won the tournament.

Lemma 5.1. *Consider a strategy profile* $\mathbf{c}$*, and let* c_i *be the root of the s.b.a.* $SE_{\mathbf{c}}$*. Then,* $\mathbf{c}$ *is a NE if and only if there is no* $C_j \in C$*, such that* $C_i \neq C_j$*, and a player* $c'_j \in C_j$*, such that* c'_j *beats all of the players in the set* $\mathsf{Opp}[p, c_j]$*, for every player* p *on the* $c_i - c_j$ *path in* $SE_{\mathbf{c}}$*.*

Proof. Take a coalitional structure (N, E, C), and a strategy profile $\mathbf{c}$. Let c_i denote the winner $SE_{\mathbf{c}}^{\log \# C}$. Then, consider a coalition C_j, such that $C_i \neq C_j$, while $c'_j \neq c_j$. Observe that c'_j is a winner of $SE_{(c'_j, \mathbf{c}_{-j})}$ if and only if c'_j beats all players $y \in \mathsf{Opp}_H[x, c_j]$, for each x on the $c_i - c_j$ path of $SE_{\mathbf{c}}$. The proof is based on the fact that for a $\mathbf{c}$ to be a NE, it is enough to check that for no coalition C_k with $c_i \notin C_k$, and for no $c'_k \neq c_k$, it holds that (1) c'_k does at least as well as c_k in $SE_{\mathbf{c}}$, and (2) c'_k beats every opponent on the path from c_k to c_i. $\square$

Figure 5.3 shows a strategy profile which is not a NE, based on the characterization in Lemma 5.1.

Furthermore, for a set S and $k \in \mathbb{N}$, we denote by $\binom{S}{\leqslant k}$ the set of subsets of S of size at most k. We further denote by $T_r(C_j)$ the set of coalitions, who could potentially meet coalition C_j within the first r-rounds, for some tournament structure with the set of coalitions C, and for some strategy profile $\mathbf{c}$, as well as coalition C_j itself. Notice that $\#T_r(C_j) = 2^r$. Thus, we have that $T_r(C_j) \setminus T_{r-1}(C_j)$ denotes the set of all possible opponent coalitions, which C_j could face exactly in the r^{th} round. For technical reasons, we set $T_r(C_j) = \emptyset$, for every $r \in \{-1, -2\}$, and every $C_j \in C$.

Let us exemplify this notion in the Figure 5.3. Consider coalition $\{1\}$ and $r = 3$. Also, select the unique strategy profile, in which 12 is chosen. Then, $T_r(\{1\})$ denotes the set of coalitions $\{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}, \{8\}\}$. Observe that this set corresponds to the tournament of size 2^3, in which $\{1\}$ participates.

Also, for a seeding π, we define by π_j^r the *restriction* of π to $T_r(C_j)$. Similarly, for a profile $\mathbf{c}$, we denote by $\mathbf{c}_j^r$, the profile $\mathbf{c}|_{T_r(C_j)}$, which is the restriction of $\mathbf{c}$ to $T_r(C_j)$. Conversely, we say that $\mathbf{c}$ is an *extension* of $\mathbf{c}_j^r$. Notice that there can be multiple possible extensions of $\mathbf{c}_j^r$.

Further, given the winner c_i of an arborescence $SE_{\mathbf{c}}$, and a player c_j, we say that c_j is the *final opponent* of c_i if c_j is the heaviest child of c_i in $SE_{\mathbf{c}}$. Also, for a set of coalitions C', we denote the set of their players, i.e, $\bigcup C'$, as $N(C')$.

We next define a mapping ζ, which allows us to reason about a NE in terms of the outcomes of specific subtournaments. Recall that as $\mathcal{C}$, we denote the set of strategy profiles in a given game.

Definition 5.4. *For a coalitional structure $T = (N, E, C)$, we define the function $\zeta : C \times N \times N \times [0, \log \#N] \times \binom{N}{\leqslant \log \#C} \to 2^{\mathcal{C}}$ as follows. Let $C_i, C_j \in C, r \in [0, \log \#C], c_i \in C_i$, and $c_j \in C_j$ be such that $C_j \in T_r(C_i) \setminus T_{r-1}(C_i)$, while it holds that $Z \in \binom{N}{\leqslant \log \#C}$ is such that $Z \cap N(T_r(C_i)) = \{c_i\}$. Then, $\zeta(C_i, c_i, c_j, r, Z)$ denotes the set of all strategy profiles $\mathbf{c}$ over $T_r(C_i)$ such that:*

(a) *player $c_i \in C_i$ wins $SE_{\pi_i^r, \mathbf{c}}$*

(b) *player c_j is the final opponent of c_i in the tournament $SE_{\pi_i^r, \mathbf{c}}$*

(c) *for every coalition $C_{j'} \in T_r(C_i) \setminus \{C_i\}$, and for every player $c_{j'} \in C_{j'}$, we either have that (1) $c_{j'}$ is beaten by a player in Z, or that (2) c_j' is beaten by a player in the set $\mathsf{Opp}[p, c_{j'}]$, for some player p on the $c_i - c_{j'}$ path in $SE_{\pi_i^r, \mathbf{c}}$*

For every other choice of C_i, c_i, c_j, r, Z, we set $\zeta(C_i, c_i, c_j, r, Z) = \emptyset$.

In other words, $\zeta(C_i, c_i, c_j, r, Z)$ is the set of all those profiles over the r-round tournament comprised of coalition C_i's first r matches, such that the player $c_i \in C_i$ wins these r rounds, player c_j is the final opponent of c_i in the r round tournament (i.e., c_i and c_j play in the r^{th} round), and all players belonging to a coalition $C_{j'}$, who could potentially meet C_i within the first r rounds, are beaten either by a player in the set Z, or by a player in the set $\mathsf{Opp}[p, c_{j'}]$, for some player p on the $c_i - c_{j'}$ path in the s.b.a $SE_{\pi_i^r,\mathbf{c}}$. In Example 5.2 we compute $\zeta(C_i, c_i, c_j, r, Z)$, for several choices of c_i and Z.

Example 5.2. *We consider a tournament played between four coalitions, A, B, C, and D, with the beating relation depicted in Figure 5.4. We assume the seeding which pairs A and B, as well as C and D in the first round. For each coalition i, we refer to the player in i, depicted in the upper side of Figure 5.4, as T_i, and to the one in the bottom side, as B_i.*

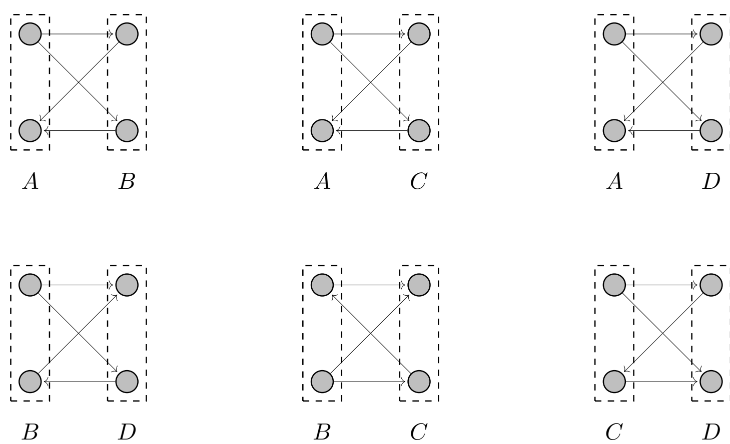

Figure 5.4: Tournament over the set of coalitions A, B, C, and D, used in Example 5.2. For clarity, we present pairwise comparisons between coalitions.

We further compute ζ for the player T_A, with $Z = \{T_A\}$, and for B_C, with $Z = \{B_C\}$. For simplicity, for a given round, we specify profiles which are in an ζ for some final opponent of a player we consider. Let us notice first that since T_A is the Condorcet winner in this tournament, all strategy profiles in which T_A participates are in $\zeta(A, T_A, c_j, r, \{T_A\})$, for some c_j, and for every $r > 0$ (note that the computation for $r = 0$ is trivial). Hence, we get the following values.

Values for the player $\mathbf{T_A}$ and $\mathbf{Z} = \{\mathbf{T_A}\}$:

- $\mathbf{r = 1}$: $(T_A, T_B), (T_A, B_B)$

- **r = 2**: (T_A, T_B, T_C, T_D), (T_A, T_B, T_C, B_D), (T_A, T_B, B_C, T_D), (T_A, T_B, B_C, B_D), (T_A, B_B, T_C, T_D), (T_A, B_B, T_C, B_D), (T_A, B_B, B_C, T_D), (T_A, B_B, B_C, B_D)

We now consider values for B_C and $\{B_C\}$. Notice that since we require B_C to be a winner of a given subtournament, and T_D beats B_C, no profile in which D nominates T_D is included in $\zeta(C, B_C, c_j, 2, \{B_C\})\}$. Similarly, profiles in which T_A or B_B meet B_C are not included in $\zeta(C, B_C, c_j, 2, \{B_C\})\}$, for every choice of c_j. Finally, observe that T_A beats all of the players that B_A encounters in a tournament in which it is nominated, and hence profiles, in which B_A participates and under which it is not a winner, are not included in $\zeta(C, B_C, c_j, 2, \{B_C\})\}$, by property (c) of Definition 5.4. This gives us the following sets of strategy profiles.

Values for the player $\mathbf{B_C}$ and $\mathbf{Z = \{B_C\}}$:

- **r = 1**: $\emptyset$
- **r = 2**: $\emptyset$

An informal description of the motivation behind Definition 5.4, is the following. If we take Z to denote c_j and its "potential future opponents" from round $r+1$ onward, then $\zeta(C_i, c_i, c_j, r, Z)$ contains precisely all those profiles over $T_r(C_i)$, such that in every strategy profile, defined over the set of all coalitions in the tournament, which is one of its extensions, there is no benefit for any coalition C_j in $T_r(C_i)$, other than C_i, to unilaterally alter its strategy. This holds because when every alternative player in C_j will either fail to win the first r rounds, by property (c) (2) of Definition 5.4), or lose to a "future opponent" in the set Z, by property (c) (1) of Definition 5.4). Notice that the set Z is never larger than $\log \#C$, as the number of rounds in the whole tournament is $\log \#C$. Then, the quasi-polynomial running time of the algorithm which we will provide arises from the fact that we have $\#N^{\mathcal{O}(\log \#C)}$ possibilities for Z. Further, in our procedure is close to iterating over all of these possibilities.

As a consequence of Definition 5.4 and Lemma 5.1, we obtain that by setting the arguments of the function ζ appropriately, one can capture all Nash equilibria. So, we get the following observation.

Observation 5.1. *For every coalitional structure (N, E, C), every NE is contained in $\zeta(C_i, c_i, c_j, \log \#C, \{c_i\})$, for some $C_i \in C$, $c_i \in C_i$, and $c_j \in C_j$. Also, for every $C_i \in C$, $c_i \in C_i$, and $c_j \in C_j$, it holds that every profile in $\zeta(C_i, c_i, c_j, \log \#C, \{c_i\})$ is a NE.*

Proof. Take a set of coalitional structure (N, E, C), and suppose that a strategy profile $\mathbf{c}$ is a NE. Let c_i be the winner of $SE_{\mathbf{c}}$, and c_j be the final opponent of c_i in $SE_{\mathbf{c}}$. Then observe that c_i beats c_j and that since $\mathbf{c}$ is a NE, for every coalition C_l and every $c'_l \in C_l$, C_l does not win in $SE_{(c_l, \mathbf{c}_{-l})}$. Notice now that condition (a) of Definition 5.4 holds, since c_i is the winner of $SE_{\mathbf{c}}$. Further, property (b) of Definition 5.4 is satisfied, since c_j is the final opponent of c_i. Finally, property (c) of Definition 5.4 holds, as by the properties of Opp, we have that c_i beats every player in $\mathsf{Opp}[p, c_{j'}]$, for all p in the path $[c_i, c_l]$ in $SE_{\mathbf{c}}$, for every coalition C_l, since $\mathbf{c}$ is a NE. But this implies that $\mathbf{c} \in \zeta(C_i, c_i, c_j, \log \#C, \{c_i\})$, and hence the first part of the claim holds. The converse holds by a similar argument. □

Given the above observation, what naturally suffices to check the existence of a NE, is to compute the function ζ, for all possible settings of the arguments. However, this may not be possible in quasi-polynomial time, because even just listing all possible Nash equilibria could be too computationally expensive. This is the case, as potentially all of the strategy profiles could be a Nash equilibrium in some game, and so their number might be exponential in the size of the input.

To overcome this obstacle, we provide a method of finding some NE, without examining all of them. Towards this end, we next prove a structural result that shows, intuitively, that if a strategy profile $\mathbf{c}$ is a NE for a given tournament, then we can reconstruct $\mathbf{c}$, or an alternative NE, by going over the subtournaments at every possible level, while examining the image of ζ.

Observe that this image is a set of strategy profiles. During this iteration, we set the arguments of ζ appropriately, and then merge the profiles corresponding to their image of ζ. In particular, we show that if there is a strategy profile $\mathbf{c}$, which is a NE, then, for every round $r \in [1, \log \#C]$, it holds that $\mathbf{c}$ extends some profile $\mathbf{c}'$ in $\zeta(C_i, c_i, c_j, r, Z)$, for some C_i, $c_i \in C_i$, $c_j \in C_j$, and Z. Moreover, for every other profile $\mathbf{c}'' \in \zeta(C_i, c_j, c_j, r, Z)$, we can replace $\mathbf{c}'$ from $\mathbf{c}$ with $\mathbf{c}''$, in order to obtain another NE.

The main algorithmic consequence of this fact is that for every choice of the arguments C_i, c_i, c_j, r, and Z, instead of computing all of the profiles contained in $\zeta(C_i, c_j, c_j, r, Z)$, it is sufficient to compute a single one. This observation holds because if any of them can be extended to a NE, then every one of them can be extended to a NE as well. The following pruning lemma constitutes the core of our algorithm.

For a pair of strategy profiles $\mathbf{c}_1 = (c_1, \ldots, c_k)$, $\mathbf{c}_2 = (c_{k+1}, \ldots, c_n)$, we denote as $\mathbf{c}_1 \cdot \mathbf{c}_2$ the *merged profile* of $\mathbf{c}_1$ and $\mathbf{c}_2$, i.e., $(c_1, \ldots, c_n)$.

Lemma 5.2. *Let $\mathbf{c} = (c_1, \ldots, c_n)$ be a NE, $r \in [1, \log \#C]$, and let the winner of $SE_{\mathbf{c}}$ be $c_{j^\star} \in C_{j^\star}$. Let further $c_i \in C_i$ be the winner of the binomial sub-arborescence $SE_{\pi_i^r, \mathbf{c}_i^r}$, and let $c_j \in C_j$ denote the final opponent of c_i in $SE_{\pi_i^r, \mathbf{c}_i^r}$. Further, let $Z = \bigcup_{p \in V(H) \setminus \{c_j\}} \mathsf{Opp}_H[p, c_j]$, where H denotes the $c_{j^\star} - c_j$ path in $SE_{\mathbf{c}}$. Then, the following facts hold:*

1. *$\zeta(C_i, c_i, c_j, r, Z)$ is non-empty.*

2. *For every c_{j_1} in some coalition C_{j_1}, for every c_{j_2} in some C_{j_2}, and for every $\hat{\mathbf{c}}_1 \in \zeta(C_i, c_i, c_{j_1}, r-1, Z \cup \{c_i\})$, and $\hat{\mathbf{c}}_2 \in \zeta(C_j, c_j, c_{j_2}, r-1, Z \cup \{c_j\})$, we have that the composed profile $\hat{\mathbf{c}}_1 \cdot \hat{\mathbf{c}}_2$ is contained in $\zeta(C_i, c_i, c_j, r, Z)$.*

Proof. Take a strategy profile $\mathbf{c}$, $c_i \in C_i$, $c_j \in C_j$, $r \in [1, \log \#C]$, $c_{j^\star} \in C_{j^\star}$, and Z, as define in the statement of Lemma 5.2. We first show that $\mathbf{c}_i^r \in \zeta(C_i, c_i, c_j, r, Z)$. Notice that, by the definition of $\mathbf{c}_i^r$, we have that c_i is the root of $SE_{\pi_i^r, \mathbf{c}_j^r}$, while c_j is the heaviest child of c_i in $SE_{\pi_i^r, \mathbf{c}_i^r}$. Hence, if $\mathbf{c}_i^r \notin \zeta(C_i, c_i, c_j, r, Z)$, then there exists $c_{j'} \in C_{j'}$, with $C_{j'} \in T_r(C_i) \setminus \{C_i\}$, such that $c_{j'}$ beats every player in Z, and $c_{j'}$ beats every player in the set $\mathsf{Opp}[p, c_{j'}]$, for every player p on the $c_i - c_{j'}$ path in $SE_{\pi_i^r, \mathbf{c}_i^r}$. Along with our choice of Z, this would then also imply that $c_{j'}$ beats every player in the set $\mathsf{Opp}[p, c_{j'}]$, for every player p on the $c_{j^\star} - c_{j'}$ path in $SE_{\mathbf{c}}$, contradicting our choice of $\mathbf{c}$ as a NE (see Observation 5.1).

For the second statement, we need to prove that properties (a), (b), and (c) in Definition 5.4 are satisfied by $\hat{\mathbf{c}} = \hat{\mathbf{c}}_1 \cdot \hat{\mathbf{c}}_2$. The first two properties are satisfied, since c_i beats c_j, $c_i = SE_{\hat{\mathbf{c}}_1}^{r-1}$, and $c_j = SE_{\hat{\mathbf{c}}_2}^{r-1}$. Suppose further that property (c) is violated. Then, for some $c_{j'} \in C_{j'}$, with $C_{j'} \in T_r(C_i) \setminus \{C_i\}$, we have that $c_{j'}$ beats all players in Z, and all players in the set $\mathsf{Opp}[p, c_{j'}]$, for each player p on the $c_i - c_{j'}$ path in $SE_{\pi_i^r, \hat{\mathbf{c}}}$. However, note that if $C_{j'} \in T_{r-1}(C_i)$, then we obtain a contradiction to our assumption that $\hat{\mathbf{c}}_1 \in \zeta(C_i, c_i, c_{j_1}, r-1, Z \cup \{c_i\})$ satisfies property (c) of Definition 5.4, and otherwise (if $C_{j'} \in T_r(C_i) \setminus T_{r-1}(C_i)$), this violates our assumption that $\hat{\mathbf{c}}_2 \in \zeta(C_j, c_j, c_{j_2}, r-1, Z \cup \{c_j\})$ satisfies property (c) of Definition 5.4. □

We now provide an algorithm for computing a NE if it exists, relying on Lemma 5.2. We establish the running time of this procedure with the following theorem.

Theorem 5.1. *There is an $\#N^{\mathcal{O}(\log \#C)}$-time algorithm for computing a NE.*

Proof. Due to Observation 5.1, our algorithm to compute a NE aims to identify and return an element of a non-empty $\zeta(C_j, c_j, c_{j'}, \log \#C, \{c_j\})$ if such $C_j, c_j, C_{j'}$, and $c_{j'}$ exist. This is necessary and sufficient. We achieve this via a dynamic programming algorithm that fills a table t, where the cells are indexed by tuples of the form $(C_j, c_j, c_{j'}, r, Z)$. Moreover, every non-empty cell indexed by the tuple $(C_j, c_j, c_{j'}, r, Z)$ contains a single element of $\zeta(C_j, c_j, c_{j'}, r, Z)$, and an empty cell, indexed by the tuple $(C_j, c_j, c_{j'}, r, Z)$, indicates that $\zeta(C_j, c_j, c_{j'}, r, Z)$ is empty. Notice that if the table is filled correctly, then the solution (i.e., a NE strategy profile) can be determined by going over all possible $C_j, c_j, C_{j'}$ and $c_{j'}$, of which there are polynomially many, and examining the entry of t indexed by the tuple $(C_j, c_j, c_{j'}, \log \#C, \{c_j\})$.

We next describe how to fill the table t. We proceed by iteratively increasing the value of r and in each iteration, filling all cells of t, which correspond to the value of r in the current iteration.

Let us consider the case where $r = 1$ and so, for every $C_j \in C$, it follows that $\#T_r(C_j) = 2$. Hence, for every $Z \in \binom{C}{\leqslant \log \#C}$, $c_j \in C_j$, and $c_{j'} \in C_{j'}$, with $C_{j'} \in T_1(C_j) \setminus \{C_j\}$, it is straightforward to decide whether $\zeta(C_j, c_j, c_{j'}, 1, Z) \neq \emptyset$ in polynomial time, by examining all possible profiles. If it is non-empty, then we compute and add to $t(C_j, c_j, c_{j'}, 1, Z)$ an arbitrary element of $\zeta(C_j, c_j, c_{j'}, 1, Z)$. Otherwise, we set $t(C_j, c_j, c_{j'}, 1, Z) = \emptyset$ (including those, indices which ζ maps to $\emptyset$ by definition). Hence, we may assume that we have filled the table t, for all entries with $r = 1$. Note that this step takes time $\#N^{\mathcal{O}(\log \#C)}$, since we have polynomially many choices for $C_j, c_j, C_{j'}, c_{j'}$, and $\#N^{\mathcal{O}(\log \#C)}$ possibilities for Z and, furthermore, determining the entry $t(C_j, c_j, c_{j'}, 1, Z)$, for each fixed choice of $C_j, c_j, C_{j'}, c_{j'}$, while computing Z, as described above, takes only polynomial time.

Now, suppose that $r > 1$, and inductively assume that for all $r' < r$, for all choices of C_j, $C_{j'}$, c_j, $c_{j'}$, and Z, we have filled the table entry $t(C_j, c_j, c_{j'}, r', Z)$ correctly. Now, let us fix a choice of C_j, $C_{j'}$, $c_j \in C_j$, $c_{j'} \in C_{j'}$, and $Z \in \binom{N}{\leqslant \log \#C}$. Further, we describe our procedure to fill the table entry $t(C_j, c_j, c_{j'}, r, Z)$. We check if there is a profile $\hat{\mathbf{c}}_1 \in t(C_j, c_j, c_{j_1}, r-1, Z \cup \{c_{j'}\})$, and $\hat{\mathbf{c}}_2 \in t(C_{j'}, c_{j'}, c_{j_2}, r-1, Z \cup \{c_{j'}\})$, for some choice of c_{j_1} and c_{j_2}. If yes, then we set $t(C_j, c_j, c_{j'}, r, Z) = \hat{\mathbf{c}}_1 \cdot \hat{\mathbf{c}}_2$, and otherwise we set it to $\emptyset$. The second point of Lemma 5.2 indicates that composing $\hat{\mathbf{c}}_1$ and $\hat{\mathbf{c}}_2$ in this way indeed results in a profile, which is contained in $\zeta(C_j, c_j, c_{j'}, r, Z)$. This implies the correctness of our algorithm.

Finally, the claimed running time bound follows from the fact that the table t has $\#N^{\mathcal{O}(\log \#C)}$ entries in total, each of which is being filled in polynomial time, by using constant-time lookups into polynomially-many previously filled entries of t.

□

5.2.2 Beyond Win-Lose Games

In earlier sections of this chapter, we assumed coalitions to be only interested in winning the tournament. However, it is natural to look at scenarios in which participants benefit from progressing as far as possible, even if they do not win. Such games occur, for instance, in football tournaments, such as the UEFA Champions League.

Here, we study tournaments played by coalitions, in which the coalitions' utility is determined by the round they reach. We call these games *beyond win/lose* (Beyond Win/Lose) games. Formally, we define utility functions for the set of coalitions $C = \{C_1, \ldots, C_n\}$, where, for every strategy profile $\mathbf{c} = (c_1, \ldots, c_n)$, we have that $u_i(\mathbf{c}) = k$ if and only $c_i \in SE^k_{\mathbf{c}} \setminus SE^{k+1}_{\mathbf{c}}$.

Hence, as in the win/lose scenario, in a NE no coalition can improve their utility, by changing their nominee.

Definition 5.5 (Beyond Win/Lose NE)**.** *A strategy profile* $\mathbf{c} = (c_1, \ldots, c_n)$ *is a* Beyond Win/Lose NE *if for all* $i, k \in [0, \log \#C]$*, and for every player* $c'_i \in C_i$*, as well as the strategy profile* $\mathbf{c}' = (c'_i, \mathbf{c}_{-i})$*, it holds that if* $u_i(\boldsymbol{c}') = k$*, then* $u_i(\boldsymbol{c}) \geqslant k$*.*

Observe that the fact that there are strategy profiles without a Beyond Win/Lose NE is witnessed by the tournament shown in Figure 5.1. In the remainder of this section, we focus on structural results characterising Beyond Win/Lose NE, and we provide algorithmic results addressing the problems of recognition and computation of this concept.

Algorithmic Properties of Beyond Win/Lose NE

We begin by providing a structural result which we then use for recognising and computing Beyond Win/Lose Nash equilibria. Intuitively, we show that a strategy profile $\mathbf{c}$ is a Beyond Win/Lose NE only if it is also the case for $\mathbf{c}$ restricted to an arbitrary subtournament.

Lemma 5.3. *Consider a strategy profile* $\boldsymbol{c} = (c_i, \ldots, c_n)$*, and let* $C_j \in C$ *be such that* c_j *wins* $SE_{\boldsymbol{c}}$*. Furthermore, suppose that* $c_{j'} \in C_{j'}$ *is the final opponent of* c_j *in* $SE_{\boldsymbol{c}}$*. Then,* $\boldsymbol{c}$ *is a Beyond Win/Lose NE if and only if :*

(a) $\boldsymbol{c}_1 = \boldsymbol{c}|_{T_{\log \#C-1}(C_j)}$ *and* $\boldsymbol{c}_2 = \boldsymbol{c}|_{C \setminus T_{\log \#C-1}(C_j)}$ *are both Beyond Win/Lose Nash equilibria.*

(b) *There is no* $c'_{j'} \in C_{j'}$*, which beats* c_j*, such that* $c'_{j'}$ *wins* $SE^{\log \#C-1}_{(c'_{j'}, \boldsymbol{c}_{2_{-j'}})}$.

Proof. In the forward direction, suppose that a strategy profile $\mathbf{c}$ is a Beyond Win/Lose NE. Observe that if property (b) is violated, then $C_{j'}$ can improve its position in $SE_{(c'_{j'},\mathbf{c}_{-j'})}$, so $\mathbf{c}$ is not a Beyond Win/Lose NE. Moreover, suppose towards contradiction that one of $\mathbf{c}_1$ or $\mathbf{c}_2$ is not a Beyond Win/Lose NE, for the respective subtournaments. Without loss of generality, suppose that for some $C_i \in C \setminus \{C_j, C_{j'}\}$, and a profile $\mathbf{c}'_1 = (c'_i, \mathbf{c}_{1_{-i}})$ it holds that $u_i(\mathbf{c}'_1) > u_i(\mathbf{c}_1)$. That is, C_i is able to improve its position in the subtournament played by $T_{\log \#C-1}(C_j)$, by nominating c'_i instead of c_i. Then, for a profile $\mathbf{c}' = (c'_i, \mathbf{c}_{-i})$, we have that $u_i(\mathbf{c}') > u_i(\mathbf{c})$. That is, C_i can also strictly improve its position in $\mathbf{c}'$, a contradiction to $\mathbf{c}$ being a Beyond Win/Lose NE. The argument for the case, in which $C_i \in C \setminus T_{\log \#C-1}(C_j)$ is symmetric, and hence property (a) also holds.

Suppose now that properties (a) and (b) hold, while $\mathbf{c}$ is not a Beyond Win/Lose NE. Then, let $c'_i \in C_i$ be such that for the profile $\mathbf{c}$'$= (c'_i, \mathbf{c}_{-i})$, we have that $u_i(\mathbf{c}\text{'}) > u_i(\mathbf{c})$. Observe that since property (b) holds, it cannot be the case that $i \in \{j, j'\}$. Moreover, if $C_i \in T_{\log \#C-1}(C_j)$, then it contradicts our assumption that $\mathbf{c}_1$ is a Beyond Win/Lose NE and, otherwise, our assumption that $\mathbf{c}_2$ is one. □

Lemma 5.3 allows us to provide a polynomial time algorithm for recognising a Beyond Win/Lose NE.

Proposition 5.3. *Recognising a Beyond Win/Lose NE is solvable in polynomial time.*

Proof. By Lemma 5.3, we have that for a given strategy profile $\mathbf{c} = (c_1, \ldots, c_n)$, and a coalition $C_j \in C$, such that c_j is the winner of $SE_{\mathbf{c}}$, it holds that $\mathbf{c}$ is a Beyond Win/Lose NE if and only if there is no $C_{j'} \in C \setminus \{C_j\}$, and no player $c'_{j'} \in C_{j'}$, such that $c'_{j'}$ beats the parent of $c_{j'}$ as well as all players in the set $\mathsf{Opp}[c_j, c_{j'}]$. The existence of a polynomial time algorithm for recognising a Beyond Win/Lose NE follows. □

We show now that computing a Beyond Win/Lose NE is solvable in polynomial time. Intuitively, our procedure is based on computing, for every round of a knockout tournament, the set of players that can possibly reach that round in some Beyond Win/Lose NE. We conclude that an equilibrium exists if the set computed for the root of the tournament is not empty.

Theorem 5.2. *Computing a Beyond Win/Lose NE is polynomial time solvable.*

Proof. Notice first that if there is a Beyond Win/Lose NE, then for $r = 1$ and for every $C_j \in C$, the subtournament over $T_r(C_j)$ can be won by exactly one out of the two coalitions in this subtournament, and that the set of potentially winning players that can participate in a Beyond Win/Lose NE can be computed in polynomial time. Indeed, suppose that C_j and $C_{j'}$ are the two coalitions in this subtournament. If every player in C_j loses to some player in $C_{j'}$, while every player in $C_{j'}$ loses to some player in C_j, then in every strategy profile, at least one out of these two coalitions will be able to improve their final position, by at least one place. This implies the non-existence of a Beyond Win/Lose NE. Hence, we identify all players in C_j (or $C_{j'}$), which beat every player in $C_{j'}$ (or, respectively, C_j), by examining all matches between players in these two coalitions. Note that this procedure runs in polynomial time.

We now inductively argue a similar property for $r > 1$. By the induction hypothesis, we have that if there is a Beyond Win/Lose NE, then for every $C_j \in C$, the subtournaments over $T_r(C_j) \setminus T_{r-1}(C_j)$, as well as $T_{r-1}(C_j)$, can be won by exactly one coalition each (say, C' and C'' respectively), and the potentially winning players from each of these coalitions (which we denote by $N^\star(C')$ and $N^\star(C'')$, respectively) that can participate in a Beyond Win/Lose NE, can be computed in polynomial time. Now, notice that for a Beyond Win/Lose NE to exist, exactly one out of the following two cases must occur. Either (1) there is a player in $N^\star(C')$, which beats every player in $N^\star(C'')$, or (2) there is a player in $N^\star(C'')$ that beats every player in $N^\star(C')$. Moreover, the set of players satisfying (1) or (2) can be computed in polynomial time. □

5.3 Dynamic Knockout Tournaments

We now allow coalitions to choose players at each round of the tournament. In this model, a strategy of a coalition C_i in a set of coalitions C consists of $\#C - 1$, not necessarily distinct, players representing, for each opposing coalition, a choice of a player of C_i to face the said opposing coalition. We therefore model choices made by coalitions as dynamic strategy profiles $\sigma : C \to (C \to N)$. Specifically, for each distinct pair of coalitions $C_i, C_j \in C$, we have that $\sigma(C_i)(C_j)$ nominates a member of coalition C_i, when facing a coalition C_j.

For every $C_i \in C$, let $\mathbf{S}(C_i)$ denote the set of all functions $C_i \to C$, i.e., the set of all possible selections of candidates from coalition C_i to compete against each other coalition. We further require a dynamic strategy profile σ to be such that $\sigma(C_i) \in \mathbf{S}(C_i)$, for each $C_i \in C$. In other words, the strategy of a coalition C_i is a selection of candidates from C_i only. For simplicity, we also use ρ_i to denote $\sigma(C_i)$, for every coalition $C_i \in C$. Equivalently, we represent σ as a tuple $(\rho_1, \dots, \rho_n)$, where $\rho_i \in \mathbf{S}(C_i)$, and denote with ρ_{ij} the player in C_i selected to play against C_j (also named $\rho_i(C_j)$). We note that in this interpretation, for every $C_i \in C$, we have that $\sigma(C_i)(C_i)$ is meaningless, as in the tournaments we consider, a coalition does not face itself. So, we assign an arbitrary player of C_i to be $\sigma(C_i)(C_i)$.

Input Representation. Notice that a dynamic strategy can be represented as a matrix M of the size $n \times n$, where n is the number of coalitions. Then, each entry of the matrix $M[i, j]$, if $i \neq j$, corresponds to $\sigma(C_i)(C_j)$. Further, if $i = j$, then $M[i, j]$ is an arbitrary player in C_i. An example of such a strategy is given in Table 5.1. Given this representation, a dynamic strategy can be encoded in space $O(n^2 \log \#N)$.

Mirroring the one-shot case, we represent tournaments between coalitions, as induced from a dynamic strategy profile. Let $T = (N, E, C)$ be a coalitional structure, and let σ be a dynamic strategy profile over C. A *dynamic coalitional digraph*, i.e., (T, σ), is a tournament defined on C, where (C_i, C_j) is an edge in (T, σ) if and only if $(\sigma_i(C_j), \sigma_j(C_i)) \in E$. This means that in a dynamic digraph, a coalition C_i is connected to a coalition C_j, when the representative of C_i, chosen to challenge C_j, beats the player selected by C_j to oppose C_i.

Example 5.3. *Let us provide an example of a dynamic coalitional digraph. The left side of Figure 5.5 shows coalitional structure, with coalitions A, B, C, and D. Some of the edges in the tournament are omitted for clarity.*

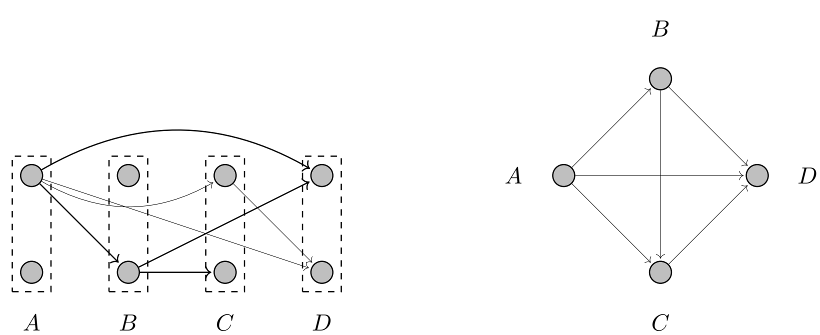

Figure 5.5: An example of a dynamic coalitional digraph. On the left, a coalitional structure. On the right, a digraph corresponding to the dynamic strategy profile depicted in Table 5.1.

Further, Table 5.1 represents a dynamic strategy profile for these coalitions, i.e., the nominations that coalitions make to compete against against a particular group. There, each row, corresponding to a coalition, specifies which of their players *is selected when opposing another coalition. For each coalition i, we refer to their top player in Figure 5.5 as T_i, and to their bottom player as B_i.*

	A	**B**	**C**	**D**
A	T_A	T_A	T_A	T_A
B	B_B	T_B	B_B	B_B
C	T_C	B_C	T_C	T_C
D	T_D	T_D	B_D	T_D

Table 5.1: Representation of the strategy profile used in Example 5.4.

Observe that under this strategy profile A beats the players selected by all other coalitions. Further, B beats C and D. Finally, given this profile, C beats D. This determines the dynamic coalitional tournament, as depicted in the right side of Figure 5.5.

Also, we lift the notion of binomial arborescence SE_σ from the one-shot case in the natural manner, by applying the definition of a binomial arborescence to dynamic coalitional tournaments. Further, we lift the notion of a Nash equilibrium from the one-shot case as expected, both for the win-lose games (dynamic NE), and the beyond win-lose games (beyond win/lose dynamic NE). We provide these definitions in later parts of this chapter (in Definition 5.6, and in Definition 5.8). The techniques used in the proofs of the results in this section are similar to their one-shot correspondents.

We define SE_σ as the s.b.a corresponding to the knockout tournament (with the fixed seeding π), played on a dynamic coalitional tournament (T, σ), and refer to it as a *dynamic knockout tournament.* Notice that SE_σ is the binomial arborescence determined by the progression of coalitions according to σ. In particular, for a coalitional structure (N, E, C), and each $i \in [0, \ldots, \log \#C]$, we denote by SE_σ^r the subtournament of SE_σ played by the the winners of the r^{th} round of SE_σ. In other words, SE_σ^0 is the same as SE_σ, and for every $i \in [1, \log \#C]$, SE_σ^r denotes the subtree of SE_σ^{r-1}, obtained by deleting all its leafs and replacing the new leafs with the corresponding winning coalitions. $SE_\sigma^{\log \#C}$ is therefore the root of SE_σ, and denotes the winning coalition.

5.3.1 Dynamic Win-Lose Games

In this section, we begin by lifting Nash equilibria to the current setting.

Definition 5.6. *For a coalitional structure (N, E, C), a dynamic strategy profile $\sigma = (\rho_1, \ldots, \rho_n)$, is a* dynamic NE *if for every coalition $C_i \in C$, and for all strategies $\rho'_i \in \mathbf{S}(C_i)$, it holds that if $SE^{\log \#C}_{\pi,(\rho'_i,\sigma_i)} = C_i$, then $SE^{\log \#C}_{\sigma} = C_i$.*

In other words, a dynamic strategy profile is a dynamic NE if no losing coalition can become a winner by changing their strategy, which now corresponds to selecting a player for each opposing coalition, unilaterally.

Now, we establish a useful structural equivalence between dynamic NE and NE, for a one-shot tournament played by coalitions, defined on an auxiliary tournament digraph where the players correspond to the set of the given coalitions' strategies. We first define what we call the *auxiliary digraph.*

Definition 5.7. *Consider a coalitional structure $T = (N, E, C)$. We define by T^{Dyn} the graph with vertex set $\bigcup_{C_i \in C} \mathbf{S}(C_i)$, and the edge set defined as follows. For every distinct pair of coalitions $C_i, C_j \in C$, and for every pair of strategies $\rho_i \in \mathbf{S}(C_i)$, and $\rho_j \in \mathbf{S}(C_j)$, there is an edge (ρ_i, ρ_j) if $(\rho_i(C_j), \rho_j(C_i))$ is an edge in E, and there is an edge (ρ_j, ρ_i), otherwise.*

That is, T^{Dyn} is defined on the set of all possible responses, of all particular coalitions. It further has the edge (ρ_i, ρ_j) if and only if the player c_i, elected to face C_j by C_i, wins against the player $c_j \in C_j$, elected to face C_i. Observe that like T, T^{Dyn} is also a $\#C$-partite tournament digraph, with a partition for every $\mathbf{S}(C_i)$, where $C_i \in C$. Example 5.4 shows an instance of T^{Dyn}, restricted to one strategy profile.

Example 5.4. *Let us consider the tournament depicted in the left side of Figure 5.5, as well as the strategy profile shown in Table 5.1. Then, Figure 5.6 presents the graph T^{Dyn}, restricted to this profile.*

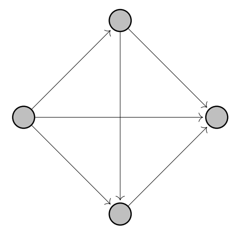

Figure 5.6: Example of a tournament T^{Dyn}, for the tournament depicted in Figure 5.5, restricted to one of the dynamic strategy profiles.

This construction allows us to reason about the dynamic solution concepts in terms of one-shot tournaments, as the following characterisation shows. Intuitively, we establish that a dynamic strategy profile is a dynamic NE exactly, when it is a NE in the corresponding one-shot tournament, based on T^{Dyn}.

Lemma 5.4. *Let $\sigma = (\rho_1, \ldots, \rho_n)$ be a dynamic strategy profile over the set of coalitions C. Then, σ is a dynamic NE for the dynamic SE-tournament over C, with a seeding π if and only if $(\rho_1, \ldots, \rho_n)$ is a NE for the one-shot SE-tournament over the coalitions $\mathbf{S}(C) = \{\mathbf{S}(C_i) : C_i \in C\}$, using the pairwise results in T^{Dyn} and seeding π, such that for every $i < \#C$, we have that $\mathbf{S}(C_i)$ is seeded at the same leaf in the one-shot tournament, as C_i in the dynamic tournament.*

Proof. Follows from the definition of dynamic solution concepts and Definition 5.7. □

Algorithmic Properties of Dynamic NE

We now address the algorithmic questions of recognising and computing a dynamic NE. Observe that in T^{Dyn}, given a coalitional structure (N, E, C), we have that the number of players in this tournament could be as large as $\#C \cdot m^{\#C}$, where m is the size of the largest coalition. As a result, although we can transfer our structural results on Nash equilibria from the one-shot setting, to the dynamic setting using the graph T^{Dyn}, we cannot simply use the same algorithms. This is because the running time, though polynomial in the size of T^{Dyn}, will no longer remain in polynomial in the size of the actual input, which is linear in the size of (N, E, C). However, using appropriate queries that can be answered in polynomial time (in the size of (N, E, C)), we can still obtain a polynomial time algorithm for recognising a dynamic

NE. In the rest of this chapter, polynomial time refers to polynomial time in the size of (N, E, C). We first demonstrate that the tractability of recognising dynamic NE is a consequence of Lemma 5.1 and Lemma 5.4.

Proposition 5.4. *Recognising a dynamic NE is polynomial time solvable.*

Proof. By invoking Lemma 5.1 on T^{Dyn}, and the equivalence, given by Lemma 5.4, we conclude that a given dynamic profile $\sigma = (\rho_1, \ldots, \rho_n)$ won by C_j is a dynamic NE if and only if there is no $C_{j'} \in C \setminus \{C_j\}$, and no strategy $\rho'_{j'} \in \mathbf{S}(C_{j'})$, such that $\rho'_{j'}$ beats all players in the set $\mathsf{Opp}[\rho, \rho_{j'}]$, for every player ρ on the $\rho_j - \rho_{j'}$ path in the b.a SE_σ contained in the graph T^{Dyn}. Observe that given σ, the arborescence SE_σ, as well as the set $\mathsf{Opp}[\rho, \rho_{j'}]$, for every strategy ρ on the $\rho_j - \rho_{j'}$ path, can be computed in polynomial time by querying, for every pair of coalitions, who is the winner in the pairwise competition between them if both played their respective strategies contained in σ. Now, for every $C_{j'} \in C \setminus \{C_j\}$, we can check whether there is a strategy $\rho'_{j'} \in \mathbf{S}(C_{j'})$, which beats the strategies selected by the respective coalitions, by inspecting the edges in the coalitional structure (N, E, C). $\square$

Then, with reasoning similar to the one-shot case we obtain a quasi-polynomial time algorithm for computing a dynamic NE if it exists.

Theorem 5.3. *A dynamic NE can be computed in $\#N^{\mathcal{O}(\log \#C)}$ time.*

5.3.2 Dynamic Beyond Win-Lose Games

Let us further analyse the final case, where coalitions can modify their choices at each round and are interested in tournament progression. The solution concept studied in this section is a natural modification of the one which we considered earlier.

We define the utility function u_i, for every coalition $C_i \in C$. For every profile $\sigma = (\rho_1, \ldots, \rho_n)$, we have $u_i(\sigma) = k$ if and only if $C_i \in SE^k_\sigma \setminus SE^{k+1}_\sigma$. That is, coalition C_i wins k rounds, but not $k+1$ rounds. Observe that this means that the utility of coalition C_i is dependent on the final round which they reach under a given strategy profile. The solution concept mirrors the previously defined Beyond Win/Lose NE.

Definition 5.8. *A dynamic strategy profile $\sigma = (\rho_1, \ldots, \rho_\ell)$ is a* Beyond Win-Lose Dynamic Nash Equilibrium *(Beyond Win/Lose dynamic NE) if for all $i, k \in [0, \log \#C]$, and every ρ'_i, as well as the profile $\sigma' = (\rho'_i, \sigma_{-i})$ it holds that if $u_i(\sigma') = k$, then $u_i(\sigma) \geqslant k$.*

Algorithmic Properties of Beyond Win/Lose Dynamic NE

Let us commence by the study of NE in the beyond win/lose, dynamic setting. We first provide a structural result on which we base the analysis of its algorithmic properties. Intuitively, we show that a strategy profile σ is a NE exactly when the player nominated by the winning coalition to compete against their final opponent beats all members of that coalition, while σ is a NE in all subtournaments.

Lemma 5.5. *Consider a dynamic strategy profile $\sigma = (\rho_i, \ldots, \rho_n)$, and let $C_j \in C$ such that C_j wins SE_σ. Furthermore, suppose that $C_{j'}$ is the final opponent of C_j in SE_σ. Then, σ is a Beyond Win/Lose dynamic NE if and only if $\sigma_1 = \sigma|_{T_{\log \# C-1}(C_j)}$, and $\sigma_2 = \sigma|_{T \setminus C_{\log \# C-1}(C_j)}$ are both Beyond Win/Lose dynamic Nash equilibria, and (b) $(\rho_j, \rho_{j'})$ is a Beyond Win/Lose dynamic NE.*

Proof. In the forward direction, consider a Beyond Win/Lose dynamic NE $\sigma = (\rho_i, \ldots, \rho_n)$. Observe that if property (b) is violated, then $C_{j'}$ can improve its position by choosing $\rho_{j'j} \in C_{j'}$ as a player which beats $\rho_{jj'}$, contradicting our assumption that σ is a Beyond Win/Lose dynamic NE. Hence, we conclude that property (b) is satisfied. On the other hand, suppose that one of σ_1 or σ_2 is not a Beyond Win/Lose dynamic NE, for the respective subtournament. Without loss of generality, suppose that for some $C_i \in C \setminus \{C_j, C_{j'}\}$, C_i is able to improve its position in the subtournament played by coalitions in $T_{\log \# C-1}(C_j)$, by choosing the strategy ρ'_i instead of ρ_i. Then, C_i can also strictly improve its position in $SE_{(\rho'_i, \sigma_{-i})}$, a contradiction to σ being a Beyond Win/Lose dynamic NE. The case when $C_i \in C \setminus T_{\log \# C-1}(C_j)$ is symmetric, and hence property (a) holds.

Assume now that properties (a) and (b) hold and suppose, towards contradiction, that σ is not a Beyond Win/Lose dynamic NE. Let $\rho'_i \in \mathbf{S}(C_i)$ be such that for the profile $\sigma' = \rho'_i, \sigma_{-i})$, we have that $u_i(\sigma') > u_i(\sigma)$. Since property (b) holds, it cannot be the case that $i \in \{j, j'\}$. Moreover, if $C_i \in T_{\log \# C-1}(C_j)$, then it contradicts our assumption that σ_1 is a Beyond Win/Lose dynamic NE. Otherwise, it contradicts our assumption that σ_2 is a Beyond Win/Lose dynamic NE. $\square$

As a straightforward consequence of Lemma 5.5, we obtain the following algorithmic result. Using the fact that for a Beyond Win/Lose dynamic NE, its restriction to any subtournament is also an equilibrium, we can check its existence in a bottom-up procedure.

Theorem 5.4. *Recognising and computing a Beyond Win/Lose dynamic NE are polynomial time solvable.*

Proof. Let us first show that computing a Beyond Win/Lose dynamic NE is can be done in polynomial time. To see that, take a coalitional structure (N, E, C). Suppose first that $\#C = 2$. Then notice that computing a NE is straightforward, since it is sufficient to check if some player in N beats all members of the opposing coalition. Indeed, if it was not the case, then, for every strategy profile, some coalition would be able to deviate profitably.

Then, consider the case in which $\#C > 2$. Observe that following this observation, by Lemma 5.5, for every pair of subtournaments $\sigma_1 = \sigma|_{T_{r-1}(C_j)}$ and $\sigma_2 = \sigma|_{C \setminus T_{r-1}(C_j)}$, which are both Beyond Win/Lose dynamic Nash equilibria, with C_j being the winner of σ_1, we can compute a Beyond Win/Lose dynamic Nash equilibrium $\sigma|_{T_{r-1}(C_j)}$, by checking if there is a member of a coalition winning one of these subtournaments, which beats all members of the winner of the other subtournament. Hence, we can compute a Beyond Win/Lose dynamic NE, if it exists, in polynomial time, as checking the existence of a NE in a tournament restricted to two coalitions can be performed in polynomial time, while the number of rounds in a knockout tournament is bounded by a polynomial. Following this observation, we can check in polynomial time if a strategy profile is a NE by evaluating if the conditions of Lemma 5.5 hold for all of the subtournaments. □

Notice now that we can have multiple dynamic Beyond Win/Lose Nash equilibria for a fixed seeding, but these are all outcome-equivalent, i.e., the winners of all subtournaments are the same under all Nash equilibria.

Proposition 5.5. *For every coalitional structure (N, E, C), $r \in [0, \log \#C]$, and dynamic strategy profiles σ_1, σ_2, which are both Beyond Win/Lose dynamic Nash equilibria, it holds that $SE^r_{\sigma_1} = SE^r_{\sigma_2}$.*

Proof. Take a coalitional structure $T = (N, E, C)$, $r \in [0, \log \#C]$, and a pair of dynamic strategy profiles σ_1, σ_2, which are both Beyond Win/Lose dynamic Nash equilibria. We show, by induction on r, that for every $C_i \in C$, it holds that the winner of $SE_{\pi^r_i, \sigma^r_{1,i}}$ is the same as the winner of $SE_{\pi^r_i, \sigma^r_{2,i}}$. Note that the claim for $r = 0$ holds trivially.

Now, let $r = 1$. Take any $C_i \in C$. Then, observe that for some coalition in $SE_{\pi^r_i, \sigma^r_{1,i}}$ there is some player in a coalition in that subtournament, which beats all members of the opposing coalition, as σ_1 is a Beyond Win/Lose dynamic NE. Without loss of generality, let this coalition be C_i. But then, there is no such player in the coalition opposing C_i in the tournament we consider. This implies that as σ_2 is also a Beyond Win/Lose dynamic NE, it holds that C_i is the winner of $SE_{\pi^r_i, \sigma^r_{2,i}}$.

We now assume that the claim holds for $r = k$. Consider further $r = k + 1$. Let then C_i and C_j be coalitions, which win the round k of the tournament, both under σ_1 and σ_2. Then, since both of these dynamic strategy profiles are Beyond Win/Lose dynamic Nash equilibria, by Lemma 5.5, we have that there is a player in C_i or in C_j, which beats all members of the opposing coalition. Without loss of generality, let this coalition be C_i. But then, following previous reasoning, C_i is the winner of the round $k + 1$ under both σ_1 and σ_2. The claim follows.

□

5.4 Conclusion

In this chapter we introduced a model for knockout tournaments played between coalitions, to allow for each group to strategically select a nominee to take part in it. We carried out an algorithmic analysis of Nash equilibrium strategies under various setups occurring in practice, showing tractability results, i.e., polynomial time or quasi-polynomial time algorithms, for all cases. In particular, we have studied one-shot scenarios, where coalitions nominate one player for the entire tournament, and dynamic ones, where choices can be refined depending on the opposing coalition. For both of these cases we have studied tournaments where coalitions only strive to win the tournament, and at tournaments where they instead take tournament progression into account. For all of the cases we considered, we analysed the algorithmic problem of computing and verifying Nash equilibria. Table 5.2 summarises the results obtained in this chapter.

	ONE-SHOT		**DYNAMIC**	
	CHECK	**FIND**	**CHECK**	**FIND**
Win/Lose NE	**P**	**QP**	**P**	**QP**
B Win/Lose NE	**P**	**P**	**P**	**P**

Table 5.2: Summary of the algorithmic results in this chapter.

We foresee various potential research directions building on our work, which involve relaxing some of the assumptions, especially on participants knowledge and strategies, and looking at alternative solution concepts. Here we discuss a few specific ones.

- It is interesting to explore the parameterised complexity of finding a one-shot or dynamic Win/Lose NE, e.g., parameterised by the number of coalitions. Although we already have a quasi-polynomial time algorithm for this problem, the existence of a fast fixed-parameter algorithm remains an interesting open question.

- All of our results provided in this chapter have assumed a tournament with a fixed seeding. However, the choice of seeding may influence the tournament significantly, in particular some seedings may admit an equilibrium, while others may not. Establishing the complexity of finding a seeding, such that a given solution concept exists is therefore an important problem in this setup. This has repercussions for tournament fixing problem, studied, e.g., in Vassilevska Williams [2010], as a malicious external attacker may be in a position to choose a seeding with a favourable winner in all resulting equilibria, and high computational complexity of finding such a seeding may act against it.

- A third direction for future research is to establish the existential and algorithmic properties of solution concepts which we considered, under relaxed versions of the model studied in this chapter. For instance, consider the scenario in which the beating relation or tournament digraph is relaxed to be stochastic. In this case, studying the existence and computational complexity of various equilibria based on the expected utility would be of high interest. Indeed, this would bring us closer to understanding a setting that models real-world scenarios more faithfully than using only a "static" tournament.

- The combination of sequential decision-making in tournaments and the beating relation between players suggests novel solution concepts. Assume that a coalition A can win a win-lose one-shot tournament, provided coalition B does not choose player b, who defeats every player in A. We have that every time B fields player b, A is indifferent to the choice of any of their players. However, there is a sense in which A must play a potential winner, should something happen to b. This suggests a trembling-hand interpretation of the beating relation (building on a classical solution concept Selten [1975]), which coalitions can try and exploit. Intuitively, a strategy profile $\mathbf{c}$ is a *trembling-hand tournament perfect equilibrium* if, for every coalition C_i, the strategy profile $\mathbf{c}$ is a Nash equilibrium at each sub-tournament, where C_i plays c_i, as if they were able to reach that sub-tournament.

- Another natural research direction concerns the possibility of cooperative and semi-cooperative behaviour among coalitions. Assume that coalition A needs coalition B to win the tournament, but B can never win. We envisage an interesting variant of endogenous games Jackson and Wilkie [2005], arising in the tournaments which we studied in this chapter. Before the tournament starts, A can transfer a part of the expected payoff to B, should B refrain from fielding a player *blocking* A. This should be seen as a form of manipulation, e.g., changing results of a certain number of matches, where incentives comes from the players themselves.

- Finally, the extension to mixed strategies is a natural follow-up, increasing the plausibility of the model, which we considered in this chapter. It is then interesting to explore whether the knockout tournament structure can add any advantage in terms of (mixed-strategy) Nash equilibrium computation.

Chapter 6

Strategic Nominations with Tournament Solutions

6.1 Introduction

In the setting of knockout tournaments played by groups, as explored in Chapter 5, coalitions are competing in rounds. So, at the start they are seeded at the leaves of a binary tree. As we have shown, the problem of checking if a NE exists in this setting is solvable in quasi-polynomial time. It is important to notice, however, that the methods used to demonstrate the tractability of this problem in the context of knockout tournaments strongly rely on the tree structure of the competition. It is therefore natural to investigate whether this result holds when other tournament solutions are considered.

In this chapter we address this problem. Towards this end, we study the problem of strategic nominee selection in the context of tournament solutions, which are based only on the pairwise comparisons between players, i.e., they do not involve a seeding of nominees. Similarly to our investigation in Chapter 5, we analyse this setting from the algorithmic game theory perspective, focusing on the existence of a Nash equilibrium. Hence, the main focus of this chapter is the study of computational complexity if a NE exists, given a set of coalitions and comparisons between individual contestants (Theorems 6.1 and 6.4). We further study the problem of whether a coalition can win under some strategy profile (Proposition 6.4 and Theorem 6.3).

Our Contribution. In this chapter we establish algorithmic properties of the Uncovered Set rule in the context of tournaments played by coalitions. We contrast it with the Condorcet Winner rule, in which only the Condorcet winner is selected if it exists. We consider three main computational questions. First, for a given method of selecting the winners from a tournament, we address the problem of checking whether a coalition can win in some strategy profile. We show that this problem is tractable for the Condorcet Winner rule, but **NP**-complete for Uncovered Set. Further, for each of the rules which we study, we analyse the problem of whether there exists a pure Nash equilibrium in the competition. Finally, we are interested in checking if a given coalition has a player which can win in some Nash equilibrium. We show that these two problems are **NP**-complete for both of the rules we consider. See Table 6.1 for an overview of our results in this chapter. Basic concepts needed for the understanding of technical content of this chapter are defined in Chapter 3.

Structure of the Chapter. In Section 6.2 we define the computational problems and tournament solutions studied in this chapter. Then, in Section 6.3, we present initial facts about both tournament solutions we consider. In Section 6.4, we investigate the properties of the Condorcet Winner rule, and in Section 6.5 we analyse the Uncovered Set rule. Finally, Section 6.6 concludes and provides directions for further research.

6.2 Computational Problems

Let us define the computational problems studied in the setting we consider in this chapter. F-Possible Winner is the problem of checking whether a given coalition has a player who is a winner under F under some strategy profile. Given a coalitional structure $T = (N, E, C)$, we say that a player c_i is a *possible winner* of T under F if there is a strategy profile $\mathbf{c}$ in which c_i is selected, such that $\mathcal{C}(c_i)$ is the winner of $T_{\mathbf{c}}$ under F.

F-Possible Winner:
Input: Coalitional structure $T = (N, E, C)$, coalition $C_i \in C$.
Question: Is there a player $c_i \in C_i$, such that c_i is a possible winner of T under F?

Further, F-Winner in NE is the problem of checking if a coalition is winner under F in some NE.

F-Winner in NE:
Input: Coalitional structure $T = (N, E, C)$, coalition $C_i \in C$.
Question: Is there a player $c_i \in C_i$ and a strategy profile $\mathbf{c}$, which is a

NE under F, such that C_i is a winner of $T_{\mathbf{c}}$ under F, and in which c_i is selected?

Finally, F-NE Existence is the problem of checking if there exists a strategy profile, which is a NE under F.

F-NE Existence:
Input: Coalitional structure (N, E, C).
Question: Is there a strategy profile $\mathbf{c}$, which is a NE under F?

Furthermore, let us define two tournament solutions, which we study in this chapter, i.e, the *Condorcet Winner* rule and the *Uncovered Set* rule. Given a tournament (N, E), a player $i \in N$ is a *Condorcet winner* if i beats every other player $j \in N$. Notice that the set of all Condorcet winners in a tournament is either a singleton or is empty. Then, the Condorcet Winner (CW) rule selects the set of all Condorcet winners.

Definition 6.1 (Condorcet Winner Rule)**.** *The Condorcet Winner rule is a tournament solution, such that for every tournament T, we have that $CW(T)$ is the Condorcet winner if it exists, and that $CW(T) = \emptyset$ otherwise.*

Furthermore, we are interested in one of the rules extending the Condorcet Winner rule, which guarantees that the set of winners is not empty. Given a tournament (N, E) and a pair of players $i, j \in N$, we say that i *covers* j, which we denote as $i \succeq j$, if $B(j) \subseteq B(i)$. So, i covers j if i beats all of the players that j beats. Observe that the fact that $i \succeq j$ implies that i beats j. Then, the Uncovered Set rule (US) selects all of the players that are not covered by any other player. Notice further that, e.g., in a cycle with three players, i.e., $E = \{(c_1, c_2), (c_2, c_3), (c_3, c_1)\}$, the set of all players is selected under US. This is because $B(c_1) = \{c_2\}$, $B(c_2) = \{c_3\}$, and $B(c_3) = \{c_1\}$. Nevertheless, in this case the set of winners under CW is the empty set.

Definition 6.2 (Uncovered Set Rule)**.** *The Uncovered Set rule is a tournament solution, such that for every tournament T we have that*

$$US(T) = \{i \in N : \text{ for every } j \in N,\ j \not\succeq i\}$$

Observe that the proposed tournament solutions readily apply to every filtration of a coalitional structure, as by the definition of a filtration it is a tournament.

Notice further how the strategy profile depicted in Example 3.2 is a NE both under CW and under US. Indeed, there, the player selected by the coalition A is a Condorcet winner in (N, E). Therefore, in every strategy profile in which it is selected, it beats and covers all other players. This implies that it is the only winner in all such profiles under both rules we consider.

6.3 Initial Remarks

Even though we restrict ourselves to two specific tournament solutions, some properties are shared by larger classes of rules, such as *Condorcet consistent* rules. We say that a rule F is *Condorcet consistent*, if for every tournament $T = (N, E)$, in which $i \in N$ is the Condorcet winner, $F(T) = \{i\}$. Observe that both *CW* and *US* are Condorcet consistent.

Let us first notice that the existence of a NE is not guaranteed for any Condorcet consistent rule.

Proposition 6.1. *For every Condorcet consistent rule F there is a coalitional structure (N, E, C) without a NE under F.*

Proof. We will proceed by showing, for every Condorcet consistent rule F, an example of a coalitional structure, which does not admit a NE under F. Consider now a Condorcet consistent rule F and the coalitional structure with two coalitions $A = \{a_1, a_2\}$ and $B = \{b_1, b_2\}$, as depicted in Figure 6.1.

Let us show now that there are no NE in this coalitional structure under F. Consider an arbitrary strategy profile (a_i, b_j) in this structure and assume that a_i beats b_j. Note that then A is the unique winner under F, since F is Condorcet consistent. But then, by the construction of the coalitional structure, there is a player $b_{3-j} \in B$, such that b_{3-j} beats a_i. Observe that B is the unique winner under the profile (a_i, b_{3-j}). So, (a_i, b_j) is not a NE. Observe that the argument is symmetric for the case in which b_j beats a_i, and hence the claim holds. □

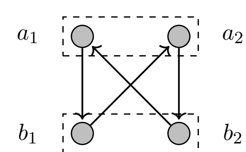

Figure 6.1: Example of a coalitional structure with no NE under any Condorcet consistent rule.

Then, under any Condorcet consistent rule, a NE exists in a coalitional structure with only two coalitions exactly when there is a player that dominates the opposing coalition.

Proposition 6.2. *For every Condorcet consistent rule F and every coalitional structure (N, E, C), such that $\#C = 2$ there exists a NE under F in (N, E, C) if and only if there is a player c which dominates the coalition C_i, such that $c \notin C_i$.*

Proof. Consider a Condorcet consistent rule F, as well as a coalitional structure $(N, E, \{C_1, C_2\})$. Observe first that since there are only two coalitions in the coalitional structure, it holds that for every filtration induced by a profile (c_1, c_2), with $c_1 \in C_1$, and $c_2 \in C_2$, there is exactly one Condorcet winner.

($\Rightarrow$) Let us reason by contraposition. So, we will show that if there is no player c, which dominates the coalition C_i, such that $c \notin C_i$, then there is no NE in the game. Suppose now that for every player $c \in C_1 \cup C_2$ there is a player $c' \notin \mathcal{C}(c)$, such that c' beats c. Further, suppose towards contradiction that there exists a profile (c_1, c_2) which is a NE. Without loss of generality, let c_1 beat c_2. Note that given this profile, as F is Condorcet consistent, we have that C_1 is the unique winner under F. Observe further that by assumption there is a $c'_2 \in C_2$, such that c'_2 beats c_1. But then C_2 wins under (c_1, c'_2), so (c_1, c_2) is not a NE, which violates the assumptions.

($\Leftarrow$) Suppose now that there exists a player $c \in C_1 \cup C_2$, such that for every player $c' \notin \mathcal{C}(c)$, we have that c beats c'. Without loss of generality, let $c \in C_1$. We will now show that there is a NE in this game. Towards this end, let us consider any strategy profile (c, c_2). Note that as we assumed that c beats c_2, and that F is Condorcet consistent, it holds that C_1 is the winner under (c, c_2). Also, as c dominates C_2, we have that C_1 wins under (c, c'_2), for every $c'_2 \in C_2$. Hence, (c, c_2) is a NE. □

6.4 Condorcet Winner Rule

Let us provide an analysis of the Condorcet Winner rule. We first observe that if a coalition contains a Condorcet winner in the initial tournament, then selecting it guarantees victory for that coalition. This simple observation will later allow us to show an interesting difference between the rules we consider.

Proposition 6.3. *For every coalitional structure $T = (N, E, C)$, strategy profile $\mathbf{c}$, and a player $c_i \in C_i$ selected in $\mathbf{c}$, if c_i is a Condorcet winner in (N, E), then C_i wins under $\mathbf{c}$ under CW.*

Proof. Take a coalitional structure $T = (N, E, C)$, strategy profile $\mathbf{c}$ and a player $c_i \in C_i$ selected in $\mathbf{c}$, such that c_i is a Condorcet winner in (N, E). Observe that c_i

is a Condorcet winner in (N, E), and hence is a Condorcet winner in $T_{\mathbf{c}}$. So, C_i is a winner in $T_{\mathbf{c}}$ under *CW*. □

Furthermore, checking if a coalition contains a player which might potentially win is tractable.

Proposition 6.4. CW-Possible Winner *is solvable in polynomial time.*

Proof. Take a coalitional structure $T = (N, E, C)$ and a coalition $C_i \in C$. We note that in order to find an answer to CW-Possible Winner for T and C_i, it is sufficient to check if there is a player $c_i \in C_i$, and a strategy profile $\mathbf{c}$, such that c_i is a Condorcet winner in $T_{\mathbf{c}}$. Observe now that it is the case exactly when, for every coalition C_j, there is a $c_j \in C_j$, such that c_i beats c_j. Notice further that this condition is verifiable by an algorithm running in $\mathcal{O}(\#N^2)$ time. Therefore, for every player c_i, we check, by examining all players in N, if there is a player beaten by c_i in every coalition $C_j \neq C_i$. The correctness of this procedure follows from previous observations. □

It is worth noting that the algorithm provided in the proof allows us to find all possible players, who are selected in a $T_{\mathbf{c}}$, under some strategy profile $\mathbf{c}$. We show, however, that checking if a NE exists is not tractable, by reduction from the 3-SAT problem. Intuitively, we construct a coalition corresponding to each variable in a formula with two players each (corresponding to literals), and a coalition containing a pair of players for each clause. We further construct what we call a *base coalition* with two players, such that the beating relation induces the matching pennies game between the base coalition and each of the pairs in the clause coalition. Having that a player in the clause coalition is beaten exactly by those players in variable gadgets, which correspond to literals in the clause, we obtain that a NE exists in the constructed game if and only if the formula is satisfiable.

Theorem 6.1. CW-NE Existence *is* **NP***-complete.*

Proof. Let us first notice that the problem we consider is in **NP**. Indeed, given a coalitional structure (N, E, C), and a strategy profile $\mathbf{c}$, we can check whether $\mathbf{c}$ is a NE by examining all potential deviations of all coalitions, which can be done in polynomial time.

Let us now show the **NP**-hardness of this problem. Take a formula φ in 3-CNF. Let $X = \{x_0, \ldots, x_n\}$ denote the set of variables in φ, and $Cl = \{Cl_0, \ldots, Cl_m\}$ be the set of clauses in φ. Let us now construct a coalitional structure $T = (N, E, C)$, which we call the *encoding* of φ.

Base, Variable, and Clause Coalitions. First, let us construct the *base coalition* S, which consists of two players, s_1 and s_2. Then, for every variable $x_i \in X$, construct a coalition $\{c_{x_i}, c_{\neg x_i}\}$, which we call a *variable coalition*. Moreover, we say that the coalition $\{c_{x_i}, c_{\neg x_i}\}$ *corresponds* to x_i and call its members *literal players* corresponding to x_i and $\neg x_i$ respectively. Observe that every strategy profile in the encoding of φ corresponds to a valuation V over X, such that the coalition corresponding to a variable x_i selects the player c_{x_i} if x_i is true in V, and $c_{\neg x_i}$ otherwise. Finally, we construct what we call a *clause coalition*, as follows. For every clause $Cl_i \in Cl$, we construct a *clause pair* $Cl'_i = \{Cl^1_i, Cl^2_i\}$. We call the members of Cl'_i *clause players*. Then, the clause coalition is the set $\{Cl^j_i : i \in [0, m-1] \text{ and } j \in \{1, 2\}\}$, i.e., it is the collection of all clause pairs. Note that the construction requires $\#X+2$ coalitions and $2\#Cl + 2\#X + 2$ players.

Tournament Relation. Let us now construct the tournament relation. First, for every literal player L, let s_1 and s_2 beat L. Furthermore, for every clause pair $Cl'_i = \{Cl^1_i, Cl^2_i\}$, let s_1 beat Cl^1_i, Cl^1_i beat s_2, s_2 beat Cl^2_i and Cl^2_i beat s_1, creating a cycle. Finally, for every literal player L and a clause player Cl^k_j, let L beat Cl^k_j if literal L is in the clause Cl_j, and Cl^k_j beat L otherwise. Construct all other edges arbitrarily. Observe that by construction a variable coalition is not a winner under any strategy profile, as its representative is always beaten by a member of the base coalition.

An example of the relation in the encoding of φ is depicted in Figure 6.2.

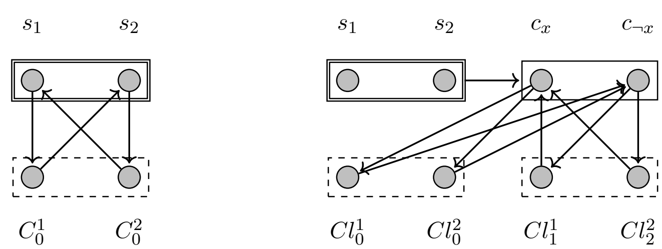

Figure 6.2: Encoding of the formula $x \wedge \neg x$. The nodes in the double rectangle depict the base pair and the nodes in the single rectangle the variable coalition corresponding to x. Moreover, in the right subfigure, the left pair in the dashed rectangle depicts a clause pair, such that x is in the clause. Further, the pair in the right dashed rectangle shows the clause, to which $\neg x$ belongs. The left figure shows the relation between the base pair and a clause pair, while the right one shows the remaining relations. It is worth noting at this stage that the tournament restricted to the pair of coalitions in the left subfigure has no NE.

Correctness of the Construction. Let us show that φ is satisfiable if and only if the encoding of φ admits a NE.

($\Rightarrow$) Suppose that φ is satisfiable. Then take a valuation V over X, which makes φ true. Further, take a strategy profile $\mathbf{c}$, such that for every variable coalition $\{c_{x_i}, c_{\neg x_i}\}$ we have that player c_{x_i} is selected whenever x_i is true in V, and the player $c_{\neg x_i}$ is selected otherwise. Also, let s_1 and C_0^1 be selected. Notice that in $T_{\mathbf{c}}$ it holds that s_1 is the Condorcet winner, as it wins against all selected literal players and against C_0^1. Furthermore, as V is a model of φ, for every clause player C_j^k there is some selected literal player L such that L beats C_j^k. Hence, the clause coalition has no profitable deviation. Finally, as we observed before, all variable coalitions lose in any strategy profile. Therefore, $\mathbf{c}$ is a NE.

($\Leftarrow$) Suppose now that φ is not satisfiable. Then, for every strategy profile there is a pair of clause players C_j', such that each of its members beats all selected literal players, as otherwise there would exist a valuation over X satisfying φ.

Let us now reason by contradiction and suppose that there is a NE strategy profile $\mathbf{c}$. Consider a pair of clause players C_j' that beats every literal player selected in $\mathbf{c}$. Observe that if there is no Condorcet winner in $T_{\mathbf{c}}$, then there exists a profitable deviation for the clause coalition. Indeed, if s_1 is selected, let the clause coalition select C_j^2, and otherwise let it select C_j^1. One can verify that in the modified profile the clause coalition is the winner under CW.

Suppose now that there is a Condorcet winner in $T_{\mathbf{c}}$. If s_1 is the Condorcet winner in $T_{\mathbf{c}}$, let the clause coalition choose C_j^2, and if it is s_2, let the clause coalition choose C_j^1. Note that in both of these cases the clause coalition becomes the winner under CW. Finally, consider the case in which a member of the clause coalition is the Condorcet winner in $T_{\mathbf{c}}$. In this case the base pair has a profitable deviation by symmetric reasoning. Hence, there is no NE in the encoding of φ. □

Furthermore, let us show that it is computationally hard to check if a coalition has a member who wins in some NE, by reduction from 3-SAT.

Theorem 6.2. CW-Winner in NE *is* **NP**-*complete.*

Proof. Let us first observe that as the verification of whether a profile is a NE can be done in polynomial time, the problem we consider is in **NP**. Let us show the **NP**-hardness of this problem. Take a formula φ in 3-CNF. Let $X = \{x_0, \ldots, x_n\}$ be the set of variables in φ, and let $Cl = \{Cl_0, \ldots, Cl_m\}$ be the set of clauses in φ. Further, consider the encoding of φ as constructed in the proof of Theorem 6.1. Let us show that the base coalition has a winner in some NE if and only if φ is satisfiable.

(⇒) Suppose that φ is not satisfiable. Then, by the reasoning used in the proof of Theorem 6.1 there is no NE in the encoding of φ. So, the base coalition is not a winner in a NE.

(⇐) Suppose that φ is satisfiable. Then, a NE in which s_1 is a winner can be constructed as in the proof of Theorem 6.1. □

6.5 Uncovered Set

Let us now consider the Uncovered Set rule. We will start with providing two observations, which showcase differences in competitions between coalitions in the context of the Uncovered Set and Condorcet Winner rules. They arise, in a large part, due to the fact that the winner under the Uncovered Set always exists but is not always unique. One can check that they do not hold for the CW rule. The first question which we ask is whether it can be the case that some player, which wins in the initial tournament, is not a winner in some filtration. Second, we check whether choosing a player which wins in the initial tournament is always beneficial for a coalition. We note that the negative answer to these questions shows that the fact that a player wins in the initial tournament does not provide an indication about their overall "strength".

Let us first observe that in the context of *US*, having a member c_i of a coalition which wins in the initial tournament does not guarantee victory in a filtration induced by a strategy profile in which c_i is selected.

Proposition 6.5. *There exists a coalitional structure $T = (N, E, C)$ and a NE profile $\boldsymbol{c}$, such that a winner i in (N, E) is selected in $\boldsymbol{c}$, but $\mathcal{C}(i)$ loses in $T_{\boldsymbol{c}}$.*

To show that Proposition 6.5 holds, let us provide the following example.

Example 6.1. *Consider the coalitional structure (N, E, C) depicted in the left side of Figure 6.3. Let us calculate the set of winners under US in (N, E). Notice that the member of the singleton coalition (call it s) is a winner under US, as s is the only player beating three rightmost players in the top tier. Further, every member i of the lower tier in the figure is a winner under US. To see that, observe that for every such player i there is a player j in the upper tier, such that i is the only one who beats both j and s. Finally, every player $i' \neq s$ in the upper tier is a winner under US, as each of them beats a different subset of players in the lower tier.*

However, notice that in the filtration induced by the profile depicted in the right side of Figure 6.3, s is beaten by all other chosen players, and thus is not a winner under US, as the set of the players that they beat is empty. Observe finally

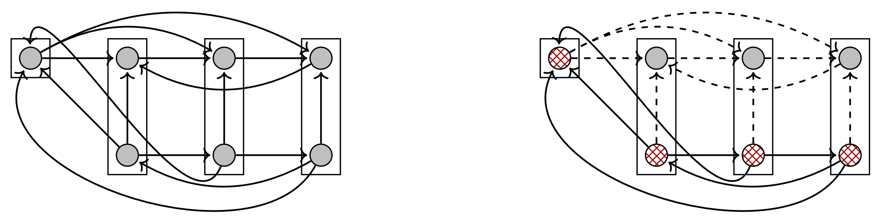

Figure 6.3: An example of a coalitional structure (N, E, C) in which a coalition whose member is winner in (N, E) does not win under some Nash equilibrium. The edges not shown in the figure point downwards.

that this profile is a NE, since all non-singleton coalitions are winners, while $\{s\}$ *only has one candidate.*

Further, we show that it can be the case that replacing a strategy with a player who is a winner in the initial tournament is not profitable.

Proposition 6.6. *There exists a coalitional structure* $T = (N, E, C)$*,* $C_i \in C$*, a winner* $c'_i \in C_i$ *of* (N, E) *under US, and a NE profile* $\mathbf{c}$ *such that a* C_i *wins in* $T_{\mathbf{c}}$*, but not in* $T_{(c'_i, \mathbf{c}_{-i})}$.

To show that Proposition 6.6 holds, let us provide the following example.

Example 6.2. *Consider the coalitional structure* (N, E, C) *depicted in the left side of Figure 6.4. Observe that the top-left player is a winner under US in the initial tournament, as it is the only one beating the top player in the centre. Observe further that the top player in the centre covers the bottom-left player, which hence is not a winner in the initial tournament. Then, consider the strategy profile inducing a filtration in right side of Figure 6.4. Observe that then, since all of the coalitions are winners under US, it is a NE. So, the left coalition is a winner in a NE. However, replacing its choice with the top-left player would result in losing the tournament, as it is beaten by all of the other selected players.*

Let us move to establishing the computational complexity of *US*-Possible Winner. We show that this problem is **NP**-complete, by reduction from 3-SAT.

Theorem 6.3. *US*-Possible Winner *is* **NP***-complete.*

Proof. Observe first that the problem is in **NP**. Indeed, given a coalitional structure and a coalition C_i, as well as a strategy profile, checking if C_i has a possible winner is

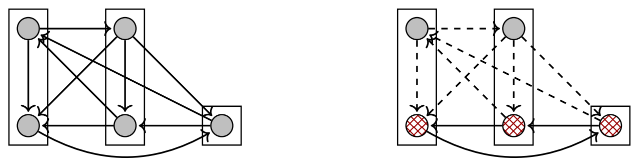

Figure 6.4: An example of a coalitional structure in which choosing a weaker player is profitable.

solvable in polynomial time, as winner determination under a given strategy profile is solvable in polynomial time for the Uncovered Set rule (see Theorem 3.6 in Brandt et al. [2016b]).

Let us show the **NP**-hardness of this problem. Take a formula φ in 3-CNF. Let $X = \{x_0, \dots, x_n\}$ be the set of variables in φ and let $Cl = \{Cl_0, \dots, Cl_m\}$ be the set of clauses in φ. Let us construct the coalitional structure, which we will call the *encoding* of φ.

Coalitions. First, let us construct a *base coalition*, with only one player s. We call s the *base player*. Further, for every variable $x_i \in X$ we construct a coalition $\{c_{x_i}, c_{\neg x_i}\}$. We say that such a coalition *corresponds* to x_i and call its members *literal players*, corresponding to literals x_i and $\neg x_i$ respectively. Finally, for every clause $Cl_i \in Cl$ let us construct a coalition $\{c_{Cl_i}\}$. We call the member of such a coalition the *clause player* corresponding to C_i. Observe that the encoding of φ includes $\#X + \#Cl + 1$ coalitions, with $2\#X + \#Cl + 1$ players.

Tournament Relation. Let us construct the tournament relation in the encoding of φ. First, for every literal player L, let s beat L. Further, for every clause player c_{Cl_i}, let c_{Cl_i} beat s. Finally, for every literal player L and every clause player c_{Cl_i}, let L beat c_{Cl_i} if the literal L is in the clause Cl_i, and let c_{C_i} beat L otherwise. Let the remaining edges be constructed arbitrarily. Notice that s beats all literal players and no other players, under any strategy profile. It is worth noting that every strategy profile in the encoding of φ corresponds to a valuation V over X, such that the coalition corresponding to a variable x_i selects the player c_{x_i} if x_i is true in V, and $c_{\neg x_i}$ otherwise. Then, we say that a clause C_i is *satisfied* in the encoding of φ under a profile $\mathbf{c}$, if the clause player c_{Cl_i} is beaten by some literal player selected in $\mathbf{c}$.

Figure 6.5 depicts the encoding of the formula $\neg x_1 \vee \neg x_2$.

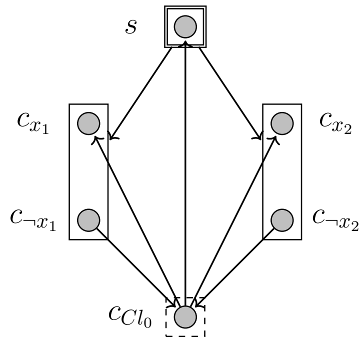

Figure 6.5: The encoding of the formula $\neg x_1 \vee \neg x_2$. The node in the double rectangle represents the base coalition. Nodes in single rectangles are variable coalitions, and in the dashed rectangle we present the clause coalition corresponding to Cl_0. Observe that the base coalition is a winner exactly when $c_{\neg x_1}$ or $c_{\neg x_2}$ is selected.

Correctness of the Construction. Let us show that $\{s\}$ has a possible winner in the encoding of φ if and only if φ is satisfiable.

($\Rightarrow$) Suppose that φ is not satisfiable. Let us show that $\{s\}$ does not have a possible winner. Consider any strategy profile $\mathbf{c}$. Notice that as φ is not satisfiable, there is a clause Cl_i, which is not satisfied in the encoding of φ under $\mathbf{c}$. This implies that there is a selected clause player Cl_i, which beats all selected literal players in $\mathbf{c}$. Observe further that as C_i also beats s, we have that s is covered by C_i. So, $\{s\}$ is not a winner under any strategy profile, and hence does not have a possible winner.

($\Leftarrow$) Suppose that φ is satisfiable. Let us show that $\{s\}$ has a possible winner in the encoding of φ. Consider a valuation V, under which φ is true. Also, take the strategy profile $\mathbf{c}$, in which every coalition $\{c_{x_i}, c_{\neg x_i}\}$, corresponding to a variable x_i, selects c_{x_i} if x_i is true in V, and $c_{\neg x_i}$ otherwise. Observe that as φ is true under V, for every clause player in $\mathbf{c}$ there is a literal player in $\mathbf{c}$ that beats it. Hence, s is the only selected player which beats all chosen literal players. Hence, s is not covered by any selected player under some strategy profile. Therefore, $\{s\}$ has a possible winner. □

Furthermore, we show that *US*-NE Existence is **NP**-complete, by reduction from 3-SAT.

Theorem 6.4. *US*-NE Existence *is* **NP**-*complete.*

Proof. Let us first observe that the problem we consider is in **NP**. Indeed, for a given strategy profile, we can check in polynomial time whether a given coalition can improve their utility by replacing their representative, as winner determination is possible in polynomial time for *US*. Let us then show the **NP**-hardness of this problem. Take a formula φ in 3-CNF. Let $X = \{x_0, \dots, x_n\}$ be the set of variables in φ and let $Cl = \{Cl_0, \dots, Cl_m\}$ be the set of clauses in φ. Let us assume for simplicity that $\#X \geqslant 3$. This is without loss of generality, as every formula in 3-CNF can be extended to an equivalent formula in 3-CNF with at least 3 variables. Let us construct the coalitional structure (N, E, C), which we will call the *encoding* of φ.

Coalitions. First, construct the *base coalition* S, consisting of two players, s_1 and s_2. We call the members of this coalition *base players.* Further, for every variable $x_i \in X$, let us construct a *variable coalition* $\{c_{x_i}, c_{\neg x_i}\}$. We say that such a coalition *corresponds* to x_i, and call its members *literal players*, corresponding to x_i and $\neg x_i$ respectively. Then, for every $x_i \in X$, we construct an *auxiliary* coalition $\{A_{x_i}\}$, and call its member an *auxiliary player* corresponding to x_i. Finally, we construct what we call a *clause coalition* as follows. For every clause $Cl_i \in Cl$, we construct the *clause pair* $Cl'_i = \{Cl^1_i, Cl^2_i\}$. We call the members of Cl'_i *clause players.* Then, the clause coalition is the set $\{Cl^j_i : i \in [0, m-1]$ and $j \in \{1, 2\}\}$, i.e., the union of all clause pairs. Observe that every strategy profile in the encoding of φ corresponds to a valuation V over X, such that for every variable $x_i \in X$, it holds that x_i is true in V if c_{x_i} is selected by the corresponding variable coalition, and false if $c_{\neg x_i}$ is selected. Note further that the encoding of φ requires $2\#X + 2$ coalitions and $2\#Cl + 3\#X + 2$ players.

Tournament Relation. Let us construct the tournament relation in the encoding of φ. First, for each literal player L, let s_1 and s_2 beat L. Further, for every clause player Cl^k_j and a literal player L, let L beat Cl^k_j if L is in the clause Cl_j, and let Cl^k_j beat L otherwise. For a strategy profile $\mathbf{c}$, in which some selected literal player L beats Cl^k_j, we say that C_j is *satisfied* in $\mathbf{c}$. Also, for every auxiliary coalition $\{A_{x_i}\}$, let A_{x_i} beat all variable, base and clause players, apart from c_{x_i} and $c_{\neg x_i}$. Instead, let c_{x_i} and $c_{\neg x_i}$ beat A_{x_i}. Furthermore, for every clause pair $Cl'_i = \{Cl^1_i, Cl^2_i\}$, let s_1 beat Cl^1_i, Cl^1_i beat s_2, s_2 beat Cl^2_i, and Cl^2_i beat s_1, constructing a cycle. Let the remaining edges be constructed arbitrarily, while ensuring that each auxiliary player A_{x_i} is beaten by some auxiliary player A_{x_j}, and for such a pair, members of the coalition corresponding to x_i are beaten by the members of the coalition corresponding to x_j.

Note that such a relation exists, since $\#X \geqslant 3$. This occurs, for instance, when there exists a cycle containing all auxiliary players, mirrored by variable players. To see that, consider the case in which $X = \{x_0, x_1, x_2\}$. Then, the condition mentioned above is satisfied, if (1) A_{x_0} beats A_{x_1}, A_{x_1} beats A_{x_2}, and A_{x_2} beats A_{x_0}, while (2) we have that $\{c_{x_0}, c_{x_0}\}$ dominates $\{c_{x_1},\ c_{x_1}\}$, $\{c_{x_1}, c_{x_1}\}$ dominates $\{c_{x_2},\ c_{x_2}\}$, and $\{c_{x_2}c_{x_2}\}$ dominates $\{c_{x_0},\ c_{x_0}\}$. The key relation in an encoding of φ is partially depicted in Figure 6.6.

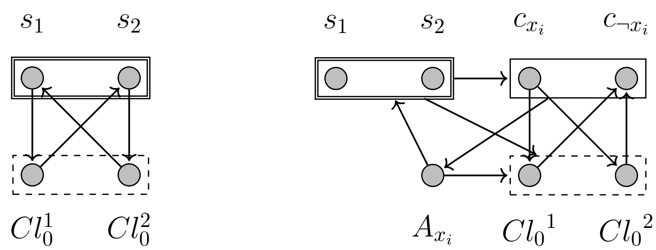

Figure 6.6: Key relations in the encoding of φ. Players in the double rectangle represent the base coalition, in the single rectangle a variable coalition corresponding to x_i, and in the dashed rectangle, a clause pair. In the depicted fragment of the construction, x_i belongs to the presented clause, while $\neg x_i$ does not. The remaining player is the auxiliary player A_{x_i}. The left figure shows the relation between the base pair and the clause pair, while the right one shows the relation between the remaining coalitions.

Correctness of the Construction. Let us start with stating a few properties of the encoding of φ.

1. Let us show first that no variable coalition is a winner under any strategy profile. To see that, take any variable coalition $\{c_{x_i}, c_{\neg x_i}\}$ and an auxiliary player A_{x_j}, which beats A_{x_i}. We know that such an auxiliary player exists by construction of the tournament relation. Observe that under any strategy profile, A_{x_j} beats all players which are beaten by c_{x_i} or $c_{\neg x_i}$. Namely, it beats all literal players apart from c_{x_j} and $c_{\neg x_j}$, players in the base coalition, all clause players, and A_{x_i}. Also, by construction of the tournament relation, c_{x_i} and $c_{\neg x_i}$ are beaten by c_{x_j} and $c_{\neg x_j}$. So, a representative of the coalition corresponding to x_i is covered by A_{x_j}, and hence this coalition is not a winner under any strategy profile.

2. We further show that if in a profile, in which the clause coalition chooses a player C_j^k, some chosen literal player L beats Cl_j^k, then the clause coalition is not a winner. Without loss of generality, let such Cl_j^k be beaten by a selected player c_{x_i}. Notice then that the player A_{x_i} beats all of the players that Cl_j^k does, as it beats all literal players, apart from the members of the coalition corresponding x_i, as well as the base players. Hence, the clause coalition is not a winner under such profile, as Cl_j^k is covered by A_{x_i}.

3. Finally, let us observe that a selected clause player Cl_j^k is not covered by any auxiliary player A_{x_i}, such that Cl_j^k beats the player selected by $\{c_{x_i}, c_{\neg x_i}\}$, as such a literal player beats A_{x_i}.

We are now ready to show that there exists a NE in the encoding of φ if and only if φ is satisfiable.

($\Rightarrow$) Suppose that φ is not satisfiable. Let us show that there exists no NE in the encoding of φ. Observe that as φ is not satisfiable, we have that for every valuation over X there is some clause Cl_j, such that all of its literals are false in X. Observe that this implies that in the encoding of φ we have that, for every strategy profile $\mathbf{c}$, there is a clause which is not satisfied in $\mathbf{c}$ (call it Cl'_j). Hence, there is a pair of clause players Cl'_j, such that both members of Cl'_j beat every literal player, which is selected in $\mathbf{c}$. We now consider the following exhaustive cases.

Case 1: The clause coalition nominates a member of a pair Cl'_j, beating all selected literal players. Then notice that by 3., the clause coalition is a winner under *US* exactly when it is not covered by the selected member of the base coalition. Similarly, as the chosen member of the base coalition dominates all of the coalitions corresponding to variables, we have that is a winner under *US* exactly when it is not covered by the selection of the clause coalition. We first show that there is no NE, in which Cl_j^1 is nominated. Note that if Cl_j^1 and s_1 are selected, then Cl_j^1 is covered by s_1. It is not the case, however, if Cl_j^2 and s_1 are selected, so the clause coalition has a profitable deviation. Further, if Cl_j^1 and s_2 are selected, s_2 is covered and the base coalition has a profitable deviation to s_1. So, there is no NE if Cl_j^1 is selected. Symmetrically, it can be shown that there is no NE in which Cl_j^2 is chosen. Hence, there is no NE in the encoding of φ.

Case 2: Otherwise. Then notice that if the clause coalition nominates a player, which is beaten by some selected literal player L, then the clause coalition is not a winner under *US*, by 2. One can verify, however, that the clause coalition becomes a winner under *US*, by switching their choice to some player in Cl'_j. Hence, there is no NE in the encoding of φ, if φ is not satisfiable.

($\Leftarrow$) Suppose that φ is satisfiable. Let us show that there exists a NE in the encoding of φ. Consider a valuation V over X that makes φ true. We know that it exists, since φ is satisfiable. Also, take a strategy profile $\mathbf{c}$, such that, for every coalition $\{c_{x_i}, c_{\neg x_i}\}$ corresponding to some variable x_i, we have that c_{x_i} is selected whenever x_i is set to true by V, and $c_{\neg x_i}$ is selected otherwise. Also, let s_1 and C_0^1 be selected. Notice that as V satisfies φ, for every player Cl_j^k in the clause coalition, there is a selected literal player, which beats C_j^k. So, by 2., the clause coalition is not a winner regardless of their choice, and thus has no profitable deviation. Further, as we have observed in 1., variable coalitions do not win under any strategy profile, and hence have no profitable deviation. Moreover, s_1 is the only selected player beating all chosen literal players and thus it is a winner under *US*. So, the base coalition has no profitable deviation. Finally, all other coalitions are singletons and therefore have no profitable deviations. So, **c** is a NE. □

Finally, we establish the complexity of *US*-Winner in NE, by reduction from 3-SAT.

Theorem 6.5. *US*-Winner in NE *is* **NP**-*complete.*

Proof. Observe first that the problem is in **NP**, as checking if a profile is a NE and winner determination is possible in polynomial time for *US*. Let us further show the hardness of this problem. Take a formula φ in 3-CNF. Let $X = \{x_0, \ldots, x_n\}$ be the set of variables in φ, and let $Cl = \{Cl_0, \ldots, Cl_m\}$ be the set of clauses in φ. We construct the coalitional structure, which we call the *encoding* of φ, as in the proof of Theorem 6.4.

Observe first that $\{A_{x_0}\}$ is a winner under any strategy profile. Indeed, in any strategy profile A_{x_0} beats the selection of base coalition and of clause coalitions, and hence is not covered by them. Also, it is not covered by selected literal players, as they do not beat the selected base player. Finally, A_{x_0} is not covered by any auxiliary player A_{x_j}, such that $j \neq 0$, as it beats the selection of $\{c_{x_0}, c_{\neg x_0}\}$, which by construction beats A_{x_j}. So, $\{A_{x_0}\}$ is a winner under *US* in any strategy profile. Let us show now that the coalition $\{A_{x_0}\}$ is a winner in some profile which is a NE if and only if φ is satisfiable.

($\Rightarrow$) Suppose that φ is not satisfiable. Then, by the reasoning similar to the proof of Theorem 6.4, it holds that there is no NE in the encoding of φ. But then ,$\{A_{x_0}\}$ is not a winner in a NE.

($\Leftarrow$) Suppose that φ is satisfiable. Then, by reasoning in the proof of Theorem 6.4, there exists a NE in the encoding of φ. Also, as we have shown, $\{A_{x_0}\}$ is a winner in this NE, so the claim holds. □

6.6 Conclusion

In this chapter we provided an algorithmic analysis of nominee selections in the context of tournament solutions. We analysed two methods of selecting the set of winners, namely Condorcet Winner and Uncovered Set rules.

Summary of Contributions. As we demonstrated, checking if a pure Nash equilibrium exists for a coalitional structure is **NP**-complete under both of the rules we considered. This result shows a significant difference between the complexity of reasoning about these rules and knockout tournaments, which allow us to find a Nash equilibrium in quasi-polynomial time if it exists, as shown in Chapter 5. Furthermore, under the Uncovered Set rule, it is not tractable even to check if a coalition can win in some strategy profile, which constitutes a major difference with respect to the competitions conducted in the context of Hotelling-Downs model or of knockout tournaments, which we studied in Chapter 4 and Chapter 5. Table 6.1 provides a summary of our contribution in the present chapter.

	CW	**US**
Possible Winner	**P**	**NP**
Winner in NE	**NP**	**NP**
NE Existence	**NP**	**NP**

Table 6.1: Algorithmic results shown in this chapter. Here, value **NP** indicates that a problem is **NP**-complete.

Future Research. Our results provide a vast range of directions for future investigations. Let us discuss some direct possibilities for further research based on the results of this chapter.

- We limited ourselves to two particular tournament rules only. However, there exists a large number of tournament solutions studied in the computational social choice literature. This would motivate a symmetric analysis for other rules, such as the *Copeland rule*. In particular, a natural direction would be to establish computational complexity of finding a NE for all Condorcet-consistent rules.

- We only considered pure Nash equilibrium as a solution concept. Nevertheless, in many cases it would not be ideal from the perspective of predicting the coalitions' choices. It is a natural avenue for further research to check algorithmic properties of other concepts, such as dominant strategy equilibria, in the setting studied in this paper.

- In our setting we assumed that the beating relation is asymmetric, which in the social choice context corresponds to the assumption that, for a pair of players, one of them is preferred to another by the strict majority of voters (or that a tie-breaking rule is applied). This assumption is not applicable in case of many social phenomena, in which accounting for the ties between players is important. We therefore propose to study the case in which the tournament relation is not asymmetric.

- Even though most of the problems studied in this chapter are computationally hard, it can be the case that they are tractable in typical cases. It is therefore interesting to analyse them from the perspective of parametrised complexity. In particular, it is natural to consider the number of coalitions as a parameter.

- We note that certain coalitional structures always admit a Nash equilibrium for all Condorcet consistent rules, e.g., when some player dominates all of the coalitions, to which it does not belong. Therefore, identification non-trivial classes of games, in which the existence of an equilibrium is guaranteed for meaningful classes of rules is a natural open problem.

- In the setting studied in this chapter, the tournament relation is deterministic, i.e., it is certain who is the winner of each pairwise contest. This observation motivates the exploration of the generalisation of this framework, in which it does not always hold, as a natural follow-up study.

- One of the interpretations of the beating relation, in the context of the social choice theory, is the result of a pairwise majority contest. Thus, a potential avenue for further research involves the strategic behaviour of voters in the setting studied in this chapter. Namely, one could consider agents misrepresenting their preferences over players to have a better player (in their view) selected by a winning coalition.

Chapter 7

Reaching Stability in Opinion Diffusion

7.1 Introduction

When debates between agents are considered, it is crucial to take into account that they might be willing to change their opinion, when influenced by their interlocutors. As such, it is natural to study *opinion diffusion protocols* in the context of such discussions. In a particularly well-studied class of such protocols, agents change their view if a specified fraction of those with whom they communicate disagrees with them. We call such mechanisms *threshold-based protocols* (see, e.g., Granovetter [1978]). It is further worth noting that opinion diffusion protocols are relevant in the context of spread of desirable opinions. For instance, a political party might be willing to convince a limited number of particular voters to support them, in order to receive a large number of votes. In relation to this problem, in their influential paper, Kempe et al. [2003] show that it is **NP**-hard to find an optimal set of such individuals.

One of the major challenges associated with the application of threshold based opinion diffusion protocols is that their convergence is not guaranteed. Imagine you would like to have your agents make a collective decision and let them discuss first, agreeing that they would cast their vote once they have made up their mind. Depending on the chosen diffusion protocol and the initial distribution of opinions, the process might never terminate. Such cases can arise, in particular, in synchronous threshold models, where all of the agents revise their opinions in every step. Clearly, any network will converge for *some* initial input, for instance when all of the agents already think the same to start with. However, this is not true in general. Therefore, it is important from the perspective of the prediction of the outcome of a debate to know whether the protocol will ever terminate.

A typical path taken to circumvent the issue in the study of problems regarding opinion diffusion is to restrict the analysis to networks that always converge, as studied by Grandi et al. [2015], Bredereck and Elkind [2017], and Botan et al. [2019]. Another is to consider specific protocols which guarantee termination, as done for instance by Auletta et al. [2018]: they propose an opinion-revision protocol for agents who disagree with a distinguished opinion. In this chapter we are interested, however, whether it is possible to efficiently check if we can employ an unrestricted opinion diffusion protocol in a given case.

It is also worth noting that recently Christoff and Grossi [2017] have provided a characterisation of networks in which termination of the threshold-based opinion diffusion protocol is guaranteed. However, we still do not know whether characterising convergent networks is of any advantage for their algorithmic analysis, in other words, whether we can have a characterisation that is easier to check than actually running the protocol until converging or looping in some way. Here, we settle this problem.

Our Contribution. We study the convergence of opinion diffusion in social networks, modelled as directed graphs over a finite set of individuals, who simultaneously update their opinions. They switch their opinions if and only if the majority of their influencers disagrees with them. We look at labelled networks, where individuals start with a binary opinion, and study the problem of whether that network converges. We also look at unlabelled networks and consider the problem of whether a labelling exists for which the network does not converge. This problem concerns the *structural* aspect of opinion diffusion's convergence, i.e., whether the structure of the influence relation alone might guarantee convergence. Our contribution is two-fold: firstly, we present some classes of networks which are guaranteed to converge, and secondly we show that the problem of establishing whether a network converges is **PSPACE**-complete even for the simplest of such protocols, closing a gap in the literature. In fact, our result implies that any characterisation of such networks, including the one provided by Christoff and Grossi [2017], cannot result in an efficient procedure for verifying the convergence of the protocol we consider (unless **P**=**PSPACE**).

We emphasize that even though our protocol is relatively simple, the computational complexity lower bounds that we obtain extend directly to more general models. For instance, the **PSPACE**-hardness of the problems we consider lifts to the scenario in which each agent has its own specific update threshold. Thus, our result implies that no complete characterisation of convergent networks can be efficiently computed in practice for a wide range of plausible diffusion protocols.

Structure of the Chapter. First, in Section 7.2 we provide examples of networks, for which it is easy to check if they are convergent for an arbitrary labelling. Subsequently, in Section 7.3 we prove that determining convergence is **PSPACE**-complete in our setting. Finally, in Section 7.4 we conclude by discussing the ramifications of our results and future research directions.

7.2 Graph Restrictions

We observe that all networks are convergent for some initial labellings, e.g., if all of the vertices are labelled with the same colour. However, some networks converge for all initial labellings, while others converge for just some of them. The left subnetwork in Figure 7.1, for example, converges for every labelling. However, the social network displayed in the right subfigure of Figure 7.1 behaves differently. I.e., it converges exactly when all of its vertices are labelled with the same colour.

Figure 7.1: In the left subfigure, a social network which converges for every labelling. In the right, a social network which does not converge, unless all vertices are labelled with the same colour.

We now focus on specific instances of social networks which converge for every labelling. Let us start with DAGs, i.e., *directed acyclic graphs.* Even though this observation is well-known, we include its proof, as it is vital for our further reasoning.

Proposition 7.1. *Let $SN{=}(N,E)$ be a DAG. Then, SN converges in at most k steps, for every labelling f, where k is the length of the longest path in SN.*

Proof. Given a network $SN{=}(N,E)$, which is a DAG, consider an arbitrary labelled f of SN. Since SN is acyclic, for every $i \in N$ that is not a source vertex, there is a path to i from some source vertex of SN. Let $level(i)$ be the length of the longest such path. We will show by induction on $level(i)$ that every $f(i)$ will stabilise after at most $level(i)$ updates.

If *level*(i) is 0, then i is a source vertex, and therefore the labelling of i never changes. Take now a natural r. Suppose further that all i, such that *level*$(i) \leqslant r$, have stabilised after r updates. Then take a vertex i with *level*$(i) = r+1$. Since SN is acyclic, for every $i' \in N$, such that $i' \to i$, we have *level*$(i') \leqslant r$. This means that i' is already stable after r updates. Hence, i will stabilise within one step after all its influencers have stabilised, i.e., after at most $r+1$ updates. □

Networks that are not DAGs do not always converge, as shown in Figure 7.1. But some of them, such as cliques, have interesting properties with respect to convergence.

Proposition 7.2. *Let SN=(N, E) be a clique. SN converges for every labelling if and only if $\#N$ is odd. Moreover, if SN converges, then it does so after a single update step.*

Proof. Suppose (N, E) is a clique and that $\#N$ is even. We will show that it does not converge for some labelling. To see that consider a labelling f of (N, E), such that $\#\{i \in N : f(i) = b\} = \#\{i \in N : f(i) = r\}$, as well as the labelled social network $SN = (N, E, f)$. As E is irreflexive, we have that, for each $i \in N$, it holds that $A(i) < D(i)$. It then follows that, for every such i, $OD(SN, i) \neq f(i)$. But this means that after the update we still have that $\#\{i \in N : f(i) = b\} = \#\{i \in N : f(i) = r\}$. So, for every network SN_j in SN_{us}, it holds that the colour of i is different in SN_j, than in SN_{j+1}. Therefore, SN does not converge.

Suppose now that (N, E) is a clique and that $\#N$ is odd. Consider further any $f : N \to \{b, r\}$, and assume without loss of generality that $\#\{i \in N : f(i) = r\} > \#\{i \in N : f(i) = b\}$. Note that the size of sets B_f and R_f are not equal, since $\#N$ is odd. Take now an arbitrary agent i, such that $f(i) = r$. Observe now that $A(i) \geqslant D(i)$. But then $OD(SN, i) = r$, since $f(i)$ is the majority colour in the clique. If instead $f(i) = b$, then $A(i) < D(i)$, and hence $OD(SN, i) = r$. So, after a single update step all agents are labelled r. Thus, the network converges. □

We note that checking if a social network (N, E) is a clique can be done in linear time, by calculating the size of E. As a consequence, the result above shows that, for some structures, finding whether they converge for every labelling is immediate.

Consider now the *strongly connected components* (SCCs) of a social network. One might expect that if we knew that each SCC always converges, then so would the whole network. Or, in other words, that every network that always converges will also do so, when its members are only influenced by agents in a network that always converges as well. This, in result, could lead to a reduction of the complexity of checking if there exists a labelling of a social network which does not converge. Remarkably, this is not true even for very simple cases, as exemplified in Figure 7.2.

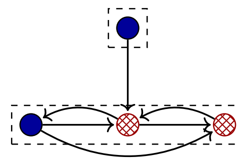

Figure 7.2: A labelled network that does not converge, whose two SCCs (marked by rectangles) do converge for every initial labelling. Observe that the fact that the SCC in the lower tier does not converge holds because of the incoming edge from the upper SCC.

The example provided above indicates that some networks admit labellings which do not converge, even if they are composed of well-behaved fragments. We now move on to the problem of checking convergence in an arbitrary social network.

7.3 The Complexity of Checking Convergence

We analyse two computational problems with respect to the protocol we are considering. The first of them is checking the convergence of a given labelled social network.

CONVERGENCE:
Input: Social network $SN = (N, E)$ and labelling f.
Question: Does SN converge from f?

The second is checking, for an unlabelled network, whether there is a labelling, for which it *does not* converge.

CONVERGENCE GUARANTEE:
Input: Social network $SN = (N, E)$.
Question: Is there a labelling of SN for which SN does not converge?

In the remainder of this section we will prove theorems associated with these two computational problems.

Theorem 7.1. CONVERGENCE *is* **PSPACE**-*complete.*

Theorem 7.2. CONVERGENCE GUARANTEE *is* **PSPACE**-*complete.*

It is important to note that due to the result which we prove in this section, the problems that we study are also **PSPACE**-hard for all opinion diffusion models for which our protocol is a special case. In particular, this holds in models with agent-dependent update thresholds, i.e., where the fraction of disagreeing neighbours needed for an individual to change their opinion might be different for distinct agents. Another example of such models is the one with weighted trust levels, i.e., with weighted majority instead of majority update rule.

Let us argue that both problems belong to **PSPACE**. This observation holds because each labelling of a social network $SN = (N, E)$ takes at most $\mathcal{O}(\#N)$ bits, as we can represent it as a string of length $\#N$, where each position indicates the colour of a vertex. Notice then that each network admits $2^{\#N}$ labellings and thus, if a a given labelled network is convergent, it reaches a stable network in at most $2^{\#N}$ opinion diffusion steps. Therefore, solution to the problems we consider in this chapter can be obtained using polynomially bounded space by examining a bounded number of opinion diffusion steps. Further, the synchronous update mapping SU can be evaluated in polynomial time (for a single step of diffusion), as in order to compute it, we only need to examine the neighbourhood of each agent once.

The hardness proof of Theorem 7.1 can be developed separately, but we choose to give a uniform presentation and derive hardness of both problems from the same construction, in order to make the proof of Theorem 7.2 easier to follow.

7.3.1 Ingredients for the Hardness Proofs

The main technical challenge for the hardness proof is that the update mapping SU is based on the majority update. This means that if for an agent i and a pair of labellings f, f' we would have that for all j in the set $N^{-1}(i) \cup \{i\}$ it holds that $f(j) \neq f'(j)$, then, after the update, i will also have a different colour in f and in f'. It then follows, informally, that the vertices are indifferent to the identity of blue and red (which will be further identified with binary truth values, 0 and 1, respectively). But then, since it is not possible to immediately distinguish between these values, we cannot simulate propositional logic directly.

Propositional Logic and Dual Rail Encoding. Let us introduce the basic technical notions appearing in the proofs of hardness of the problems we consider. We will use Boolean circuits, the description of which can be found in Chapter 3 and, more extensively, e.g., in [Papadimitriou, 1994, section 4.3]. Signals in these circuits are Boolean values, TRUE and FALSE, and we will encode them in our social networks. We need to encode logical gates (AND and NOT) and constant gates (TRUE and FALSE) too. We also encode the NOP (i.e., no-operation) gate.

We use the dual rail encoding due to the monotonicity of the opinion diffusion protocol. Indeed, notice that in the current setting, for a pair of vertices i, j, labelled with the same colour, adding an edge from i to j will not result in the change of j's label in one step. This makes it impossible to model logical negation, while representing this operation as a relation between individual vertices.

In the dual rail encoding, instead of considering individual vertices in a social network, we will be often considering related pairs of vertices, called *dual pairs*. The two vertices in a dual pair are ordered. Given a labelling of the network, a dual pair is *valid* if its two vertices disagree, i.e., take different values, and *invalid* otherwise. Dual pairs will be the building blocks in our construction, and our network will have a mechanism to ensure their validity.

Our first step is to build constant gates. We introduce a distinguished dual pair, the *base pair*. As long as it is valid, we assume without loss of generality that its two vertices are coloured (r, b). There is only one base pair in the network. Now, for every valid dual pair in the network, we interpret the colouring of this pair (r, b) as *true*, and (b, r) as *false*.

The next step is to build logical gates. All these gates in our circuits have indegree 1 or 2, that is, each gate receives input from at most two other gates. The gates are depicted in Figures 7.3 and 7.4 and described in Example 7.1. We encode AND, NOT, and NOP gates. It is worth mentioning at this stage, that NOP gates will be important for our reduction, as they will allow us to ensure that the length of paths in the constructed network is appropriate.

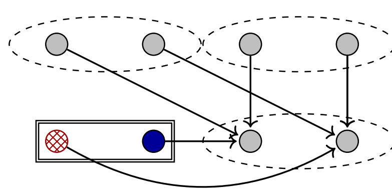

Figure 7.3: The AND gadget.

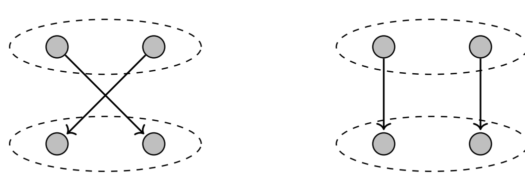

Figure 7.4: NOT gadget on the left, NOP gadget on the right.

Example 7.1. *The gadget in Figure 7.3 models an* AND *gate, and the gadget in the left side of Figure 7.4 models a* NOT *gate. The* AND *gadget relies on the base pair, which is depicted as a double rectangle. In more detail, if at time t the input dual pairs (the two upper ovals) in the* AND *gadget are valid, then at time $t+1$ the output dual pair is valid and represents the* AND *of the two input values. Similarly, for the* NOT *gate, if at time t the input dual pair (upper oval) is valid, then the output dual pair corresponds to the negation of the input value. Finally, the gadget in the right side of Figure 7.4 models a* NOP *(no operation) gate. There, at time $t+1$ the output pair is a copy of the input pair at time t.*

Turing Machines. Further, in our reduction we will need to build Boolean circuits to simulate the behaviour of Turing machines.

We will describe a restricted version of Turing machines that we use to prove Theorems 7.1 and 7.2. These Turing machines are *polynomially space-bounded*, or **PSPACE** *machines* (referring to the complexity class). These machines only use space bounded by a function on their input. See Chapter 3 for the description of Turing machines (and Papadimitriou [1994] for an extensive overview).

We will rely on the following properties of such Turing machines:

1. Any Turing machine has a finite description.
2. Any Turing machines can be *run* on arbitrary input strings of arbitrary length $m \geq 0$ over a fixed finite alphabet.
3. A *run* is a finite or infinite sequence of configurations. Each configuration is either *halting* or has a unique *successor* configuration.
4. At any point during a run, an instantaneous description of a Turing machine M (a *configuration*) can be encoded by a bit string of length $c \cdot m^d$, where the constants c and d depend only on the machine M.
5. A Turing machine may either *halt* at some point during the run, or *diverge* (run forever).

We will identify configurations of a Turing machines with their encodings as n-bit strings (strings of truth values), with n dependent on the machine. Here, $n = c \cdot m^d$, where m is fixed, while n is the same in all possible configurations the machine.

For a given n, we will assume for the sake of simplicity that all n-bit strings represent valid configurations. This assumption does not invalidate our reduction and can in fact be eliminated using the technique of the following lemma.

Lemma 7.1. *Given a Turing machine M and an integer $n \geq 1$, there exists an acyclic social network SN with the following properties:*

- *SN contains the base pair and has $2n$ further sources, grouped into n dual pairs, and $2n$, grouped into n dual pairs as well, which we associate with output pairs, as well as three additional dual pairs, one of each is a sink.*
- *Every path from a source to an output pair has the same length h, independent of n;*
- *SN* simulates M. *I.e., if at time t the base pair and input dual pairs are valid and represent a configuration $s^{(0)} \in \{0,1\}^n$, then at time $t+h$ if $s^{(0)}$ is non-halting, the output dual pairs are valid and represent $s^{(1)}$, the successor configuration of $s^{(0)}$. Otherwise at least one dual pair becomes invalid.*
- *SN can be constructed in time polynomial in n and in the description of M.*

Proof. The assertion relies on the observation (following the lines of [Arora and Barak, 2009, Theorem 6.6], or [Papadimitriou, 1994, section 8.2]) that for every polynomially space-bounded Turing machine M and every integer n, there exists a Boolean circuit C, satisfying the following properties

- C has n inputs and n outputs.
- C has equal-length paths from inputs to outputs (where this length h is independent of n).
- C transforms an arbitrary non-halting configuration of M into its successor configuration.
- C can be constructed in time polynomial in n and in the description of M.

These properties map into the assertions of the lemma, using dual pairs as vertices in the circuit, and AND and NOT gadgets from Example 1 as gates. To make the network satisfy the second assertion of the lemma, we extend it using NOP gadgets where necessary.

To make sure that the third assertion holds, we include two additional dual pairs. We further assume that for every non-halting configuration of M, we include three additional pairs of vertices, i.e., $(a_1, b_1), (a_2, b_2)$, (c_1, c_2), with c_1 being influenced only by a_1, and c_2 only by b_2. If the starting configuration of M is non-halting, we assume that both of these pairs are valid and that a_1 as well as b_1 are labelled r.

Finally, using AND and NOT gadgets, and starting at the output we construct a gate influencing the pair pairs (a_1, b_1), so that a_1 and b_2 change their labels if and only if the configuration given by the output pairs at time h corresponds to a halting configuration of M. Observe that if that happens, then (c_1, c_2) becomes invalid.

□

Fuse Line, Valve, and Alarm. We will need a mechanism to check the initial validity of dual pairs in our construction, as well as to detect the halting of a Turing machine, following Lemma 7.1. If a dual pair is or becomes invalid, this will force the convergence of the social network. We stress that it is enough for one of them to become invalid for this to happen. As we will observe, this would cause a "cascading" effect, resulting in the stabilisation of the network.

The mechanism consists of a *fuse line* (sequence of pairs of vertices) leading to a *valve* and *alarm* (an even clique), as shown in Figure 7.5.

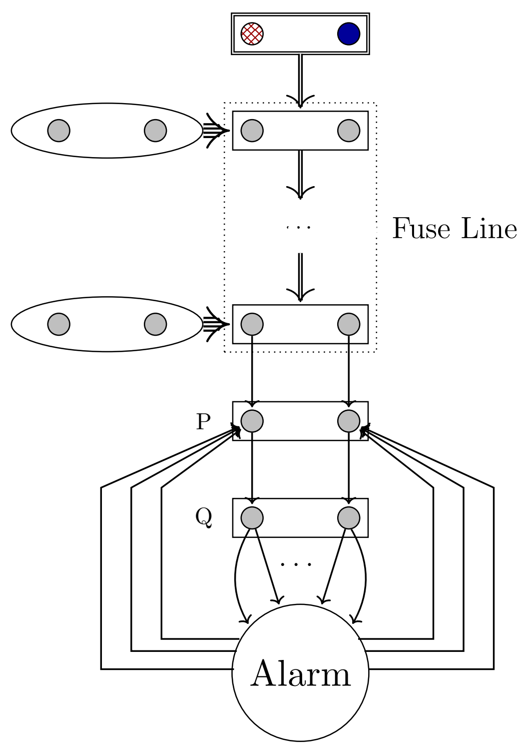

Figure 7.5: The fuse line. Recall that we assume that if the base pair is valid, then the left vertex in this pair is labelled r, and the right one is labelled b.

We first discuss the fuse line itself. Pairs of vertices in the fuse line are depicted by rectangles. Each pair in the fuse line (except for the last) feeds into the succeeding pair, as shown in Figure 7.6.

Figure 7.6: Two pairs in the fuse line, one feeding into the other. Left: in detail. Right: simplified drawing (corresponding to connections between pairs in the fuse line as depicted in Figure 7.5), abbreviating the connections in the left picture.

In addition, all other dual pairs in the entire network (depicted for the sake of clarity as ovals) will also connect to distinct pairs in the fuse line as shown in Figure 7.7. We will not think of the pairs in the fuse line as dual pairs.

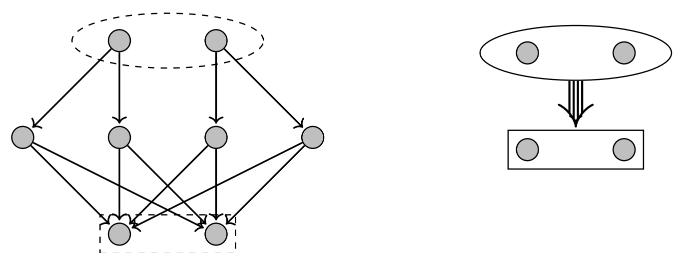

Figure 7.7: Dual pair connected to pair from the fuse line. Left: in detail. Note how the influence of the input pair on the output pair is stronger than in Figure 7.6. Right: simplified drawing (used in Figure 7.5), abbreviating the connections in the left picture.

At the end of the fuse line shown in Figure 7.5, the big circle is a clique of $2k$ vertices (an *alarm*), with $k \geq 2$, and the *valve* mechanism is formed by the two rectangles (pairs) P, Q, and the alarm. Both vertices of pair Q have edges to each vertex in the alarm, and all vertices in the alarm have edges to both vertices of pair P. In the following analysis, we say that the alarm is *evenly split* if exactly k of its vertices are labelled b. We say that the alarm *goes off* at time t if all of its vertices will have agreed by this time (we will usually imply that this was not the case at time $t-1$).

We show now several properties of this network, which will be crucial for the **PSPACE**-hardness reduction.

Lemma 7.2. *If at time $t \geq 1$ a pair p in the fuse line is invalid, then the following properties hold:*

(a) *It remains invalid forever.*

(b) *the succeeding pair is invalid from time $t + 1$ onward.*

Proof. Let us first show that the assertion (a) holds. Suppose that at the time t, a pair $p = (p_1, p_2)$ in the fuse line is invalid. Notice then that $N^{-1}(p_1) = N^{-1}(p_2)$. But this implies that since p_1 and p_2 have the same colour at t, we have that in every time $t' > t$ they either both change their label, or they both do not. Assertion (a) follows.

Further, following assertion (a), in order for assertion (b) to fail, the succeeding pair must be valid at times t and $t + 1$. Then, since the two vertices in this succeeding pair have the same set of six influencers, this set should be evenly split at time t. But this is impossible, because two of these influencers agree by the assumption of the lemma, and the remaining four cannot be split into one and three for every $t \geq 1$ by the construction of the connection in Figure 7.7. □

This entails the following fact.

Lemma 7.3. *If some dual pair in the network is invalid at some time, then the last pair in the fuse line becomes invalid at some time and remains invalid forever.*

Proof. Follows directly from Lemma 7.2. □

The final part of our construction of the network is that every vertex in the alarm has edges to every vertex in the network, except for the fuse line, vertices connecting dual pairs to pairs in the fuse line, the base pair, and pair Q of the valve. So, the alarm is connected to all of the dual pairs, as well as the pair P. This entails that when the alarm goes off, all vertices in the network eventually adopt the same value, as shown in the following lemma.

Lemma 7.4. *Suppose at time t at least one of the following conditions hold:*

(a) *The last pair in the fuse line is invalid.*

(b) *The two vertices of P agree.*

(c) *The two vertices of Q agree.*

(d) *The alarm is not evenly split.*

Then, by time $t+3$, the alarm will have gone off. Further, by time $t+6$, all vertices in the network agree.

Proof. Suppose that at some time t' the alarm is split into sets of size $m \leq k$ and $2k - m$. If $m \leq k - 2$, the influence of Q on the alarm is negligible, so the alarm goes off at time $t' + 1$. If $m = k$, then all vertices in the alarm will change their label at time $t' + 1$, if the two vertices of Q disagree. Otherwise, the alarm goes off. Finally, if $m = k - 1$, then all vertices in the alarm will change their label at time $t' + 1$ if the two vertices of Q agree and side with the minority, otherwise the alarm goes off.

We now show that, under the conditions of the lemma, the alarm will necessarily go off at some time $t' \leq t + 3$. If this does *not* happen then, by the argument above, either (1) the alarm remains evenly split (and the vertices in each of the pairs P and Q disagree), or (2) the alarm is split into sets of size $k - 1$ and $k + 1$, with vertices changing their colour in each step, and the vertices of Q keep agreeing with each other and alternating between labellings (b, b) and (r, r).

Consider scenario (1). Note that cases (b) and (c) are incompatible with this scenario, because Q copies P and influences all vertices in the alarm. Case (d) is not compatible with this scenario either. So only case (a) remains. But since in this scenario the alarm is evenly split, the pair P will copy the last pair of the fuse line, and at time $t + 1$ case (b) is true. So, in scenario (1), the alarm will go off by time $t + 3$.

We now consider scenario (2). Assume without loss of generality that at time t the alarm has $k - 1$ vertices labelled b and $k + 1$ vertices labelled r, and that Q is labelled (b, b) at this time (siding with the minority). Note that as (2) is the case, the vertices in pair Q change their colour in every step. Further, since the alarm is split into sets of size $k - 1$ and $k + 1$, the vertices of P copy the majority in the alarm. Therefore, the labellings must follow the following diagram.

	t	$t+1$	$t+2$
P		(r,r)	
Q	(b,b)	(r,r)	(b,b)
$k-1$ in the alarm	b	r	b
$k+1$ in the alarm	r	b	r

But this labelling of Q at time $t+2$ is not possible, because the pair Q simply copies the pair P. Therefore, Q will remain at (r, r) instead. So, at time $t + 3$ the alarm will go off.

It remains to prove that, in all of the cases we consider, in at most three steps from the alarm going off, all vertices in the network will agree. Since all $2k$ vertices in the alarm influence all dual pairs in the network, and the indegree of each vertex in every dual pair is at most three (excluding the edges from the alarm), the influence of the alarm will prevail as long as $k \geq 2$. This means that all dual pairs will become invalid and assume this value by time $t + 4$. All pairs in the fuse line will follow by time $t + 6$. At the same time, pairs P and Q of the valve will follow the alarm no later than at times $t + 4$ and $t + 5$, respectively. This completes the proof. □

Auxiliary Labelling $a(s)$. Let $SN_T = (N, E)$ be the social network satisfying the conditions described in Lemma 7.1, i.e., the fragment of our construction which computes subsequent configurations of a Turing machine, and let $s \in \{0, 1\}^n$ be a configuration of such a machine. We note that then SN_T is acyclic, while all of paths from source to an output pair in SN_T have equal length h. Hence, the set of all vertices on such paths in SN_T can be partitioned into $h + 1$ layers $(l_0, \ldots, l_h)$, where l_0 is the source layer and layer l_h is the layer containing the output pairs. In other words, vertices in l_0 have no influencers, while those in a layer l_i, with $i > 0$, only have influencers in the layer l_{i-1}. Let us now denote by $SN'_T = (N', E')$ the social network obtained from SN_T by removing pairs outside of $\bigcup_{i \in [0,h-1]} l_i$. So, $N' = \bigcup_{i \in [0,h-1]} l_i$, and E' is the restriction of E to N'. We now define the labelling $a(s)$ of SN'_T as follows, noticing that every dual pair is contained in one layer only. Consider first any labelling of SN'_T, and let the n dual pairs in layer 0 be assigned the values that represent s. Observe now the network SN_T converges after a constant number of updates by Proposition 7.1. We then pick as $a(s)$ the labelling of the limit network, i.e., the fixpoint of the sequence $SN'_{T_{us}}$.

Construction of Network *MN* and Labelling f. We construct a social network from the components described above. Given a Turing machine M, we take the network *SN* satisfying the conditions described in Lemma 7.1 and combine it with the fuse line, valve, and alarm as follows:

- For each $i = 1, \ldots, n$, the i^{th} source dual pair of *SN* is identified with the i^{th} output dual pair of *SN*, which transforms *SN* into a cyclic network, where all cycles have length divisible by h.
- Every dual pair in *SN* connects to a distinct pair in the fuse line as described above.
- Every vertex in the alarm has edges to all dual pairs in *SN*, except for the fuse

line and pair Q of the valve (in other words, to all dual pairs and to pair P), as described above.

Notice that the fuse line needs as many pairs as there are dual pairs in *SN*, and that k can be chosen as 2 (based on the proof of Lemma 7.4). This completes the construction of the network *MN* which we use in our reductions.

Given a configuration $s \in \{0,1\}^n$ of the Turing machine M, consider any labelling of *MN* that satisfies the following conditions:

1. Vertices in *SN* are labelled according to the auxiliary labelling $a(s)$ defined above.
2. Each pair in the fuse line and the valve is valid (i.e., its nodes disagree).
3. The alarm is evenly split.
4. In every connection of the form shown in Figure 7.7, exactly two out of four intermediate vertices have value b.

We denote this labelling by f.

7.3.2 Hardness proofs

We proceed to proving the computational hardness of the problems for Theorems 7.1 and 7.2.

Proof of Theorem 7.1. We have already argued membership of these problems in **PSPACE** above and we will prove its hardness here. We rely on the fact that there exists a universal, polynomial-space[1] Turing machine U, for which the following problem is **PSPACE**-complete. This statement follows from the **PSPACE**-completeness of In-place Acceptance (see Theorem 19.9 in Papadimitriou [1994]).

Input: An integer $n \geq 1$ and a configuration $s^{(0)} \in \{0,1\}^n$ of U.
Question: Does U diverge when started from configuration $s^{(0)}$?

We now apply the construction above to the Turing machine U. Take the network *MN* and the labelling f defined above. First note that dual pairs in *SN* have inputs from inside *SN* and $2k$ inputs from the alarm. This means that *SN* will function "autonomously" as long as the alarm remains evenly split. By Lemma 7.1, *SN* will in this case compute consecutive configurations of the Turing machine U.

[1] A universal Turing machine simulates the computation of any Turing machine, on an arbitrarily chosen input.

Indeed, observe that the labelling of the source level of SN will be set to $s^{(0)}$ at time 0. Then, at times $1, \ldots, h$ it will correspond to to $s^{(1)}$, i.e, the successor of $s^{(0)}$. The analagous transformation takes place between every configuration $s^{(i)}$ and $s^{(1+1)}$.

Observe that if the Turing machine U diverges when started from the configuration $s^{(0)}$ then, by the observations made above, the alarm will always remain evenly split, with vertices changing their colour in every step. This means that MN does not converge. On the other hand, if U terminates, then some dual pair will become invalid (Lemma 7.1), the alarm will go off (Lemmata 7.3 and 7.4), and the network will converge. Theorem 7.1 follows.

Proof of Theorem 7.2. Again, we already argued membership in **PSPACE** above and will prove hardness here. We will now rely on **PSPACE**-completeness of the following problem:

> *Input:* Integer $n \geq 1$ and (a description of) a Turing machine M.
> *Question:* Is there a configuration $s \in \{0,1\}^n$ such that M diverges when started from s?

The hardness of this problem is a variation of the Corollary of Theorem 19.9 in Papadimitriou [1994].

The proof of Theorem 7.2 extends the proof of Theorem 7.1. Instead of U, we now have any polynomial-space Turing machine M. Recall from the previous proof that if there is a configuration $s \in \{0,1\}^n$ from which M diverges, then there is an initial labelling from which MN fails to converge. So we will now consider the case where M terminates starting from every configuration. Let us now determine whether there is a labelling from which MN fails to converge.

Let us consider an arbitrary labelling g of MN. By Lemma 7.4, if there exists a time $t \in \{0, 1, \ldots, h-1\}$, for which the network MN has an invalid dual pair, then MN converges. The same holds if MN has an invalid pair in the fuse line or valve, or if the alarm is not evenly split. Suppose now that none of the above applies. Then consider configurations $s_0, \ldots, s_{h-1} \in \{0,1\}^n$ formed by the values of the source-layer dual pairs of SN at times $0, 1, \ldots, h-1$. By the arguments above, the network MN simulates the Turing machine M in the following way. For each $i \in \{0, 1, \ldots, h-1\}$, at times $t \in \{i, i+h, i+2h, \ldots\}$ the source-layer dual pairs of SN form consecutive configurations of M starting from s_i. If M terminates when started from some $s' \in \{s_0, \ldots, s_{h-1}\}$, then MN converges when started from the labelling g. This means that a necessary condition for MN to fail to converge (starting from

g) is that M diverges when started from every s_i, $i \in \{0, 1, \ldots, h-1\}$. In this case, there exists a configuration s_i from which the Turing machine M diverges. This completes the proof of Theorem 7.2.

7.4 Conclusion

In this chapter we studied the problem of convergence of the opinion diffusion protocol, in which all agents synchronously change their binary opinion if the strict majority of their neighbours disagrees with them.

Summary of Contributions. We have shown that checking convergence of opinion diffusion in social networks is **PSPACE**-complete. In particular, we have shown that the problems of checking if a network with a given labelling converges (Theorem 7.1), and of determining if it admits any converging colouring (Theorem 7.2) are intractable. We note that our results extend to majority-based multi-issue opinion diffusion (see Grandi et al. [2015]), also in presence of integrity constraints (Botan et al. [2019]), and to all update rules that admit suitable modification of our gadgets, such as quota rules, in which an agent switches an opinion if a specified fraction of their influencers disagrees with them.

Future Research. The results shown in this chapter open many possible directions for further research. Let us mention a few of them.

- In this chapter we have only investigated a limited number of cases for which the problem of convergence of the opinion diffusion protocol which we considered is easy to solve. Hence, as our results imply that there is no efficiently computable characterisation of convergent networks, the problem of identifying classes of networks which always converge is an appealing open problem. In particular, complexity of checking convergence in cases resembling real-life social networks would be of high interest.

- We have shown that it is intractable to check if a network converges for a given labelling. It remains open, however, whether the existence of a *non-trivial* (i.e., different from all-red and all-blue) fixed-point configuration in our model is an **NP**-complete property.

- We have limited ourselves to the study of synchronous opinion diffusion protocols. This is possibly the simplest social network update model, widely adopted in the literature. However, we have not investigated the *asynchronous* protocols, in which subsets of agents change their opinions in each step. We note that in case of many protocols losing synchronicity makes the system nondeterministic, so the question of convergence changes significantly.

- Finally, we note that our results are based on a worst-case complexity analysis and an important question remains regarding the complexity of verifying convergence in random networks, or networks based on real-world data.

Chapter 8

Majority Illusion

8.1 Introduction

Social networks shape the way people think. Individuals' private opinions can change as a result of social influence and a well-placed minority view can become what most people come to believe (Stewart et al. [2019]). It is worth noting that the COVID-19 vaccination debate has brought to the fore the dramatic effects that misperception can have in people's lives (Johnson et al. [2020]) and highlighted the importance of social networks, where participants receive the most unbiased information possible.

When individuals use their social network as a source of information, it may be the case that minority groups are more "visible" as a result of being better placed. This makes them over-represented, and even appear to be majorities in many friendship groups – a phenomenon known as *majority illusion*. Majority illusion was originally introduced by Lerman et al., who studied the existence of social networks in which most agents belong to a certain binary type, but most of their peers belong to a different one. Thus, they acquire the wrong perception, i.e., the illusion, that the majority type is different from the actual one. Figure 8.1 shows an example of this.

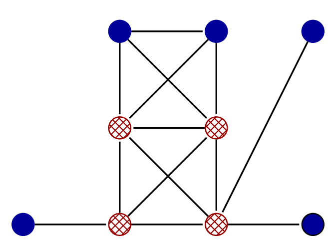

Figure 8.1: An instance of majority illusion. The well-placed red (shaded) minority is perceived as majority by everyone.

Majority illusion has important consequences when paired with opinion formation. If, for example, individuals change their mind based on what their friends say, e.g., they follow a threshold model (Granovetter [1978]), then majority illusion means that strategically placed minorities may well become stable majorities. Furthermore, we note that majority illusion can have negative repercussions outside of the context of opinion diffusion. For instance Santos et al. [2021] show how such phenomenon can skew public good games towards unwanted outcomes. As such, it is important to predict its occurrence in a network and, crucially, to see to it that this undesirable phenomenon is eliminated.

The graph structure of majority illusion was analysed by Lerman et al., who studied network features that correlate with having many individuals under illusion. They demonstrated how disassortative networks, i.e. those in which highly connected agents tend to link with lowly connected ones, increase the chances of majority illusion. However, no algorithms have yet been provided to check whether majority illusion can occur in a social network.

Likewise, the approach of eliminating undesirable properties by network transformation is not new, and extensively pursued in the context of election manipulation (see, e.g., Castiglioni et al. [2021]), influence maximisation (Zhou and Zhang [2021]), anonymisation (see, e.g., Kapron et al. [2011]) and of k-core maximisation (see, e.g., Chitnis and Talmon [2018] and Zhou et al. [2019]). However, such natural operations have yet to be studied in the context of eliminating majority illusion.

All in all, the computational questions of checking whether a network admits majority illusion, and how this can be eliminated, are still unexplored.

Our Contribution. In this chapter we initiate the algorithmic analysis of majority illusion in social networks, focusing on two computational questions. We first consider the problem of verifying illusion, i.e., deciding whether there is a labelling of the vertices such that a set majoritarian fraction of agents are under illusion, and we prove it to be **NP**-complete. Our **NP**-hardness proof techniques also imply **NP**-hardness on bipartite networks, planar networks, networks where the maximum degree is bounded by a constant, and networks of constant c-closure.

In light of these negative results, we aim to identify tractable restrictions of the problem by carrying out a parametrised complexity analysis involving well-established graph width measures and their variants. We note that the study of parametrised complexity with respect to social networks analysis is a direction present in the literature. For instance, Bredereck and Elkind [2017] use parametrisation by tree width in the context of manipulating a majority opinion in synchronous majority dynamics, where a fixed point is not guaranteed to exist. In particular, we obtain a fixed-parameter algorithm (FPT algorithm) for verifying illusion, parametrised by the maximum degree of the network plus its tree width, as well as by the size of the minimum vertex cover. Along the way, we show that for every constant value of the network's tree width, the problem can be solved in polynomial time (i.e., an XP algorithm parametrised by the tree width). These two results are of specific interest to sparse networks. We then also consider dense networks by parameterising by the neighbourhood diversity of the input network and obtain an FPT algorithm. Finally we move to the problem of eliminating illusion, which we model as edge transformation by bounded Hamming distance. We show this problem to be **NP**-complete in general and **W**[1]-hard when parametrised by arguably the most natural parameter, i.e., the number of modified edges.

Structure of the Chapter. In Section 8.2, we focus on checking whether illusion can occur in a network. Further, Section 8.3 studies illusion elimination. Also, in Section 8.4, we provide an example of a network in which it is possible to have that all agents observe as the most popular a different label, than the most popular one in the network, even though they cannot all be under the majority illusion. Finally, Section 8.5 concludes the chapter, presenting various potential future directions.

8.2 Verifying Illusion

We are interested in the computational problem of checking, for a rational q, if a given network admits q-majority illusion. Formally, we will study the following problem.

> q-MAJORITY ILLUSION:
> *Input:* Social network (N, E).
> *Question:* Is there a labelling $f : N \to \{b, r\}$, such that f induces a q-majority illusion?

8.2.1 Hardness

Observe that q-MAJORITY ILLUSION is in **NP** for every q, since verifying if a labelling induces a q-majority illusion can be done by checking, for every vertex i, if i is under illusion. We now prove that q-MAJORITY ILLUSION is **NP**-hard for every rational $q \in (\frac{1}{2}, 1]$, by providing a reduction from the **NP**-hard problem 2P2N-3-SAT for every such q.

Let φ be a formula in 2P2N-3-CNF. We will construct an instance of q-majority illusion, for which the answer is positive if and only if φ is satisfiable. We commence with constructing a social network, which we call the *encoding* of φ, or $E_\varphi = (N, E)$. We will further show that it admits 1-majority illusion if and only if φ is satisfiable, entailing the **NP**-hardness of 1-majority illusion. We show that in Lemma 8.3. Finally, for each $q \in (\frac{1}{2}, 1]$, we construct a network E^q_φ, which we obtain by appending a non-trivial network construction to E_φ. We then conclude the proof, by showing in Theorem 8.1 that E^q_φ admits a q-majority illusion if and only if φ is satisfiable, using the fact that increasing the blue surplus in E_φ results in an increased number of vertices under illusion (see Lemma 8.4), and a technical Lemma 8.5.

Variable, Clause and Balance Gadgets. For a formula φ in 2P2N-3-CNF, we denote the set of variables in φ as $X = \{x_1, \ldots, x_m\}$, and the set of clauses in φ as $C = \{C_1, \ldots, C_n\}$. Let us first encode propositional variables. For a variable x_i, we define a subnetwork, which we call a *variable gadget*, as depicted in Figure 8.2. There, the left subfigure presents what we call a *filling structure*.

The left part of this structure is a complete bipartite network $K_{3,3}$, while the right consists of seven vertices. We assume that there are at least six such vertices that are dependents of vertices in the complete network in the same filling structure. If it is the case for all seven of these vertices, then we assume that the vertex in the $K_{3,3}$ in this filling structure, depicted as top-right in Figure 8.2, has two dependents. We further assume that each of the vertices in the complete network in a filling structure has a dependent. Then, the variable gadget contains three copies of the filling structure. In two of them, seven vertices are a dependent of a vertex in the same filling structure.

Furthermore, a variable gadget contains three additional vertices, connected as shown on the right side of Figure 8.2. We refer to two top-left vertices in the right side of Figure 8.2 as *literal vertices*. Further, we say that upper literal vertex corresponds to x_i, and the lower literal vertex corresponds to $\neg x_i$. Finally, in the third filling structure, one of the vertices is a dependent of the upper-right vertex in the right side of Figure 8.2, which is further adjacent to two vertices in the complete bipartite network in this filling structure.

The following lemma shows that it is necessary for exactly one of the literal vertices in a variable gadget to be labelled r in a labelling of this structure, which induces 1-majority illusion. This observation will be crucial in demonstrating that a labelling of the encoding of φ, in which all vertices in variable gadgets are under

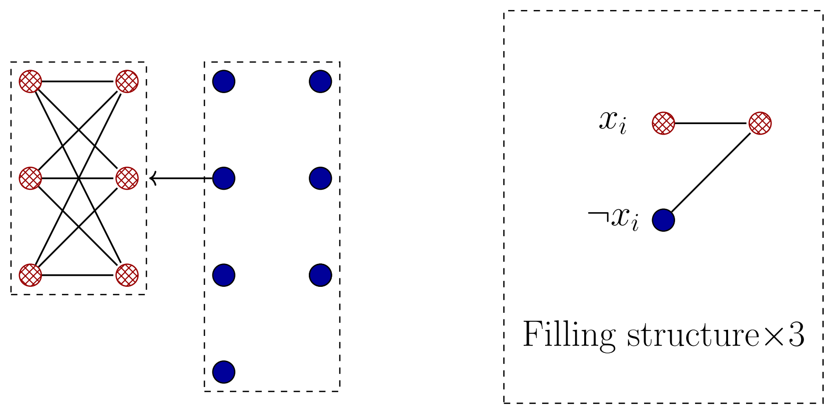

Figure 8.2: Filling structure with a unique labelling such that all members are under illusion on the left, labelling of type A on the right. Note that the vertex, which is adjacent to literal vertices, is further adjacent to two vertices in a $K_{3,3}$ in a filling structure, and has one dependent.

illusion, corresponds to a valuation over X.

Lemma 8.1. *A labelling of a variable gadget (considered as a separate network) induces a 1-majority illusion only if at most one of the literal vertices is labelled r.*

Proof. Take a variable gadget V_i, as defined above. Also, suppose that there is a labelling f of V_i, which induces 1-majority illusion. Let us begin by observing that all vertices in the complete networks in filling structures are labelled r in f. This observation holds, as each of them has a dependent. Notice further that since the vertex adjacent to literal vertices has five neighbours, we have that at least three of them are coloured r in f.

We will now show that it cannot be the case that both literal vertices are labelled r in f. Let us first observe that the vertex adjacent to the literal vertices needs to be labelled r in f, as it has a dependent. Then, observe that there are fourty-two vertices in a variable gadget. Thus, in a at most twenty vertices can be labelled r in f, as we assume that the strict majority colour in f is b. As we observed, at least eighteen of the vertices in filling structures need to be labelled r in such a labelling of this gadget, as they have dependents. Similarly, the unique vertex in the gadget, which is adjacent to the literal vertices needs to be labelled r in f, as it has a dependent. Hence, at most one of the literal vertices can be labelled r in this labelling. It follows that f induces a 1-majority illusion only if at most one of the literal vertices is labelled r. □

It is worth noting that there are two labellings of a variable gadget (as a separate network), which admit 1-majority illusion. In one of them, the vertex corresponding to x_i is labelled r (we say that such a labelling is of *type* A). In the second, the vertex corresponding to $\neg x_i$ is labelled r (we say that such a labelling is of *type* B). Consider now a labelled network (N_1, E_1, f), where a variable gadget V_i is a subnetwork. We note that if the margin of victory is 1 in $(N_1 \setminus V(V_i), E \setminus E(V_i), f')$, where f' is such that, for every $i \in N \setminus V(V_i)$, we have that $f(i) = f'(i)$, then f induces 1-majority illusion only if V_i if labelled in type A or type B, but not if both literal vertices are labelled r.

We further introduce, for each clause $C_i \in C$, what we call a *clause gadget* corresponding to C_i, as depicted in Figure 8.3. The three vertices in the right side of this figure do not belong to the clause gadget. Instead, they correspond to the literals in C_i. Then, we call the top vertex in the middle of the Figure 8.3 the *verifier vertex* for C_i. Then, the gadget includes five filling structures, such that in one of them seven vertices are dependents of a member of the same filling structure. Further, there are four vertices outside of filling structures in this gadget, such that each of them has a dependent in some filling structure in the same gadget. Also, the verifier vertex is adjacent to one vertex in a complete network of some filling structure. Observe that as all of the vertices in complete networks in filling structures have a dependent, it holds that in a labelling of a clause gadget gadget which induces 1-majority illusion, they are all labelled r.

We will now show that a labelling of a clause gadget can only induce 1-majority illusion if at least one of the adjacent literal vertices is labelled r. This fact will later allow us to show that the fact that all vertices in a clause gadget for C_i are under illusion for some labelling of an encoding means that C_i is satisfied in a valuation.

Lemma 8.2. *There exists a labelling of a clause gadget (not as a separate network), which induces 1-majority illusion with blue being a majority winner in this structure, if and only if at least one vertex is adjacent to a literal vertex labelled r in this structure.*

Proof. Take some clause gadget, and call it G. Let us first observe that there are sixty-nine vertices in G, i.e., thirteen vertices in each of five filling structures and four additional vertices. Let us further observe that, for every filling structure F, it holds that in every labelling f of F, which induces 1-majority illusion, at least six members, which have dependents, are labelled r. Further, by previous observations, it holds that all vertices outside of filling structures are labelled r in f.

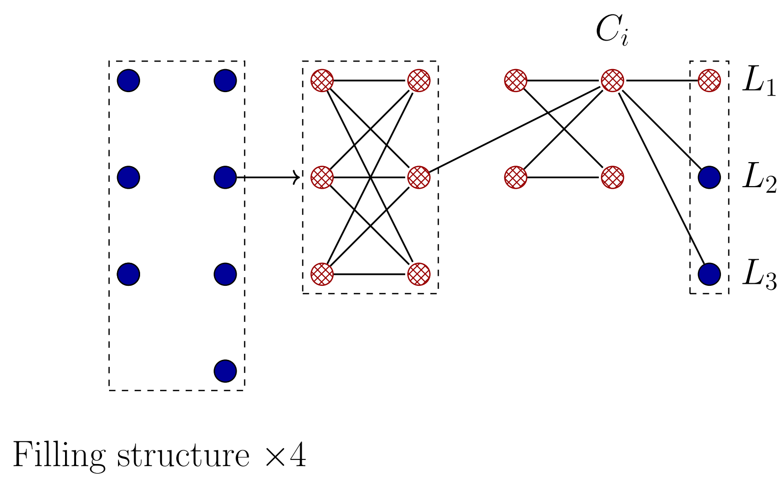

Figure 8.3: Clause gadget with a unique labelling such that all members are under illusion. Note that each of the vertices in the gadget outside of filling structures have a dependent, while the verifier vertex is also adjacent to one of the vertices in a $K_{3,3}$

Let us now show f exists, if at least one literal vertex, adjacent to the verifier vertex in G, is labelled r. Suppose that this is the case. Then, let us construct such a labelling f. First, let us label r all members of the complete networks in filling structures, as well as all of the vertices in G, which are not in filling structures. Further, let all other vertices in the gadget be labelled b. Observe that in this labelling, thirty-four vertices, i.e., six vertices in each of the five filling structures, as well as four additional vertices are labelled r. The remaining thirty-five, i.e., seven vertices in each filling structure, are labelled b. Observe further that this implies that all nodes, which are a dependent, are labelled b in f. Notice that then all of the vertices in G, other than the verifier vertex, are under majority illusion. Notice now that, by assumption, one of the literal vertices adjacent to the verifier vertex is labelled r. Hence, the verifier vertex is under illusion, given the proposed labelling. So, the claim holds.

Suppose now, towards contradiction, that all literal vertices adjacent to the verifier vertex are labelled b, but there is a labelling f of G that induces 1-majority illusion. Then, it follows by previous reasoning, that the vertex which is a dependent of the verifier vertex is labelled b in f. Notice, however, that then the verifier vertex is not under illusion, as four out of seven of its neighbours are labelled b, which contradicts the assumptions.

□

Finally, for $k \geqslant 2$, we define what we call a *balance gadget*. If k is even, then the balance gadget is the collection of $\frac{k}{2}$ pairs of vertices, which are disconnected from the rest of the vertices in the encoding. Otherwise, we construct five vertices, such that four of them form a bipartite complete graph $K_{2,2}$, while the fifth is a dependent of one of the other vertices, as well as the balance gadget for $k-3$, if $k \geqslant 5$. Observe that the balance gadget is bipartite, and that, for every labelling of this gadget, which induces 1-majority illusion (not as a separate network), at most one vertex in this structure is labelled b.

Encoding of a 2P2N-3-CNF Formula. We are now ready to construct a social network E_φ, which *encodes* φ. First, for every variable $x \in X$, let us construct a variable gadget, as depicted in Figure 8.2. Further, for every clause $C_i \in C$, i.e., $\{L_i^1, L_i^2, L_i^3\}$, let us create a clause gadget, as shown in Figure 8.3, with literal vertices corresponding to L_i^1, L_i^2, and L_i^3 being adjacent to the verifier vertex in the clause gadget corresponding to C_i. As a final step, let us construct a balance gadget for $k = 2m + n - 1$, which by construction is always greater or equal than two (this is because there is at least one variable and one clause in φ).

Observe that, since there are $2m + n - 1$ vertices in the balance gadget, if k is even, and $2m + n + 4$ vertices otherwise. Recall that we have $42m$ vertices in variable gadgets, and $69n$ vertices in clause gadgets. Hence, there are $44m + 70n - 1$, or $44m + 70n + 4$, vertices in E_φ. Let us further notice that, following previous observations, for every labelling of E_φ, which induces 1-majority illusion, and for every variable gadget (consisting of fourty-two vertices), at least twenty of its members are labelled r. Similarly, in such a labelling, for every clause gadget, we have that at least thirty-four out of sixty-nine members of the gadget are labelled r. Finally, by previous observations, all vertices in the balance gadget are labelled r, if k is even, and $2m + n + 4$ vertices otherwise. We note that these observations imply that in every labelling f of E_φ, which induces 1-majority illusion, it holds that $\#B_f - \#R_f = 1$.

Lemma 8.3. *Let φ be a formula in 2P2N-3-CNF. Then, φ is satisfiable if and only if E_φ admits 1-majority illusion.*

Proof. Let us consider a formula φ in 2P2N-3-CNF, with the set of variables $X = \{x_1, \ldots, m\}$, and the set of clauses $C_\varphi = \{C_1, \ldots, C_n\}$. Then, we will construct the encoding E_φ of φ, and show that it admits 1-majority illusion if and only if φ is satisfiable.

Let us first suppose that it is. Then, take a model M of φ and label E_φ as follows. First, colour variable gadgets, so that, for every such gadget corresponding to a variable x_i, it is of type A if x_i is true in M, and of type B otherwise. Then, observe that, by previous observations on the construction of E_φ, we have that every verifier vertex in E_φ is adjacent to some literal vertex which is labelled r, as all clauses are satisfied under M. Hence, following previous observations, E_φ admits 1-majority illusion.

Suppose now that φ is not satisfiable. Then, observe that, following Lemma 8.1, every labelling of E_φ that admits a 1-majority illusion requires variable gadgets not to have both literal vertices labelled r. Further, as φ is not satisfiable, it holds that at least one verifier vertex would need to be adjacent to three literal vertices labelled b. But then, it would not be under majority illusion, which contradicts the assumptions. Hence, E_φ admits 1-majority illusion if and only if φ is satisfiable. □

We now show some further properties of E_φ. We will henceforth assume, for simplicity, that $k = 2m + n - 1$ is even. The subsequent claims can be shown for odd k similarly. Given a 2P2N-3-CNF formula φ, let $I_\varphi = 22m + 35n - 1$, where m is the number of variables and n the number of clauses in φ. Observe that this is the maximum number of vertices which can be labelled red in E_φ, if blue is the strict majority colour in this network.

Lemma 8.4. *For every 2P2N-3-CNF formula φ, $k \leqslant I_\varphi$ and any labelling f of $E_\varphi = (N, E)$, such that $R_f = I_\varphi - k$, the number of vertices under illusion in E_φ under f is at most $\#N - k$.*

Proof. Consider a formula φ in 2P2N-3-CNF and a natural $k < I_\varphi$, as well as a labelling f of $E_\varphi = (N, E)$, such that $R_f = I_\varphi - k$. We will show that the number of vertices in E_φ, which are not under illusion given f, is at most $\#N_\varphi - k$.

Let us denote as A the set of all vertices in complete bipartite graphs in filling structures, and all of the vertices in variable gadgets, which are not literal vertices. Then, let B' be the set of all vertices in clause gadgets, which are not in filling structures. Further, let C be the set of literal vertices, and D be the set of vertices in the balance gadget. Finally, let E' be the set of all remaining vertices in N. Observe now that $A \cup B' \cup C \cup D \cup E' = N$. Moreover, by construction, we have that $\#A + \#B' + \frac{\#C}{2} + \#D = I_\varphi$.

We further show some crucial properties of A, B', C, and D. Observe now that each vertex in A has a dependent. Hence, there exists a set $N_A \subseteq E'$, such that for every $i \in N_A$, we have that i is not under illusion, while $\#N_A = \#B^A$. Similarly, for every $i \in B'$, we have that i has a dependent. Hence, there is a set $N_{B'} \subseteq E'$, such that, for each $j \in N_{B'}$, we have that j is not under illusion, while $\#N_{B'} = \#B^{B'}$. Let further $M_C = \frac{\#C}{2} - \#B^C$ if $\frac{\#C}{2} - \#B^C > 0$, and 0 otherwise. Notice that, by construction, there is a set $N_C \subseteq A$, such that, for every $i \in N_C$, i is not under illusion, while $\#N_C = M_C$. Finally, notice that, by construction of a balance gadget, there is a set $N_D \subseteq D$, such that, for every $i \in D$, we have that i is not under illusion, while $\#N_D \geqslant \#B^D$. Let us also observe that N_A, $N_{B'}$, N_C, and N_D are pairwise disjoint.

We are now ready to show that at least k are not under illusion under f. Notice that $\#A + \frac{\#B'}{2} + \#C + \#D = I_\varphi$, and that at most I_φ vertices are labelled r in f, as otherwise b would not be the strict majority colour. But then, at least k vertices are labelled b in $A \cup B' \cup C \cup D$. This implies, however, that $N_A + N_{B'} + N_C + N_D \geqslant k$, and hence at least k vertices are not under illusion under f. □

Observe also that, by the reasoning similar to the proof of Lemma 8.4, we also get that, for a labelling f of E_φ, which maximizes the number of vertices under illusion (which we call M), $k \leqslant I_\varphi$ and any labelling f' of $E_\varphi = (N, E)$, such that $R_{f'} = I_\varphi - k$, the number of vertices under illusion in E_φ under f' is at most $M - k$.

We further need the following technical lemma.

Lemma 8.5. *Let q be a rational number in $(\frac{1}{2}, 1]$, and $k > 0$ be a natural number. Then, there exists a natural number h^* such that $\frac{k+h^*}{k+2h^*} \geqslant q$, but $\frac{k+h^*-1}{k+2h^*} < q$.*

Proof. Take a $k > 0$, and a fraction $\frac{a}{b} \cap (\frac{1}{2}, 1]$. Observe that if $\frac{a}{b} = 1$, then the claim holds immediately. So, we will only consider fractions such that $\frac{a}{b} < 1$. Then, we define a function $f : \mathbb{N} \to \mathbb{Q}$ such that, for a natural h, we have that $f(h) = \frac{k+h}{k+2h}$. Observe first that $f(0) = 1$. Also, observe that f is strictly downwards monotone, and is bounded by $\frac{1}{2}$. But then, as $q \in (\frac{1}{2}, 1]$, there needs to exist a maximal h such that $f(h) \geqslant q$, and as f is strictly downwards monotone, $f(h+1) < q$. We denote such a number as h^*.

Notice now that (1) $\frac{k+h^*}{k+2h^*} \geqslant \frac{a}{b}$, by definition of h^*. Note further that if $\frac{k+h^*}{k+2h^*} = \frac{a}{b}$, then the claim holds immediately. Let us assume then that $\frac{k+h^*}{k+2h^*} > \frac{a}{b}$. Further, suppose towards contradiction that (2) $\frac{k+h^*-1}{k+2h^*} \geqslant \frac{a}{b}$. Also observe that, by definition of h^*, we have that (3) $\frac{k+h^*+1}{k+2h^*+2} < \frac{a}{b}$.

Now, from (1) we get that $b(k+h^*) \geqslant a(k+2h^*)$. Also, from (2) we get that $b(k+h^*-1) \geqslant a(k+2h^*)$, which is equivalent to $bk+bh^*-b \geqslant ak+2ah^*$, and also to $-bk-bh^*+b \leqslant -ak-2ah^*$. We denote this inequality as α. Also, from (3) we have $b(k+h^*+1) < a(k+2h^*+2)$, which is equivalent to $bk+bh^*+b < ak+2ah^*+2a$.

We denote this inequality as β By adding α and β, we get that $2b \leqslant 2a$, which is impossible since $a < b$. $\square$

We refer to such a number as $h^*_{k,q}$. It is not difficult to show that we can compute $h^*_{k,q}$ in polynomial time. This observation is crucial to ensure that the intended reduction is constructable in polynomial time.

We are now ready to prove the **NP**-hardness of q-MAJORITY ILLUSION, for each $q \in (\frac{1}{2}, 1]$. Towards this end, we construct a network E^q_φ, for every formula φ in 2P2N-3-CNF and a such a fraction q. We start with constructing E_φ, and a set of $h^*_{\#V(E_\varphi),q}$ pairs of vertices. Then, it follows from Lemma 8.3, as well as Lemmata 8.4 and 8.5, that E^q_φ admits q-majority illusion if and only if φ is satisfiable. Below we include the full proof of this claim. Observe further that q-MAJORITY ILLUSION is in NP, as one can easily check the number of vertices under illusion in a labelled network.

Theorem 8.1. *q-MAJORITY ILLUSION is **NP**-complete for every rational q in $(\frac{1}{2}, 1]$, even for bipartite networks.*

Proof. Take any rational q in $(\frac{1}{2}, 1]$. We will now show that it is **NP**-hard by reduction from *2P2N-3-SAT*.

Consider a 2P2N-3-CNF formula φ with the set $X = \{x_1, \ldots, x_m\}$ of variables, and the set $C = \{C_1, \ldots, C_n\}$ of clauses. Let us construct what we call a *q-encoding* of φ. First, let E_φ be a subnetwork of the q-encoding of φ. Moreover, construct $h^*_{\#V(E_\varphi),q}$ pairs of vertices, such that vertices in each such pair are connected to each other, but not to any other vertex in the network. We call this set of pairs H. Observe further that, following previous observations, the q-encoding of φ can be constructed in polynomial time. Also, by Lemma 8.3 and Lemma 8.5, the q-encoding of φ admits q-majority illusion, if at least $\#V(E_\varphi) + h^*_{\#V(E_\varphi),q}$ vertices are under illusion in f.

Let us show now that the q-encoding of φ admits q-majority illusion if and only if φ is satisfiable. First, suppose that φ is satisfiable. Observe further that, as φ is satisfiable, by Lemma 8.3, we have that E_φ admits 1-majority illusion as a separate network. Hence, there is a labelling of the q-encoding of φ, such that exactly I_φ vertices in E_φ, as well as one of vertices in each pairs in H, are labelled red, while $\#V(E_\varphi) + h^*_{\#V(E_\varphi),q}$ vertices are under illusion. Hence, the q-encoding of φ admits q-majority illusion.

Suppose now that φ is not satisfiable. Then, suppose towards contradiction that there is a labelling f of the q-encoding of φ, which induces q-majority illusion. Let us first observe that if less than $h^*_{\#V(E_\varphi),q}$ are labelled red in H, then f does not induce q-majority illusion. Indeed, if it was the case, then less than $h^*_{\#V(E_\varphi),q}$ vertices in H would be under illusion, and hence the number of vertices under illusion in the q-encoding of φ would be strictly smaller than $\#V(E_\varphi) + h^*_{\#V(E_\varphi),q}$. But then, as f induces q-majority illusion, at least $h^*_{\#V(E_\varphi),q}$ are labelled red in H. So, the number of vertices labelled red in E_φ is smaller or equal to I_φ. If it is equal to I_φ, then the number of vertices under illusion in H is $h^*_{\#V(E_\varphi),q}$. But, as φ is not satisfiable, not all members of E_φ are under illusion, and hence f does not induce q-majority illusion. Now, suppose that less than I_φ vertices are labelled red in E_φ. Let $k = I_\varphi - \#R^{V(E_\varphi)}$. Further, let us denote as M the maximum number of vertices under illusion in E_φ, if I_φ vertices are labelled red in this subnetwork. Now, by Lemma 8.4, we have that the number of vertices under illusion in E_φ is at most $I_\varphi - k$. But then, the number of vertices labelled red in H is at most $h^*_{\#V(E_\varphi),q} + k$, and hence the number of vertices under illusion in the q-encoding of φ is at most $M - k + h^*_{\#V(E_\varphi),q}$, which is smaller than $\#V(E_\varphi) + h^*_{\#V(E_\varphi),q}$ since $M < \#V(E_\varphi)$. □

Moreover, by inspecting all pairs of vertices in the construction in the proof of Theorem 8.1, we get that q-MAJORITY ILLUSION is **NP**-complete also for networks in which minimum c-closure is bounded by a constant.

Observation 8.1. *q-MAJORITY ILLUSION is* **NP***-complete for every rational q in $(\frac{1}{2}, 1]$, even for networks with minimum c-closure bounded by 3.*

Proof. Let us show that claim holds, by demonstrating that minimum c-closure of E_φ is at most 3. We will show that, for every pair of vertices i, j in E_φ, we have that if $\#N(i) \cap N(j) \geqslant 3$, then i and j are adjacent. We consider the following, exhaustive cases. (1) i and j are both in the same variable gadget, (2) i and j are both in the same clause gadget, (3) i is in some variable gadget while j is in some clause gadget, (4) i and j belong to distinct variable gadgets, (5) i and j belong to distinct clause gadgets, and (6) i and j are in the balance gadget. Observe that in all other cases i and j do not have neighbours in common.

If (1) is the case, observe that if i and j are in the filling structure, then either they are adjacent, or $\#N(i) \cap N(j) \leqslant 2$. Hence, the claim holds. Similarly, if they both belong to additional three vertices in the gadget, then $\#N(i) \cap (j) \leqslant 2$, since we assume that a literal does not appear more than twice in a formula. If (2) is the case, then the claim holds by similar reasoning. Further, if (3) is the case, then notice that i and j have at most two neighbours in common. Also, if (4) is the case, then the only neighbours that i and j can have in common are verifier vertices. Notice further that $\#N(i) \cap N(j) \leqslant 2$ as φ is in 2P2N-3-CNF. Moreover, if (5) holds, then the only vertices that i and j have in common are literal vertices. But then, we have that $\#N(i) \cap N(j) \leqslant 3$, as the size of the clauses in φ is limited by three. Finally, if (6) is the case, then i and j have at most two neighbours in common, so the claim follows as well.

□

Furthermore, again by examining the reduction used in the proof of Theorem 8.1, we get that q-MAJORITY ILLUSION is **NP**-complete even if the maximum degree of a vertex in a network is bounded by a constant.

Observation 8.2. *q-MAJORITY ILLUSION is **NP**-complete for every rational q in $(\frac{1}{2}, 1]$, even for networks with maximum degree bounded by 6.*

Proof. Let us show that the claim holds by demonstrating that, for a formula φ in 2P2N-3-CNF, no vertex in E_φ has the degree greater than 6. To see that, take any vertex i in E_φ. Let us examine the following exhaustive cases. (1) i is in a filling structure, (2) i is in a variable gadget, but is not a literal vertex and is not in a filling structure, (3) i is a literal vertex, (4) i is in the balance gadget, (5) i is in a clause gadget, but is not a verifier vertex and is not in a filling structure, (6) i is a verifier vertex.

Let us then notice that if (1) is the case, then by construction i has the degree of at most 5. Similarly, if (2) holds, then the degree of i is bounded by 5. Further, if (3) is the case, then as each literal appears exactly twice in φ, we have that i is adjacent to at most two verifier vertices. Hence, the degree of i is at most 3. Also, if (4) holds, then by construction we have that i has the degree of at most 2. Moreover, if the (5) is the case, then we get that the degree of i is bounded by 3. Finally, if (6) holds, then i is adjacent to at most three literal vertices, as the clause that its gadget corresponds to is limited to three literals. Hence, i is adjacent to at most 6 vertices. Then, the claim follows. □

It is important to note that in order to obtain Observations 8.1 and 8.2, we crucially use the fact that the formulas we encode are 2P2N-3-CNF.

q-majority illusion on Planar Networks. We further show that q-MAJORITY ILLUSION is **NP**-complete also for *planar* networks. Observe that this result rules out using generalisations of planarity as possible structural restrictions to get polynomial-time algorithms. We prove it by reduction from PLANAR 3-SAT. The reduction that we use to show this result follows a similar structure to the one we construct in the proof of Theorem 8.1. So, we first construct a network E_φ, a planar encoding of

φ, which is an input of Planar 3-SAT. We show, in Lemma 8.8, that it admits 1-majority illusion if and only if φ is satisfiable. Then, in Theorem 8.2, we show the **NP**-hardness of q-majority illusion on planar networks similarly to the proof of Theorem 8.1.

Variable, Clause, and Balance Gadgets. For a formula φ in CNF with a planar incidence graph, we denote the set of variables in φ as $X = \{x_1, \ldots, x_m\}$, and the set of clauses in φ as $C = \{C_1, \ldots, C_n\}$. We assume, without loss of generality, that there are at least two clauses in φ. Let us first encode the propositional variables. For each variable $x_i \in X$, we construct a subnetwork, which we call a *variable gadget*, as depicted in Figure 8.4. Further one of the vertices in this structure corresponds to x_i, and one to $\neg x_i$, as shown in Figure 8.4. We call them *literal vertices.* Let us first observe that there are seventeen vertices in this gadget. Further, notice that seven of them have dependents, which implies that in every labelling of a variable gadget, which induces 1-majority illusion, they are labelled r. We also observe that this gadget is planar.

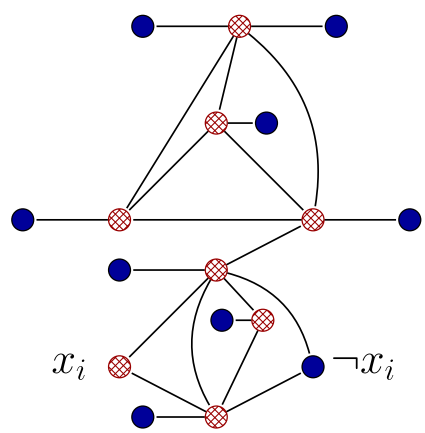

Figure 8.4: Variable gadget for the variable x_i, with a labelling such that all members are under illusion. Two of the vertices in the gadget correspond to literals, x_i and $\neg x_i$. Observe that this structure is planar.

Similarly to the encoding used in the previous part of this section, we show that in a labelling of a variable gadget, which induces 1-majority illusion, exactly one of the literal vertices is labelled r. We will later use this fact to demonstrate that every labelling of the encoding of φ, which induces 1-majority illusion, corresponds to some valuation over X.

Lemma 8.6. *A labelling of a variable gadget (considered as a separate network)* *induces a 1-majority illusion only if exactly one of the literal vertices is labelled r.*

Proof. First observe that in every labelling f of a variable gadget V_i for x_i, which induces 1-majority illusion, seven of the members of V_i need to be labelled r, as they have dependents. Then observe that as there are seventeen vertices V_i, at most eight of them are coloured r in f. Let us now suppose that exactly one of literal vertices is coloured r and, without loss of generality, let this vertex correspond to x_i. Then it is enough to consider a labelling as presented in Figure 8.4, and to observe that then, there exactly eight vertices are labelled r, while all of the members of the gadget are under illusion.

Then let assume towards contradiction that both of the literal vertices are labelled b in f. Then, by previous observations, it holds that seven vertices with dependents are labelled r, and hence at most one node which is a dependent is labelled r. Note that this implies that at least a half of neighbours of one of the vertices adjacent to the vertex corresponding to x_i is labelled b, and thus it is not under illusion. Contradiction. □

We note that there are exactly two labellings of a variable gadget for x_i, which induce 1-majority illusion. We say that such a labelling is of type A, if the literal vertex corresponding to x_i is coloured r, and of type B, if the literal vertex corresponding to $\neg x_i$ is coloured r. Intuitively, if this gadget is labelled in type A, then x_i is set to true, while if it is labelled in type B, it is set to false.

Furthermore, let us define, what we call a *clause gadget*, which corresponds to a clause $C_i \in C$. We begin with introducing what we call a *planar filling structure* (PFS), as depicted in Figure 8.5. It consists of nine vertices, four of which have dependents. Observe that in every labelling of this structure, which induces 1-majority illusion, at least four vertices are labelled r. Notice that this gadget is planar.

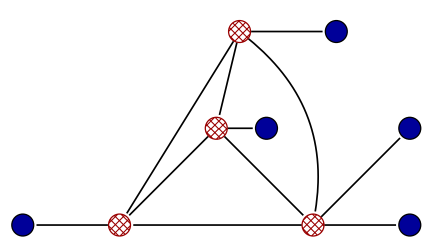

Figure 8.5: Planar filling structure (PFS) with a labelling, such that all members are under illusion.

Then, the clause gadget, as shown in Figure 8.6, for a clause C_i, consists of three copies of PFS, as well as four additional vertices. The central vertex, which we call a *verifier vertex*, is adjacent to literal vertices corresponding to the members of C_i. Further, it is adjacent to the top-central vertex of each PFS in the gadget. Finally, it has one dependent, and the remaining two vertices form an isolated pair. Observe that this structure is planar.

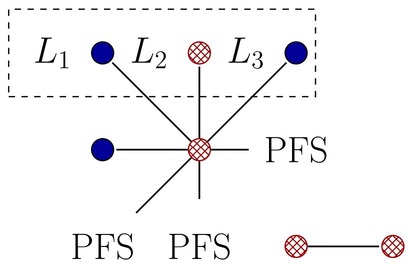

Figure 8.6: Clause gadget, with a labelling, such that all members are under illusion.

We further show that as in the case of the encoding used earlier in this section, a clause gadget admits 1-majority illusion only when the verifier vertex is adjacent to some literal vertex, which is labelled r. We will later associate the fact that it is the case for a clause gadget corresponding to a clause C_i with C_i being satisfied in a valuation over X.

Lemma 8.7. *There exists a labelling of a clause gadget for C_i (not as a separate network), which induces 1-majority illusion, if and only if at least one of literal vertices corresponding to a member of C_i is labelled r.*

Proof. Let us first observe that there are thirty-one vertices in the clause gadget corresponding to the clause C_i. Thus, in a labelling f inducing 1-majority illusion, at most fifteen of them are labelled r. Further, as observed before, we have that twelve members of PFS subnetworks in the gadget are labeled r in f, as they have dependents. Moreover, in such a labelling, the isolated pair of vertices also needs to be labelled r, as otherwise one of them would not be under illusion. Furthermore, the verifier vertex needs to be labelled r as well, since it has a dependent. Then, it follows that all other vertices in the gadget, including one of verifier vertex's neighbours, are labelled b in f, as otherwise r would not be the strict minority colour.

Let us then assume that at least one of the literal vertices adjacent to the verifier vertex is labelled r. Without loss of generality, let it be L_i^2. Then, it is enough to construct a labelling, as shown in Figure 8.6 and Figure 8.5, and notice that then all vertices in the gadget are under illusion. Now, suppose towards contradiction that f induces 1-majority illusion, but all literal vertices corresponding to literals in C_i are labelled b. Then notice that, by previous observations, one of the neighbours of the verifier vertex within the gadget is labelled b. But then, four out of seven vertices adjacent to the verifier vertex are labelled b, and thus it is not under majority illusion. Contradiction. □

Finally, for $k \geqslant 2$, we define what we call a *balance gadget*, similar to the construction for bipartite graphs. If k is even, then the balance gadget is the collection of $\frac{k}{2}$ pairs vertices, which are disconnected from all other vertices in the encoding. Otherwise, we construct a triple of vertices, which forms a clique disconnected from the rest of the network, as well as the balance gadget for $k-3$, if $k \geqslant 5$. Observe that the balance gadget is planar, and that for every labelling of this gadget, which induces 1-majority illusion (not as a separate network), all of its members are labelled r. Observe that this structure is planar.

Encoding of a 3-CNF Formula. We can now construct a social network $E_\varphi = (N, E)$, which *encodes* a 3-CNF φ, in which the incidence graph between variables and clauses is planar. First, for every variable $x_j \in X$, let us construct a variable gadget, as depicted in Figure 8.4. Further, for every clause $C_i \in C$, which we denote as $\{L_i^1, L_i^2, L_i^3\}$, let us create a clause gadget, as shown in Figure 8.6, with literal vertices corresponding to L_i^1, L_i^2, and L_i^3 being adjacent to the verifier vertex in the clause gadget that corresponds to C_i. As a final step, let us construct a balance gadget with $k = m + n - 1$, which by construction is always at least two, since we assumed that there are at least two clauses in φ. Observe how since the incidence graph of φ is planar, we can obtain that so is E_φ.

Notice further that since there are $m + n - 1$ vertices in the balance gadget, there are $18m + 32n - 1$ vertices in E_φ. Let us further notice that following previous observations, for every labelling of E_φ that induces 1-majority illusion, all vertices in the balance gadget are labelled r. Also, then, for every variable gadget, which consists of seventeen vertices, at least eight of its members are labelled r. Similarly, in such a labelling, for every clause gadget, we have that exactly fifteen out of thirty one-members of the gadget are labelled r. This implies that in such a labelling each variable gadget is either labelled in type A, or in type B, as otherwise b would not be a strict majority colour.

We can now show that the encoding of φ admits 1-majority illusion exactly when φ is satisfiable.

Lemma 8.8. *Let φ be a formula in 3-CNF. Then, φ is satisfiable if and only if E_φ admits 1-majority illusion.*

Proof. Take a formula φ in 3-CNF, with the set of variables $X = \{x_1, \dots, x_m\}$, and the set of clauses $C = \{C_1, \dots, C_n\}$. Then, we show that the ecncoding E_φ of φ admits 1-majority illusion if and only if φ is satisfiable.

Let us first suppose that it is, and show that E_φ admits 1-majority illusion. Take a model M of φ and label E_φ as follows. First, colour variable gadgets, so that every such gadget corresponding to some variable $x_i \in X$ is of type A, if x_i is true in M, and of type B, if x_i is false in M. Then observe that by the construction of E_φ, every verifier vertex in the construction is adjacent to some literal vertex, which is labelled r, as all clauses are satisfied under M. Hence, following previous observations, E_φ admits 1-majority illusion.

Otherwise, suppose that φ is not satisfiable. Then, observe that since every labelling of E_φ that admits a 1-majority illusion requires variable gadgets to be labelled in type A or type B, and φ is not satisfiable, it holds that at least one verifier vertex would need to be adjacent to three literal vertices labelled b. But then, it would not be under the majority illusion, which contradicts the assumptions. Hence, E_φ admits 1-majority illusion if and only if φ is satisfiable. □

We now show some further properties of E_φ, which will be crucial towards showing that q-MAJORITY ILLUSION is **NP**-hard for every $q \in (0, 1]$, even for planar network. Given a 3-CNF formula φ, let $I_\varphi = 18m + 32n - 1$, where m is the number of variables and n the number of clauses in φ. Observe that this is the maximum number of vertices, which can be labelled red in E_φ if blue is the strict majority colour in this network. Also notice that following previous observations it is the minimum number of vertices, which need to be coloured red in a labelling that induces 1-majority illusion in E_φ.

We can now show, that if less than I_φ vertices are coloured r in E_φ, then the number of vertices under illusion is limited.

Lemma 8.9. *For every 3-CNF formula φ, $k \leqslant I_\varphi$, and labelling f of $E_\varphi = (N, E)$, such that $R_f = I_\varphi - k$, the number of vertices under illusion in E_φ under f is at most $\#N - k$.*

Proof. Take a formula φ in 3-CNF and a natural k, such that $k < I_\varphi$, as well as a labelling f of E_φ, such that $R_f = I_\varphi - k$. We will show that the number of vertices in E_φ, which are not under illusion under f, is at most $\#N - k$. Let us denote as A the set of all vertices in the four-cliques in the PFS structures, verifier vertices, and all vertices in variable gadgets, which have dependents. Further, let B' be the set of all literal vertices. Then, we denote as C the set of all vertices in balance gadgets, and in isolated pairs in clause gadgets. Finally, D is the set of all other vertices in E_φ. Notice, that $A \cup B' \cup C \cup D = N$. Moreover, by construction, we have that $\#A + \#B' + \frac{\#C}{2} = I_\varphi$.

We now show several properties of A, B', and C. First, observe that all vertices in A have dependents. This implies that there exists a set $N_A \subseteq D$, such that for every $i \in N_A$, i is not under illusion, while $\#N_A = \#B^A$. Let now $M_{B'} = \frac{\#B'}{2} - \#B^{B'}$, if $\frac{\#B'}{2} - \#B^{B'} > 0$, and 0 otherwise. Observe that, by construction, there is a set $N_{B'} \subseteq D$, such that for every $i \in N_{B'}$, i is not under illusion, while $\#N_{B'} = M_C$. Finally, we note that there is a set $N_C \subseteq C$, such that for every $i \in C$, i is not under illusion, while $\#N_C \geqslant \#B^C$. Let us also observe that N_A, $N_{B'}$, and N_C, are pairwise disjoint.

We are now ready to show, that at least k of them are not under illusion under f. Notice, that $\#A + \frac{\#B'}{2} + \#C = I_\varphi$, and that at most I_φ vertices are labelled r in f, as otherwise b would not be the strict majority colour. But then, at least k vertices are labelled b in $A \cup B' \cup C \cup D$. This implies, however, that $N_A + N_{B'} + N_C + D \geqslant k$, and hence at least k vertices are not under illusion under f. □

Observe also that, by the reasoning similar to the proof of Lemma 8.9, we also get that, for a labelling f of E_φ, which maximizes the number of vertices under illusion (which we call M), $k \leqslant I_\varphi$ and any labelling f' of $E_\varphi = (N, E)$, such that $R_{f'} = I_\varphi - k$, the number of vertices under illusion in E_φ under f' is at most $M - k$. We are now ready to show **NP**-completness of q-MAJORITY ILLUSION for planar graphs. The proof is similar to the proof of Theorem 8.1.

Theorem 8.2. *q-MAJORITY ILLUSION is **NP**-complete for every rational q in $(\frac{1}{2}, 1]$, even for planar networks.*

Proof. Take any rational q in $(\frac{1}{2}, 1]$. First, notice that as observed before, q-MAJORITY ILLUSION is in **NP**. We will now show that it is **NP**-hard, by reduction from *Planar 3-SAT*.

Consider a 3-CNF formula φ with the set $X = \{x_1, \dots, x_m\}$ of variables, and the set $C = \{C_1, \dots, C_n\}$ of clauses, with a planar incidence graph. Let us construct what we call a *q-encoding* $E^q_\varphi$of φ. First, let E_φ be a subnetwork of the q-encoding of φ. Moreover, we construct $h^*_{\#V(E_\varphi),q}$ pairs of vertices, such that vertices in each such pair are connected to each other, but not to any other vertex in the network. We call this set of pairs H. Observe further that the q-encoding of φ

can be constructed in polynomial time. Also, by Lemma 8.8 and Lemma 8.5, the q-encoding of φ admits q-majority illusion if at least $\#V(E_\varphi) + h^*_{\#V(E_\varphi),q}$ vertices are under illusion in f.

Let us show now that the q-encoding of φ admits q-majority illusion if and only if φ is satisfiable. First, suppose that φ is satisfiable. Then observe that as φ is satisfiable, by Lemma 8.8, it holds that E_φ admits 1-majority illusion as a separate network. Hence, there is a labelling of the q-encoding of φ, such that exactly I_φ vertices in E_φ, as well as one of vertices in each additional pairs, are labelled red, and $\#V(E_\varphi) + h^*_{\#V(E_\varphi),q}$ vertices are under illusion. Hence, the q-encoding of φ admits q-majority illusion.

Suppose now that φ is not satisfiable. Then, suppose that there is a labelling f of the q-encoding of φ, which induces q-majority illusion. Let us first observe that if less than $h^*_{\#V(E_\varphi),q}$ are labelled red in H, then f does not induce q-majority illusion. Indeed, if it was the case, then less than $h^*_{\#V(E_\varphi),q}$ vertices in H would be under illusion, and hence the number of vertices under illusion in the q-encoding of φ would be strictly smaller than $\#V(E_\varphi) + h^*_{\#V(E_\varphi),q}$. But then, as f induces q-majority illusion, we have that, following Lemma 8.5, at least $h^*_{\#V(E_\varphi),q}$ vertices are labelled red in H. So, the number of vertices labelled red in E_φ is smaller or equal to I_φ. If it is equal to I_φ, then the number of vertices under illusion in H is $h^*_{\#V(E_\varphi),q}$, but as φ is not satisfiable, not all members of E_φ are under illusion, and hence f does not induce q-majority illusion. Now, suppose that less than I_φ vertices are labelled red in $\#\varphi$. Let $k = I_\varphi - R^{V(E_\varphi)}$. Further, let us denote as M the maximum number of vertices under illusion in E_φ, if I_φ vertices are labelled red in this subnetwork. Now, by Lemma 8.9, we have that the number of vertices under illusion is at most $M - k$. But then, the number of vertices labelled red in H is at most $h^*_{\#V(E_\varphi),q}+k$, and hence the number of vertices under illusion in the q-encoding of φ is at most $M - k + h^*_{\#V(E_\varphi),q}$, which is smaller than $\#V(E_\varphi) + h^*_{\#V(E_\varphi),q}$, since $M < \#V(E_\varphi)$. It follows, by Lemma 8.9, that less than $q \cdot \#V(E^q_\varphi)$ vertices are under illusion, which contradicts the assumptions.

□

Then, from the fact that networks with clique size greater than 4 are not planar, we get the following observation.

Observation 8.3. q-MAJORITY ILLUSION *is* **NP**-*complete, even for networks with maximum clique-size bounded by some constant greater than 4, for every rational* $q \in (\frac{1}{2}, 1]$.

8.2.2 Parametrised Complexity Results

Our **NP**-completeness results for q-MAJORITY ILLUSION motivate the study of this problem from the perspective of parametrised complexity, with the aim of identifying various restrictions on its input, which allow for tractability. Note that our result that q-MAJORITY ILLUSION is **NP**-hard on networks of constant max-degree implies that, unless **P**=**NP**, q-MAJORITY ILLUSION does not have an algorithm deciding it, with a running time $\#N^{f(\Delta)}$, for any computable function f, where Δ is the max-degree. In other words, q-MAJORITY ILLUSION is para-**NP**-hard, when parametrised by Δ. Hence, we extend this parameterisation, using other structural properties of the graph. Our first fixed-parameter tractability result, i.e., Theorem 8.3, states that if we parameterise q-MAJORITY ILLUSION by the max-degree *and* tree width of the input network, then we can obtain a FPT algorithm. The idea behind our proof is that we can use dynamic programming over a nice tree decomposition of a network to check if it admits q-MAJORITY ILLUSION, assuming that the maximum degree of vertices in this network is bounded.

We next prove our first parametrised tractability result.

Theorem 8.3. *q-MAJORITY ILLUSION can be solved in time $\Delta^{\mathcal{O}(k)}\#N^{\mathcal{O}(1)}$ on networks of tree width k and max-degree Δ.*

Proof. Let the input graph be G, with tree width k. We first run the $2^{\mathcal{O}(k)}\#N^{\mathcal{O}(1)}$-time 2-approximation algorithm of Korhonen [2021], in order to to compute a tree decomposition of width at most $2k+1$, and then use the well-known polynomial-time algorithm to convert any given tree decomposition to a nice tree decomposition of the same width (see Cygan et al. [2015]). We now design a dynamic programming algorithm over this nice tree decomposition (T, β), of width at most $2k + 1$.

We define a boolean function H (i.e., to the set $\{0, 1\}$), whose domain is the set of all tuples, where each tuple comprises a vertex $t \in V(T)$, a labelling $\mathsf{col} : \beta(t) \to \{r, b\}$ of vertices in the bag $\beta(t)$, a function $\mathsf{esurp} : V_t \to \{-\Delta, \dots, \Delta\}$, where $\mathsf{esurp}(v) = 0$ for all vertices $i \notin \beta(t)$, a function $\mathsf{isurp} : \beta(t) \to \{-\Delta, \dots, \Delta\}$, some $\alpha \in [0, \#N]$, and some $\ell_r \in [0, \#N]$. If $\beta(t) = \emptyset$, then we have that $\mathsf{col} = \mathsf{esurp} = \mathsf{isurp} = \emptyset$. We further define $H(t, \mathsf{col}, \mathsf{esurp}, \mathsf{isurp}, \alpha, \ell_r) = 1$ if and only if there exists a labelling $\rho : V_t \to \{r, b\}$, such that the following hold:

1. For every $i \in \beta(t)$, we have that $\rho(i) = \mathsf{col}(i)$.

2. The size of the set $R_\rho^{V_t} = \{i \in V_t : \rho(i) = r\}$ is ℓ_r.

3. α is the size of the set

$$\{i \in V_t : \#N^{V_t}_{\rho,r}(i) > \#N^{t}_{\rho,b}(i) + \mathsf{esurp}(i)\}$$

4. For every $i \in \beta(t)$, we have that $\mathsf{isurp}(i) = \#N^{V_t}_{\rho,b}(i) - \#N^{V_t}_{\rho,r}(i)$ captures the internal blue surplus of every vertex in $\beta(t)$ under ρ.

The intuition behind the description of the function H is the following. Consider a hypothetical labelling f for the social network $SN = (N, E)$ that witnesses q-majority illusion. Then, fix a bag $\beta(t)$, and let δ be the restriction of f to the set V_t. Subsequently, we have that:

1. col is the restriction of δ to the vertices of the bag $\beta(t)$.

2. The function esurp (read *external surplus*) describes the blue surplus for the vertices in V_t, i.e., provided by the vertices outside of the set V_t. Note that then only vertices of the bag $\beta(t)$ get non-zero blue surplus from outside of V_t, since only these vertices (among those in V_t) have any neighbours outside of V_t, by the definition of a tree decomposition. Hence, we may assume a value of 0 "external" blue surplus, for all vertices in V_t, which are not in $\beta(t)$. On the other hand, since the max-degree of the graph is Δ, the "external" blue surplus of any vertex in $\beta(t)$ is at least $-\Delta$, and at most Δ.

3. The value of ℓ_r is the number of vertices of V_t that are assigned r by f, and hence also by δ.

4. The number α is the number of vertices of V_t which are under illusion with respect to f. This includes all vertices in $V_t \setminus \beta(t)$, which have more red neighbours than blue neighbours under δ, and all vertices in $\beta(t)$, for which, if we add the blue surplus given by vertices in V_t (which can be deduced from δ) and the blue surplus from outside V_t (which is given by the function esurp), we get at most -1.

5. Finally, the function isurp (read *internal surplus*) describes the blue surplus for the vertices in $\beta(t)$, which is provided by the vertices within V_t. As for esurp, since the max-degree is Δ, we have that the range of the function lies in $\{-\Delta, \dots, \Delta\}$.

The crux of the correctness of the procedure, which we will define, is that if we could find a labelling, say ρ, for V_t, which is not necessarily in accordance with δ, but has the same "signature" of δ in terms of $\mathsf{col}, \ell_r, \alpha, \mathsf{isurp}$, then, given the same esurp, then we can "cut" δ from f and replace it with ρ. This allows us to obtain another labelling of SN, which has exactly the same number of vertices under illusion as γ. This gives us the so-called *optimal substructure property*, that is crucial for our dynamic programming algorithm.

Notice that there are only $2^{2k+2} \cdot (2\Delta + 1)^{2(2k+2)} \# N^{\mathcal{O}(1)} = \Delta^{\mathcal{O}(k)} \# N^{\mathcal{O}(1)}$ possible tuples. This is because each bag contains at most $2k+2$ vertices, implying at most 2^{2k+2} possibilities for col at any bag and since, for every bag, we have that esurp can only have non-zero values for vertices in the bag (and at most $2\Delta + 1$ possible values at that), we infer that there are at most $(2\Delta+1)^{2k+2}$ possibilities for esurp at any bag. The same bound extends to isurp as well. The remaining elements of the tuple, i.e., α and ℓ_r, are both bounded by $\# N$, and hence there are at most $\# N^2$ possibilities for them at any bag.

Now, suppose that we have computed $H(t, \mathsf{col}, \mathsf{esurp}, \mathsf{isurp}, \alpha, \ell_r)$ for all possible valid values of the arguments. Notice that if this is achieved, then we can answer whether G admits q-majority illusion by examining the table entries corresponding to the root bag $\beta(t^\star)$. Observe that, by the definition of a nice tree decomposition, this bag is empty. Then, we have that SN admits q-majority illusion if and only if there exist values $l_r \in [0, \# N]$ and $\alpha \in [0, \# N]$, such that $\alpha \geqslant \lceil q \cdot \# N \rceil$, $l_r < \frac{\# N}{2}$ and $H(\emptyset, \emptyset, \emptyset, \alpha, \ell_r) = 1$.

We next describe, how to compute the table entries at each bag, by going over the following, exhaustive, cases and, assuming that all the table entries at all descendant bags have been computed correctly.

Leaf Node. Let t be a leaf node. This is our base case. By the definition of a nice tree decomposition, we have that $\beta(t) = \emptyset$. Then, we set $H[t, \emptyset, \emptyset, \emptyset, 0, 0] = 1$. For all other values of α and ℓ_r, we set $H[t, \emptyset, \emptyset, \emptyset, \alpha, \ell_r] = 0$.

Introduce Node. Let t be an introduce node and t' be its child in T, such that $\beta(t) \setminus \beta(t') = \{u\}$. Then, consider the tuple $(t, \mathsf{col}, \mathsf{esurp}, \mathsf{isurp}, \alpha, \ell_r)$, for which we want to fill the table entry. We next define the tuple $(t', \mathsf{col}', \mathsf{esurp}', \mathsf{isurp}', \alpha', \ell_r')$. Let col' denote the restriction of col to $\beta(t')$. If $\mathsf{col}(u) = r$, then we set $\ell_r' := \ell_r - 1$, and otherwise we set $\ell_r' = \ell_r$. Let $\mathsf{esurp}' : V_{t'} \to \{-\Delta, \ldots, \Delta\}$, and $\mathsf{isurp} : \beta(t) \to \{-\Delta, \ldots, \Delta\}$, be defined as follows. For every vertex v in $V_{t'} \setminus \beta(t')$, set $\mathsf{esurp}'(v) = 0$. Also, for every vertex v in $\beta(t)$, which is a neighbour of u, if $\mathsf{col}(u) = r$, then we set $\mathsf{esurp}'(v) = \mathsf{esurp}(v) - 1$, and set $\mathsf{isurp}'(v) = \mathsf{isurp}(v) + 1$. Further, for every vertex v in $\beta(t)$, which is a neigbhbour of u, if $\mathsf{col}(u) = b$, then we set $\mathsf{esurp}'(v) = \mathsf{esurp}(v) + 1$, and we set $\mathsf{isurp}'(v) := \mathsf{isurp}(v) - 1$. Finally, we define α' as follows. If $\mathsf{esurp}(u)$ plus the number of neighbours of u in $\beta(t')$, which are labelled blue under col, minus the number of neighbours of u in $\beta(t')$, which are labelled red under col is at most -1, then we set $\alpha' = \alpha - 1$. Otherwise, we set $\alpha' = \alpha$. Now, we set

$$H[t, \mathsf{col}, \mathsf{esurp}, \mathsf{isurp}, \alpha, \ell_r] := H[t', \mathsf{col}', \mathsf{esurp}', \mathsf{isurp}', \alpha', \ell_r'].$$

Forget Node. Let t be a forget node and t' be its child in T, such that $\beta(t') \setminus \beta(t) = \{u\}$. Consider the tuple $(t, \mathsf{col}, \mathsf{esurp}, \mathsf{isurp}, \alpha, \ell_r)$, for which we want to fill the table entry. We set $H[t, \mathsf{col}, \mathsf{esurp}, \mathsf{isurp}, \alpha, \ell_r] = 1$ if and only if there exists $\mathsf{col}', \mathsf{esurp}', \mathsf{isurp}'$, such that (1) $H[t', \mathsf{col}', \mathsf{esurp}', \mathsf{isurp}', \alpha, \ell_r] = 1$, (2) col is the restriction of col' to $\beta(t)$, and (3) esurp (isurp) is the restriction of esurp' (respectively, isurp') to $\beta(t)$.

Join Node. Let t be a join node and t_1, t_2 be its children in T. Then, by the definition of a nice tree decomposition, we have that $\beta(t) = \beta(t_1) = \beta(t_2)$. Consider the tuple $(t, \mathsf{col}, \mathsf{esurp}, \mathsf{isurp}, \alpha, \ell_r)$, for which we want to fill the table entry. We set $H[t, \mathsf{col}, \mathsf{esurp}, \mathsf{isurp}, \alpha, \ell_r] = 1$ if and only if there exists a pair of tuples $(t_1, \mathsf{col}, \mathsf{esurp}_1, \mathsf{isurp}_1, \alpha_1, \ell_{r,1})$ and $(t_2, \mathsf{col}, \mathsf{esurp}_2, \mathsf{isurp}_2, \alpha_2, \ell_{r,2})$, such that

1. The table entries $H[t_1, \mathsf{col}, \mathsf{esurp}_1, \mathsf{isurp}_1, \alpha_1, \ell_{r,1}]$ and $H[t_2, \mathsf{col}, \mathsf{esurp}_2, \mathsf{isurp}_2, \alpha_2, \ell_{r,2}]$ are both 1.

2. $\alpha = \alpha_1 + \alpha_2 - x$, where x is the number of vertices of $\beta(t)$ forced to be under illusion by the combination of esurp and isurp. That is, x is the size of the set $\{v \in \beta(t) : \mathsf{esurp}(v) + \mathsf{isurp}(v) \leqslant -1\}$. We are subtracting x from $\alpha_1 + \alpha_2$, because these vertices are counted in both α_1 and α_2.

3. $\ell_r = \ell_{r,1} + \ell_{r,2} - y$, where y is the number of vertices of $\beta(t)$ labelled red by col. That is, y is the size of the set $\{v \in \beta(t) : \mathsf{col}(v) = r\}$. We are subtracting y from $\ell_{r,1} + \ell_{r,2}$, because the vertices of $\beta(t)$ labelled red by col is counted once in $\ell_{r,1}$, and once in $\ell_{r,2}$.

4. For every $v \in \beta(t_1)$, we have that $\mathsf{esurp}_1(v) = \mathsf{esurp}(v) + \mathsf{isurp}_2(v) - \#N^{\beta(t_1)}_{\mathsf{col},b}(v) + \#N^{\beta(t_1)}_{\mathsf{col},r}(v)$. Here, we are saying that the blue surplus of a vertex v, external to the set V_{t_1}, should be obtained by taking the blue surplus of v, external to both V_{t_1} and V_{t_2} (which is given by $\mathsf{esurp}(v)$), and then adding to it the blue surplus of v *internal* to V_{t_2} (while accounting for edges between vertices in $\beta(t_1)$). Precisely, we subtract $\#N^{\beta(t_1)}_{\mathsf{col},b}(v) - \#N^{\beta(t_1)}_{\mathsf{col},r}(v)$, because these quantity deals with blue surplus, which is given by the edges between vertices of $\beta(t_1)$, and these should not be counted in the external surplus of v with respect to the bag $\beta(t_1)$.

5. For every $v \in \beta(t_2)$, we have that $\mathsf{esurp}_2(v) = \mathsf{esurp}(v) + \mathsf{isurp}_1(v) - \#N^{\beta(t_2)}_{\mathsf{col},b}(v) + \#N^{\beta(t_2)}_{\mathsf{col},r}(v)$. The reasoning behind this constraint is symmetrical to the previous one.

6. For every $v \in \beta(t)$, we have that $\mathsf{isurp}(v) = \mathsf{isurp}_1(v) + \mathsf{isurp}_2(v) - \#N^{\beta(t)}_{\mathsf{col},b}(v) + \#N^{\beta(t)}_{\mathsf{col},r}(v)$. Here, we are saying that the surplus of v internal to V_t should be obtained by taking the surplus of v internal to V_{t_1} and to V_{t_2}, and adding them, while accounting for the fact, that we are double-counting the contribution of edges within $\beta(t)$. This motivates the subtraction of $\#N^{\beta(t)}_{\mathsf{col},b}(v) - \#N^{\beta(t)}_{\mathsf{col},r}(v)$.

Notice that filling all the table entries corresponding to any specific bag is dominated the time taken for the join nodes, which in turn is dominated by $\#N^{\mathcal{O}(1)}$ times the number of possible tuples to consider, from each of the two children bags. Hence, the time taken to fill the entries for any one bag is bounded by $\Delta^{\mathcal{O}(k)}\#N^{\mathcal{O}(1)}$, and as we have argued earlier, there are at most $\Delta^{\mathcal{O}(k)}\#N^{\mathcal{O}(1)}$ possible tuples corresponding to each bag. The stated running time then follows. This completes the proof of the theorem. $\square$

We next discuss some immediate implications of the above result. First of all, notice that Δ, i.e., the max-degree, is at most $\#N$. Hence, our FPT algorithm, parametrised by Δ and the tree width, is in fact an XP algorithm, parametrised by the tree width alone.

Corollary 8.1. q-MAJORITY ILLUSION *can be solved in time* $\#N^{\mathcal{O}(k)}$ *on networks of tree width* k.

Secondly, consider the following relation between tree width and another well-studied graph width parameter, i.e., cliquewidth (Gurski and Wanke [2000]), which we denote by $cw(G)$), on bounded-degree graphs. We make use of the following result.

Proposition 8.1. *Gurski and Wanke [2000] Let* G *be a graph that does not contain the complete bipartite graph* $K_{d,d}$ *as a subgraph. If* $cw(G) \leqslant k$, *then it holds that* $tw(G) \leqslant 3k(d-1)-1$.

Since graphs with max-degree Δ exclude $K_{\Delta+1,\Delta+1}$ as a subgraph, Proposition 8.1, along with Theorem 8.3, implies that q-MAJORITY ILLUSION in **FPT**, when parametrised by the maximum degree and cliquewidth of the input graph.

Corollary 8.2. MAJORITY ILLUSION *can be solved in time* $\Delta^{\mathcal{O}(\Delta \cdot k)}\#N^{\mathcal{O}(1)}$ *on networks of max-degree* Δ *and cliquewidth* k.

Neighbourhood Diversity. Here, we provide an FPT algorithm for q-MAJORITY ILLUSION parametrised by neighbourhood diversity. The following properties of labellings of social networks form the crux of our algorithm.

Lemma 8.10. *Let (N, E) be a social network, and let $\mathcal{C} = \{T_1, \ldots, T_k\}$ denote a partition of N into k modules. Further, let $f : N \to \{r, b\}$ be a labelling, where b is the majority colour. Then, the following hold:*

1. *If one vertex of an independent module is under illusion under f, then every vertex of this module is under illusion.*
2. *If a blue vertex (i.e, a vertex labelled b) of a clique module is under illusion under f, then all blue vertices in this module are also under illusion.*
3. *If a red vertex of a clique module is under illusion under f, then every vertex in this module is also under illusion.*

Proof. Let us first of all recall that, by the definition of neighbourhood diversity, we have that every pair of vertices in each module have exactly the same neighbourhood outside the module. Hence, the first statement immediately follows since, in an independent module, there are no edges within the module. Now, consider the second statement, and fix a clique module C. Let further $u, v \in C$ be a pair of blue vertices. Then, the number of red (blue) neighbours of u within C is exactly the same, as the number of red (respectively, blue) neighbours of v within C. This implies that if u is under illusion, then so is v. Now, consider the third statement. By an argument identical to that for the second statement, we can conclude that if a red vertex is under illusion, then all red vertices are under illusion.

It is now sufficient to argue that if a red vertex is under illusion, then at least one blue vertex (assuming it exists) is under illusion. Let u be a red vertex in the module, which is under illusion, and let v be a blue vertex in the module. Recall that u and v have the same neighbourhood outside C. Moreover, the number of blue neighbours of u within C is strictly greater than the number of blue neighbours of v in C. Consequently, u has strictly fewer red neighbours within C than v. This implies that if u is under illusion, then so is v. This completes the proof of the lemma. □

We will use as a subroutine the well-known FPT algorithm for the ILP-FEASIBILITY problem (recall Proposition 3.1 in Chapter 3). Intuitively, we will make use of Lemma 8.10 to construct a number of ILP-FEASIBILITY instances, sufficient to solve q-MAJORITY ILLUSION. Towards this end, following Claim 8.1, we show that if the proposed constraints are satisfied, then we can conclude that the network admits q-majority illusion. This will allow us to subsequently solve q-MAJORITY ILLUSION efficiently, for small value of neighbourhood diversity.

Theorem 8.4. *q-MAJORITY ILLUSION can be solved in time $2^{\mathcal{O}(k \log k)} \# N^{\mathcal{O}(1)}$ on networks of neighbourhood diversity k.*

Proof. Let $SN = (N, E)$ be a given input social network, and let $\mathcal{T} = \{T_1, \ldots, T_k\}$ denote the partition of N into k modules, each of which is a clique or an independent set. Observe that the set $\mathcal{C}$ can be computed in polynomial time (see Lampis [2012]). Then, for every $i \in [0, k]$, let $adj(i)$ denote the set $\{j \in [0, k] : j \neq i$ and $\exists u \in T_i, v \in T_j : (u, v) \in E\}$. That is, $adj(i)$ comprises the indices of all those modules T_j, in which least one vertex (and hence all vertices) is adjacent to a vertex of T_i (and hence to all vertices of T_i). Let further $\chi = \lceil q \cdot \# N \rceil$ denote the required number of vertices to be under illusion, in order for q-majority illusion to hold. The main intuition behind our algorithm is to construct $2^{\mathcal{O}(k)}$ instances of ILP-FEASIBILITY, each with $\mathcal{O}(k)$ variables, such that if there is a labelling of SN, which induces q-majority illusion, then the solution to one of these ILP-FEASIBILITY instances can be used to obtain a solution to the given instance of q-MAJORITY ILLUSION.

Let now $\mathcal{C}$ denote the set of all clique modules in $\mathcal{T}$, and let $\mathcal{I}$ denote the set of all independent modules in $\mathcal{T}$. We are now ready to start describing the design of the ILP-FEASIBILITY instances. For every function:

- Clique-col: $\mathcal{C} \to \{r, b, \mathsf{both}\}$
- Clique-maj: $\mathcal{C} \to \{b, \mathsf{all}, \mathsf{none}\}$
- Ind-maj: $\mathcal{I} \to \{\mathsf{all}, \mathsf{none}\}$

we construct one ILP-feasibility instance, for which the set of variables and constraints will be discussed later in this proof. We first sketch the intuition behind these functions. Let $f : N \to \{r, b\}$ be a labelling, which places at least χ vertices under illusion (if one exists). Then, the function Clique-col expresses, for every clique module, whether it contains both red and blue vertices according to f. Note that if this is case, then this module is mapped to both. Further, if it contains only red vertices, then this module is mapped to r. Finally, if it contains only blue vertices, then this module is mapped to b. Furthermore, the function Clique-maj expresses, for every clique module, whether no vertices are under illusion (mapped to none), or only blue vertices are under illusion (mapped to b), or all vertices are under illusion (mapped to all) under f. Recall that from the second and third statements of Lemma 8.10, we have that these are the only three possibilities. The function Ind-maj expresses, for every independent module, whether all vertices in the module are under illusion (mapped to all) in the optimal labelling, or none of them are under illusion (mapped to none). Recall, that from the first statement of Lemma 8.10, we have that these are the only two possibilities. If f exists, then a "correct" triple of these functions exist. Notice that there are at most 3^k possibilities for Clique-col and Clique-maj, and at most 2^k possibilities for Ind-maj. Hence, we may iterate over all possible at most 18^k triples of functions, and we know that at least one of these triples is the "correct" one if ρ exists.

Now, let us fix the functions Clique-col, Clique-maj, Ind-maj and describe the ILP-FEASIBILITY instance corresponding to it. In order to better understand the constraints we will design, we consider the three selected functions to be the "correct" ones that correspond to f. We will also assume that these functions are consistent with each other. That is, if Clique-col(T_i) is r (respectively, b), then it cannot be the case that Clique-maj(T_i) is b (respectively, r). In other words, if we guess that every vertex of T_i is labelled red, then we will not guess that all of the blue vertices of T_i will be under illusion. Moreover, we have a convention that in Clique-maj, the value all takes "priority" over r or b. That is, if Clique-col(T_i) is b, then Clique-maj(T_i) is either none or all, and never b. This is because setting it to all achieves the same effect as setting it to b, since all vertices in T_i are blue. Any triple of functions where these conditions are not satisfied are not considered further.

We now proceed to describe the ILP-FEASIBILITY instance. For every $i \in [0, k]$, let s_i denote the size of $V(T_i)$. We know the value of each s_i, since we have we know $\mathcal{T}$. The set of variables in this instance is $\bigcup_{i \in [0,k]} \{r_i, b_i, p_i\}$. The intuitive meaning of these variables is the following. Recall that $f : V(SN) \to \{r, b\}$ is a hypothetical optimal labelling that places at least χ vertices under illusion. Then, for every module T_i, we have that r_i represents the number of vertices of T_i labelled red in f. Similarly, b_i is the number of vertices in T_i labelled blue in f, while p_i will be used to represent the number of vertices of T_i, which are under illusion following f. Notice that we have $3k$ variables in total. We are now ready to present our constraints.

1. For every $i \in [0, k]$, we have the constraint $r_i + b_i = s_i$[1]. This constraint says that the numbers of red and blue vertices of each module should add up to the total number of vertices in that module.

[1]Notice that equality constraints can always be expressed as two inequalities, as required in the definition of ILP-FEASIBILITY. For instance, in this case, we have $r_i + b_i \leqslant s_i$ and $-r_i - b_i \leqslant -s_i$.

2. $-\sum_{i\in[0,k]} b_i + \sum_{i\in[0,k]} r_i \leqslant -1$. This constraints says that the total number of vertices labelled blue must exceed the total number of vertices labelled red. So, as a consequence, we get that blue would be the majority colour.

3. $-\sum_{i\in[0,k]} p_i \leqslant -\chi$. This says that as long as we ensure that the number of vertices under illusion from each T_i is at least p_i, then the total number of vertices under illusion is at least χ.

4. For every $i \in [0,k]$, such that T_i is an independent module, if $\mathsf{Ind\text{-}maj}(T_i) = \mathsf{all}$, then we add the constraint $\sum_{j\in adj(i)} b_j - \sum_{j\in adj(i)} r_j \leqslant -1$. That is, for every vertex in T_i, the number of red neighbours must exceed the number of blue neighbours, i.e., they are all under illusion.

5. For every $i \in [0,k]$, such that T_i is an independent module, if $\mathsf{Ind\text{-}maj}(T_i) = \mathsf{all}$, then we add the constraint $p_i = s_i$, and otherwise (i.e., when $\mathsf{Ind\text{-}maj}(T_i) = \mathsf{none}$), we add the constraint $p_i = 0$.

6. For every $i \in [0,k]$, such that T_i is a clique module, we add a constraint as follows:

   - If $\mathsf{Clique\text{-}maj}(T_i) = \mathsf{none}$, then $p_i = 0$.
   - If $\mathsf{Clique\text{-}maj}(T_i) = b$, then $p_i = b_i$.
   - If $\mathsf{Clique\text{-}maj}(T_i) = \mathsf{all}$, then $p_i = s_i$.

   These constraints ensure that the number of vertices of T_i, which are supposed to be under illusion, match with the information provided by the function Clique-maj, i.e., whether the set of vertices under illusion is empty, or is equal to the set of all blue vertices, or to all of the vertices in the module.

7. For every $i \in [0,k]$, such that T_i is a clique module, we do the following:

   - If $\mathsf{Clique\text{-}col}(T_i) = r$, then we add the constraint $s_i = r_i$. Further, if it also holds that $\mathsf{Clique\text{-}maj}(T_i) = \mathsf{all}$, then we add the constraint

   $$-(r_i - 1) + \sum_{j\in adj(i)} (b_j - r_j) \leqslant -1$$

   - If $\mathsf{Clique\text{-}col}(T_i) = b$, then add the constraint $s_i = b_i$. Further, if it also holds, that $\mathsf{Clique\text{-}maj}(T_i) = \mathsf{all}$, then we add the constraint

   $$b_i - 1 + \sum_{j\in adj(i)} (b_j - r_j) \leqslant -1.$$

These constraints say that if every vertex in T_i is labelled red (blue), then the number of red vertices (respectively, blue vertices) is the total number of vertices in T_i. Moreover, if every vertex is required to be under illusion according to the function $\mathsf{Clique\text{-}maj}$, then the blue surplus of any vertex in T_i is at most -1.

8. For every $i \in [0,k]$, such that T_i is a clique module with $\mathsf{Clique\text{-}col}(T_i) = \mathsf{both}$, we do the following

   - If $\mathsf{Clique\text{-}maj}(T_i) = \mathsf{all}$, then we add the constraint

   $$b_i - r_i + 1 + \sum_{j \in adj(i)} (b_j - r_j) \leqslant -1$$

     This constraint says that if we take a red vertex in T_i, and compute its blue surplus, then it is at most -1. That is, it is under illusion. This in turn implies that every vertex is under illusion, as required by $\mathsf{Clique\text{-}maj}(T_i)$.

   - If $\mathsf{Clique\text{-}maj} T_i = b$, then we add the constraint

   $$b_i - r_i - 1 + \sum_{j \in adj(i)} (b_j - r_j) \leqslant -1$$

     This constraint says that if we take a blue vertex in T_i, and compute its blue surplus, then it is at most -1.

9. For every variable $x \in \bigcup_{i \in [0,k]} \{r_i, b_i, p_i\}$, we have a constraint $-x \leqslant 0$, which imposes a non-negativity constraint on every variable. This will allow us to treat r_i, b_i, p_i as sizes of vertex sets.

10. Finally, for every $i \in [0,k]$, we add the constraint $p_i \leqslant s_i$, in order to indicate that the number of vertices of T_i, which are under illusion, can never be more than the total number of vertices in T_i.

This completes the description of the ILP-Feasibility instance. We refer to the previously defined constraints as C1-C10. Observe that the ILP-Feasibility instance can be computed in polynomial time, given the three functions which we consider in such an instance.

One can further observe that for the optimal labelling ρ, and the corresponding three functions, these constraints are satisfied. This implies that if ρ exists, then at least for one of the triples, the corresponding ILP-Feasibility instance can be solved – for each $i \in [0,k]$, set r_i (b_i) to be the number of vertices of T_i, which are labelled red (blue) by f, and set p_i to be the number of vertices of T_i under illusion.

We next prove that if we solve the ILP-Feasibility instance corresponding to some triple, then we can recover a labelling inducing the required q-majority illusion (which may not be the same as f). Let $\bigcup_{i\in[0,k]}\{r_i^*, b_i^*, p_i^*\}$ be a solution for the ILP-Feasibility instance. Observe that due to C9, we have that all variables get non-negative values. We now define a labelling f^* as follows. For every $i \in [0,k]$, we select an arbitrary set of r_i^* vertices from T_i, and label them red. Furthermore, we label the remaining vertices of each T_i (of which must there must be exactly b_i^*, due to C1), with blue. Since C2 is satisfied, it follows that blue is the majority label. We next prove the following claim, which, along with C3, would then imply that at least χ vertices in total are under illusion, as required.

Claim 8.1. *For every $i \in [0,k]$, the number of vertices of T_i under illusion, with respect to f^*, is at least p_i^*.*

Proof. Consider an independent module T_i. Suppose that $\mathsf{Ind\text{-}maj}(T_i) = \mathsf{all}$. Since C4 is satisfied, it follows that all s_i vertices of T_i are under illusion. Moreover, C5 implies that $p_i^* = s_i$, hence validating our claim that the number of vertices of T_i under illusion with respect to f^* is at least p_i^*. On the other hand, if $\mathsf{Ind\text{-}maj}(T_i) = \mathsf{none}$, then $p_i^* = 0$, and the claim is trivially true, because the number of vertices of T_i under illusion is always at least 0. The same argument applies if we consider a clique module T_i, such that $\mathsf{Clique\text{-}maj}(T_i) = \mathsf{none}$. That is, $p_i^* = 0$, and the claim is trivially true, because the number of vertices of T_i under illusion is always at least 0. Hence, we assume that we are only left with clique modules T_i, such that $\mathsf{Clique\text{-}maj}(T_i) \neq \mathsf{none}$. Now, we have the following, exhaustive, subcases.

Case 1: $\mathsf{Clique\text{-}col}(T_i) = \boldsymbol{r}$ Since we have assumed that $\mathsf{Clique\text{-}maj}(T_i) \neq \mathsf{none}$, it must be the case that $\mathsf{Clique\text{-}maj}(T_i) = \mathsf{all}$. Then, C7 guarantees that $-(r_i^* - 1) + \sum_{j\in adj(i)}(b_j^* - r_j^*) \leqslant -1$. But notice then that we have labelled exactly r_j^* vertices of each T_j red, and the remaining vertices blue. Hence, it must be the case that the blue surplus of every vertex in T_i (as expressed in C7) is at most -1, and so all of the s_i vertices of T_i are under illusion. This satisfies the claim, since p_i^* is always at most s_i (due to C10).

Case 2: $\mathsf{Clique\text{-}col}(T_i) = \boldsymbol{b}$. Again, it must be the case that $\mathsf{Clique\text{-}maj}(T_i) = \mathsf{all}$. Then, C7 guarantees that, since it holds that $b_i^* - 1 + \sum_{j\in adj(i)}(b_j^* - r_j^*) \leqslant -1$, and we have labelled exactly r_j^* vertices of each T_j red and the remaining blue, it follows that the blue surplus of every vertex in T_i is at most -1. So, every vertex of T_i is under illusion. As before, this satisfies the claim, since $p_i^* \leqslant s_i$ (C10).

Case 3: $\mathsf{Clique\text{-}col}(T_i) = \mathsf{both}$. In this case, $\mathsf{Clique\text{-}maj}(T_i)$ could be all or b. In the former case, C8 guarantees that $b_i^* - r_i^* + 1 + \sum_{j \in adj(i)}(b_j^* - r_j^*) \leqslant -1$, implying that at least one red vertex is under illusion. So, by Lemma 8.10(3), we have that every vertex in T_i is under illusion. In the latter case, C8 guarantees that $b_i^* - r_i^* - 1 + \sum_{j \in adj(i)}(b_j^* - r_j^*) \leqslant -1$, implying that at least one blue vertex is under illusion, and so, by Lemma 8.10(2), we have that every blue vertex in T_i is under illusion. Hence, the number of vertices of T_i under illusion is at least the number of blue vertices, i.e., b_i^*. However, in this case C6 guarantees that $p_i^* = b_i^*$ and hence, the number of vertices under illusion in T_i is again at least p_i^*, as required. This completes the proof of the claim. □

We have argued the correctness of the algorithm. Notice further that the running time is bounded by the time required to compute $\mathcal{T}$, which is polynomial, plus 18^k, multiplied by the time required to construct an ILP-Feasibility instance and to execute Proposition 2 in Chapter 3, which is bounded by $2^{\mathcal{O}(k \log k)} \# N^{\mathcal{O}(1)}$. This gives an overall bound of $2^{\mathcal{O}(k \log k)} \# N^{\mathcal{O}(1)}$ on our algorithm. This completes the proof of the theorem. □

Since graphs with vertex cover number at most k have neighbourhood diversity at most $k + 2^k$ (see Lampis [2012]), Theorem 8.4 has the following corollary.

Corollary 8.3. *q-majority illusion can be solved in time $2^{2^{\mathcal{O}(k)}} \# N^{\mathcal{O}(1)}$, on networks with vertex cover number k.*

Table 8.1 shows an overview of parametrised complexity results obtained in this section.

	Parameters
FPT	$\Delta + tw$, $\Delta + cw$, ND, VC
XP	tw
Para-NP-Hard	Δ, c-closure, max-clique-size

Table 8.1: Summary of the main parametrised complexity results on q-majority illusion . Here, ND denotes neighborhood diversity and VC denotes vertex cover number.

8.3 Eliminating Illusion

We now turn to the problem of reducing the number of vertices under illusion in a given labelled network, by modifying the connections between them. Namely, we consider the problem of checking if it is possible to ensure that q-majority illusion does not hold in a labelled network, by altering only a bounded number of edges.

q-ILLUSION ELIMINATION:
Input: $SN = (N, E, f)$, such that f induces q-majority illusion in (N, E, f), $k \in \mathbb{N}$, such that $k \leqslant \#E$.
Question: Is there a $SN' = (N, E', f)$, such that $\#\{(e \in N^2 : e \in E \textit{ iff } e \notin E'\} \leqslant k$, while f does not induce q-majority illusion in SN' ?

We also consider two refinements of this problem. First, let us introduce ADDITION q-ILLUSION ELIMINATION, in which we restrict the possible actions to *adding edges*.

ADDITION q-ILLUSION ELIMINATION:
Input: $SN = (N, E, f)$, such that f induces q-majority illusion in SN, $k \in \mathbb{N}$, such that $k \leqslant \#E$.
Question: Is there a $SN' = (N, E', f)$, such that $E \subseteq E'$, $\#E' - \#E \leqslant k$, and f does not induce q-majority illusion in SN'?

We also define REMOVAL q-ILLUSION ELIMINATION, for *removing edges.*

REMOVAL q-ILLUSION ELIMINATION:
Input: $SN = (N, E, f)$ such that f induces q-majority illusion in SN, $k \in \mathbb{N}$ such that $k \leqslant \#E$.
Question: Is there a $SN' = (N, E', f)$ such that $E' \subseteq E$, $\#E - \#E' \leqslant k$ and f does not induce q-majority illusion in SN'?

8.3.1 Hardness

In this section we show that q-ILLUSION ELIMINATION is **NP**-complete. In fact, our construction implies that this problem is also $\mathbf{W}[1]$-hard, when parametrised by the number of changed edges in a social network. We obtain that by providing the required reduction from k-CLIQUE. In the following, we give a sketch of our reduction to convey the necessary intuition behind our proof, which is followed by a formal proof.

Consider an instance (G,k) of k-Clique, where $G = (V_G, E_G)$. We now construct a social network (N,E,f) as follows. First, we add the vertex set V_G to N, and the edge set E_G to E. We further assign each vertex i in V_G the label red, that is $f(i) = r$. Next, for each vertex $i \in V_G$, we add a set r_i of red labelled vertices, and a set b_i of blue labelled vertices, while ensuring that number of red neighbours of i is exactly $k-1$ more than the number of blue neighbours of i. The idea behind adding these vertices is to make sure that each vertex in V_G has a red surplus of exactly $k-1$. Then, the vertices in V_G are under illusion. Now, we impose the condition that only the vertices in V_G remain under illusion by adding, for each vertex j in the sets r_i and b_i, two blue labelled vertices j_1, j_2 and adding edges $(j_1, j), (j_2, j), (j_1, j_2)$. Then, j is not under illusion, as it has two blue labelled neighbours, as well as one red labelled neighbour. Moreover, j_1, j_2 are not under illusion, as they have one red labelled and one blue labelled neighbour.

We show, in Lemma 8.13, that it is possible to eliminate illusion from k vertices in this structure by altering at most $\binom{k}{2}$ edges exactly when there exists a k-clique in G. Next, we add some extra red and blue labelled vertices, which are not under illusion, to guarantee that blue is the majority label, and the ration of vertices under illusion minus k, to the total number of vertices, is at most q. We set our budget (of edge modifications in the network) to be $\binom{k}{2}$, i.e., $\frac{k^2-k}{2}$. This completes our reduction.

In order to argue the correctness of our reduction, we show that in order to remove q-majority illusion from the constructed network, we must make sure that at least k more vertices are not under under illusion, and that these must come from V_G, as only vertices in V_G are under illusion. In order to achieve this, at least $k-1$ edges on each such vertex must be removed. Achieving this goal by deleting at most $\binom{k}{2}$ edges is only possible, if there is a clique on k vertices in G. Conversely, if we can delete any $\binom{k}{2}$ edges to make k vertices under illusion in (N,E,f), then there must be a k-clique in G. We show that this is the case in the proof of Lemma 8.14, which implies the $\mathbf{W}[1]$ hardness of the problem we consider.

We start with two additional structures, which we call *m-pump-up* and *m-pump-down* gadgets. These gadgets will help us to ensure the correctness of the construction, which we will provide, for a chosen q. We note that adding them to a network, in which r is the minority colour, does not affect the fact that b is the majority colour.

m-Pump-Up Gadget. Let us construct what we call an *m-pump-up gadget.* For a natural number $m \geqslant 1$, we create $m+4$ blue vertices, which are not connected to each other. In addition, we construct 4 red vertices, which are also not connected to each other. Furthermore, let each red vertex in the gadget be connected to all blue vertices in this structure. Note that this forms $K_{m+4,4}$. Observe that if a m-pump-up gadget is embedded in a network in which blue is the majority colour, then $m+4$ vertices are under illusion in this structure, while 4 are not. Also, for every blue vertex i in the gadget, the margin of victory of i is -4. Figure 8.7 depicts a 2-pump-up gadget.

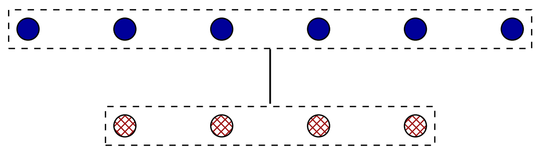

Figure 8.7: m-pump-up gadget for $m = 2$.

m-Pump-Down Gadget. Let us further construct what we call an *m-pump-down gadget*. For an odd, natural number $m \geqslant 1$, the m-pump-down gadget is a m-clique, in which blue has the majority of 1. Also, for an even, natural number $m \geqslant 2$, we construct the pump-down gadget for $m-1$, and a disjoint red vertex. Observe that if an m-pump-down gadget is embedded in a network in which blue is the majority winner, then all m members of the structure are not under illusion. Moreover, if a blue vertex in the gadget would be adjacent to an additional red vertex, then it would be pushed into illusion. Figure 8.8 depicts a 4-pump-down gadget.

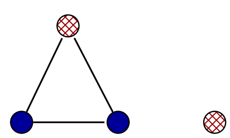

Figure 8.8: m-pump-down gadget for $m = 4$.

The following technical lemmas will help us decide what is the number m for which we are required to add either an m-pump-up or an m-pump-down gadget. The first of them is related to the pump-up gadget.

Lemma 8.11. *For every pair of natural numbers $m, k > 0$ and every rational number q in $(0,1)$, such that $\frac{m}{k} < q$ there exists an h, such that $\frac{m+h}{k+h+4} < q$, but $\frac{m+h+1}{k+h+4} \geqslant q$.*

Proof. Take any such k, m and $q = \frac{a}{b}$, such that $\frac{m}{k} < q$. We define a function $f_{m,k} : \mathbb{N} \to \mathbb{Q}$ such that for a natural number h, $f_{k,m}(h) = \frac{m+h}{k+h+4}$. First, observe that as $\frac{m}{k} < q$ it holds that $f_{m,k}(0) < q$. Moreover, observe that $f_{m,k}$ is strictly increasing, and that it is bounded by 1. Therefore, there exists an h , such that $f_{m,k}(h) < q$, while $f_{m,k}(h+1) \geqslant q$. We call such a number $h^{\#}$.

Suppose now, towards contradiction, that $\frac{m+h^{\#}+1}{k+h^{\#}+4} < \frac{a}{b}$. Then, we have that $b(m+h^{\#}+1) < a(k+h^{\#}+4)$, which is equivalent to $a(k+h^{\#})+4a > b(m+h^{\#})+b$. We denote this inequality as α. Additionally, as $f_{m,k}(h^{\#}+1) \geqslant q$, we know that $\frac{m+h^{\#}+1}{k+h^{\#}+5} \geqslant \frac{a}{b}$. So, $a(k+h^{\#}+5) \leqslant b(m+h^{\#}+1)$, and thus $-a(k+h^{\#}+5) \geqslant -b(m+h^{\#}+1)$. This is equivalent to $-a(k+h^{\#}) - 5a \geqslant -b(m+h^{\#}+1)$. We denote this inequality as β. By adding α and β we get that $-a \geqslant 0$, so $a \leqslant 0$. But this is impossible, since $\frac{a}{b} > 0$.

□

We will further denote such a number as $h^{\#}_{m,k,q}$, or $h^{\#}$, if m, k and q are clear from the context. The following lemma will be relevant for the placement of a pump-down gadget.

Lemma 8.12. *For every rational number $q \in (0,1)$ and $m, k \in \mathbb{N}$, such that $\frac{m}{k} \geqslant q$ there is a natural h, such that $\frac{m}{k+h} < q$, but $\frac{m+1}{k+h} \geqslant q$.*

Proof. Take any such m, k and $q = \frac{a}{b}$. We first define a function $g_{m,k} : \mathbb{N} \to \mathbb{Q}$, such that, for each natural number h, we have that $g_{k,m}(h) = \frac{m}{k+h}$. Observe that $g_{m,k}(0) = \frac{m}{k}$ and that $g_{m,k}$ is strictly decreasing, while it is bounded by 0. So, there exists a natural h, such that $g_{m,k}(h) < q$, but $g_{m,k}(h-1) \geqslant q$, as $q > 0$. We will further call such a number h^{+}.

Then, suppose towards contradiction that $\frac{m+1}{k+h^{+}} < \frac{a}{b}$.Then, we have that $bm + b < ak + ah^{+}$, and so $-bm - b > -ak - ah^{+}$. We denote this inequality as α. Also, notice that by definition of h^{+} we get that $\frac{m}{k+h^{+}-1} \geqslant \frac{a}{b}$. So, $bm \geqslant ak + ah^{+} - a$. We denote this inequality as β. By adding α and β, we get that $-b \geqslant -a$, and so $a \geqslant b$ which is impossible, since $\frac{a}{b} < 1$. □

We denote such a number as $h^{+}_{m,k,q}$, or h^{+}, if m, k and q are clear from the context.

We now construct, for a graph $G = (V_G, E_G)$, a labelled social network $EN_G = (N, E, f)$, which we call an *encoding* of G. Let us first describe the subnetwork of EN_G, which we call a *G-gadget*. For every vertex in V_G, we create a vertex in the G-gadget, which is labelled r, with the relation between them being identical to E_G. Further, for every vertex i in the G-gadget, we create a set of vertices labelled r, which we denote as r_i, and a set of vertices labelled b, which we call b_i. We require all members of r_i and of b_i to be adjacent to i. Further, we set the cardinalities of r_i and b_i to be smallest, such that $\#R^{N(i)} + \#r_i - \#b_i = k - 1$.

Further, for every vertex j in $r_i \cup b_i$, we create two vertices labelled b, adjacent to j, and to each other. Finally, we construct the minimum number of isolated vertices labelled b, satisfactory for b to be the strict majority colour in EN_G.

Observe now that the only vertices under illusion in this encoding are those in the G-gadget. Moreover, all of the members of this gadget are under illusion. Figure 8.9 depicts an example of EN_G.

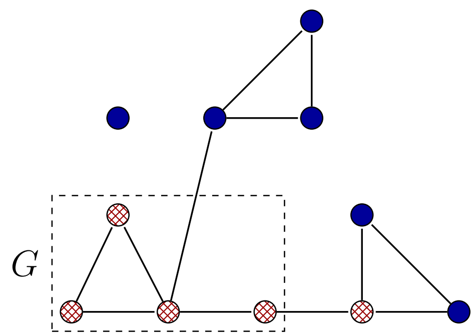

Figure 8.9: Example of an encoding EN_G, for a graph G with four vertices, such that three of them form a clique, and one of them is a dependent of a member of this clique, and $k = 3$.

We further call $\#V_G - k$ the *requirement*, or r_G. Also, we call $\binom{k}{2}$ the *budget*, or b_G. We say that network $EN'_G = (N, E', f)$ satisfies the requirement an the budget if $\#\{e \in N^2 : e \in E \text{ iff } e \notin E'\} \leqslant b_G$, while at most r_G vertices are under illusion in EN'_G.

Lemma 8.13. *For every graph G, there is a network $EN'_G = (N, E', f)$, which satisfies the requirement and the budget if and only if there exists a k-clique in G.*

Proof. Take a graph $G = (V_G, E_G)$. First, suppose that there exists a k-clique in G. Then, take such a clique, and call the corresponding set of vertices in the G-gadget C. Observe that since, following previous observations, all of the vertices, which are under illusion in the encoding $EN_G = (N, E, f)$, are in the G-gadget, it holds that the network (N, E', f) ,with $E' = E \setminus \{(i, j) : i, j \in C\}$, satisfies the budget, since $\#\{(i, j) : i, j \in C\} = \binom{k}{2}$. Observe that it also satisfies the requirement, as $\#C = k$, and we have that for every $i \in C$, it holds that the margin of victory in i's neighbourhood amounts to $\#N_r^{V_G}(i) + \#r_i - \#b_i - k - 1$, which by construction is equal to 0, and hence i is not under illusion.

Suppose now, that there is no k-clique in G. Further, suppose towards contradiction, that there is a network $EN'_G = (N, E', f)$, which satisfies the requirement and the budget. Then, there is a set of vertices $S \subseteq V(C)$, with $\#S = k$, such that, for every $i \in S$, we have that illusion is eliminated from i in EN'_G. Further, by assumption, we have that S is not a clique. Notice, however, that then, as $b_G = \binom{k}{2}$, at least one member of S is under illusion in EN'_G. Contradiction.

□

This observation allows us to show **NP**-hardness of q-Illusion Elimination.

Lemma 8.14. *q-Illusion Elimination is **NP**-complete, for every $q \in (0, 1)$,*

Proof. Consider any rational $q \in (0, 1)$. First, observe that q-Illusion Elimination is in **NP**, as verifying if a labelling induces a q-majority illusion is possible in polynomial time. Let us further construct a network E^q_G and a number m for graph G, such that the answer to q-Illusion Elimination for E^q_G and m is positive if and only if G contains a k-clique.

In the instance we consider, we will check whether we can find a subnetwork of E^q_G, in which connections between at most b_G pairs of vertices can be changed, and in which q-majority illusion does not hold. The first component of E^q_φ is E_G. If $\frac{\#V(G)-k}{V(E_G)} < q$, then we construct a l-pump up gadget for $l = h^{\#}_{\#V(E_G)-k,k,q}$. Otherwise, we construct a l'-pump down gadget, for $l' = h^{+}_{\#V(E_G)-k,k,q}$. Let us show now that the answer to q-Illusion Elimination for E^q_G and k is positive if and only if G contains a k-clique.

First, suppose that G contains a k-clique. We will show that the answer to q-Illusion Elimination for E^q_G and k is positive. Let us first consider the case, in which $\frac{V(G)-k}{\#V(E_G)} < q$. Then observe that, as G is contains a k-clique, by Lemma 8.13 we have that it is possible to find a subnetwork E'_G of E_G, in which $\binom{k}{2}$ edges are altered, and where illusion was eliminated from k vertices. But then, by Lemma 8.11, we get that $\frac{\#V(E_G)-k+l}{\#V(G)-k+l+4} < q$. So we can construct a network of E^q_φ, in which only $\binom{k}{2}$ edges are altered, but q-majority illusion does not hold. Similarly, if $\frac{\#V(G)-k}{\#V(E_\varphi)} \geqslant q$, we observe that, by Lemma 8.12, we get that $\frac{\#V(E_G)-k}{\#V(E_G)+l'} < q$. So, we get that we can eliminate illusion from k vertices in E_G by modifying $\binom{k}{2}$ edges. But then, we can construct a network E^q_G, in which only $\binom{k}{2}$ edges are removed, while q-majority illusion does not hold.

Suppose now that G does not contain a k-clique. We will show that the answer to the considered problem for E^q_G and k is negative. Let us first consider the case in which $\frac{V(G)-k}{\#V(E_G)} < q$. Notice that, by reasoning in Lemma 8.13, we have that the minimum number of vertices from which illusion needs to be removed for q-majority illusion not to hold in E^q_G is k. Furthermore, let us notice that in the pump-up gadget, the minimum number of edges that is needed to be added to eliminate the illusion from a single vertex is greater than 4. Thus, we get from Lemma 8.13 that since G does not contain a k-clique, it is not possible to remove the illusion from at least k in E^q_φ by altering connections between at most $\binom{k}{2}$ pairs of vertices The reasoning for the case in which $\frac{\#V(G)-k}{\#V(E_G)} \geqslant q$ is symmetric.

□

As a consequence, we get the following hardness result.

Theorem 8.5. *For all $q \in (0,1)$ q-Illusion Elimination is* **W***[1]-hard parametrised by the number of altered edges.*

Proof. Follows immediately from Lemma 8.14, noticing that the budget is bounded by a quadratic function of k. □

Using reductions similar to the one provided above, we obtain **W**[1]-hardness of Removal q-Illusion Elimination and Removal q-Illusion Elimination. To show hardness of Removal q-Illusion Elimination, by reduction from k-Clique, we use the same encoding, as in the proof of Theorem 8.5. We observe that, following the reasoning in the proof of Theorem 8.5, in this construction we get that for a graph G, $q \in (0,1)$, and $E^q_G = (N, E)$, there exists a network $E'^q_G = (N, E')$, such that q-majority illusion does not hold in E'^q_G, while $\{(e \in N^2 : e \in E \textit{ iff } e \notin E'\}$ if and only if $\#E' - \#E \leqslant \binom{k}{2}$. The following result follows.

Lemma 8.15. Removal q-Illusion Elimination *is* **NP***-complete, for every $q \in (0,1)$.*

To show the hardness of Addition q-Illusion Elimination, we provide a reduction from k-Independent Set problem, similar to the previously shown construction. We now construct, for a graph $G = (V_G, E_G)$, a labelled social network $EN_G = (N, E, f)$, which we call an *encoding* of G.

Let us first describe the subnetwork of EN_G, which we call a *G-gadget.* For every vertex in V_G, we create a vertex in the G-gadget, which is labelled b, with the relation between them being identical to E_G. Further, for every vertex i in the G-gadget, we create a set of vertices labelled r, which we denote as r_i, and a set of vertices labelled b, which we call b_i. We also have that all members of r_i and of b_i are adjacent to i. We also denote the set of neighbours of i in the G-gadget as G_i. Further, we set the cardinalities of r_i and b_i to be smallest, such that $\#r_i - \#G_i - \#b_i = k - 1$. Also, for every vertex j in $r_i \cup b_i$, we create three vertices labelled b, adjacent to each other, and with one of them adjacent to j. Finally, we construct the minimum number of isolated vertices labelled b, satisfactory for b to be the strict majority colour in EN_G. Observe now that the only vertices under illusion in this encoding are in the G-gadget. Moreover, all of the members of this gadget are under illusion. An example of an encoding EN_G is shown in Figure 8.10.

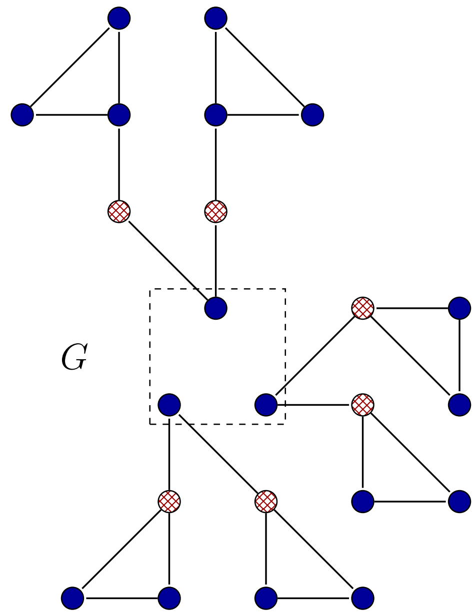

Figure 8.10: An encoding EN_G, for a graph consisting of the independent set of three vertices.

We further call $\#V(G) - k + 1$ the *requirement*, or r_G. Also, we call $\binom{k}{2}$ the *budget*, or b_G. Moreover, we say that a network $EN'_G = (N, E', f)$ satisfies the requirement an the budget if $\#E' - \#E \leqslant b_G$, while less than r_G vertices are under illusion in EN'_G.

Lemma 8.16. Addition q-Illusion Elimination *is* **NP**-*complete for every* $q \in (0, 1)$.

Proof. To show that the claim holds, we observe that, by reasoning symmetric to the proof of Lemma 8.14, for a given graph G, we can construct a network $EN'^q_G = (N, E', f)$ in which q-majority illusion does not hold, while $\#E' - \#E < \binom{k}{2}$ if and only if G contains a k-independent set. Then, the result follows by reasoning symmetric to the proof of Lemma 8.14.

□

By combining Lemma 8.15 and Lemma 8.16, and noticing that in both cases the budget is bounded by a quadratic function of k. we obtain the following theorem.

Theorem 8.6. *For all* $q \in (0, 1)$*,* Addition q-Illusion Elimination *and* Removal q-Illusion Elimination *are* **W***[1]-hard.*

8.4 Plurality Illusion

Having studied the majority illusion for networks labelled with two colours, it is natural to ask whether similar results hold for networks labelled with a larger number of colours. Towards this end, we define the *plurality illusion*, i.e., a social network phenomenon, in which agents see a colour which is not the most popular in the network as a plurality winner. In this section we show that there are networks that admit plurality illusion for all agents, but not 1-majority illusion. This observation motivates further research on the plurality illusion.

Let us show that there are social networks, which allow for colouring with multiple colours, where all agents perceive an option different than the plurality winner as the most popular option, but which do not admit 1-majority illusion. We now define the *plurality illusion*. Let C be a finite set of colours. Given a labelled social network $SN{=}(N, E, f)$, where $f : N \to C$ is a labelling, we denote the set of *most popular colours* in SN as Pl_{SN}. So, $Pl_{SN} = \arg\max_{c \in C} \#\{i \in N : f(i) = c\}$. If the most popular colour is unique, we will call it the *plurality winner*. Similarly, for an agent $i \in N$, we say that Pl^i_{SN} is the set of most popular options in i's neighbourhood. Formally, $Pl^i_{SN} = \arg\max_{c \in C} \#\{i \in N(i) : f(i) = c\}$. If $Pl^i_{SN} = \{c\}$, for some $c \in C$, we say that c is the plurality winner in i's neighbourhood. Then, we say that an agent $i \in N$ is under plurality illusion, if plurality winner in i's neighbourhood is different than the plurality winner (while both exist). Further, we say that f *induces plurality illusion* if all agents in N are under plurality illusion in (N, E, f). Also, we say that (N, E) *admits* plurality illusion if some labelling $f : N \to C$ induces plurality illusion.

Observation 8.4. *There are networks which admit a plurality illusion, but not 1-majority illusion.*

The following example shows that Observation 8.4 holds.

Example 8.1. *Consider the social network shown in Figure 8.11. Let us begin with showing that this network admits a plurality illusion with three colours. To see that consider the labelling in Figure 8.11. Notice that in this case five vertices are labelled blue, four are labelled red, and four are labelled green. Thus, blue is the plurality winner. However, one can verify that there is a plurality winner other than blue in the neighbourhood of every vertex in the network. So, the proposed labelling induces a plurality illusion.*

Furthermore, let us demonstrate that the social network, depicted in Figure 8.11, does not admit 1-majority illusion. Suppose towards contradiction that there

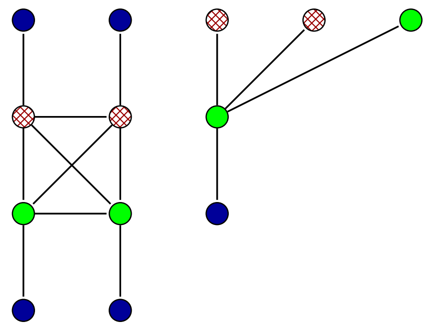

Figure 8.11: Example of a social network admitting a 1-plurality illusion with three colours, but not admitting a 1-majority illusion.

is a labelling f of this network, which induces a 1-majority illusion. Then, observe that there are thirteen vertices in the network, and hence at most six vertices are labelled red in f. Moreover, all vertices in the clique in the left subnetwork, as well as the central vertex in the right subnetwork, have dependents. Hence, they are labelled red in f. Furthermore, the central vertex in the right subnetwork has four neighbours, and hence, by assumption that it is under illusion, at least three of the vertices adjacent to it are labelled red in f. But then, at least eight vertices are labelled red in f, which contradicts the assumptions. So, the network in Figure 8.11 does not admit a 1-majority illusion.

8.5 Conclusion

In this chapter we provided an analysis of computational aspects of majority illusion, focusing on checking the possibility of its occurrence in a social network, and of its elimination. We note that while in this chapter we view majority illusion as an undesirable phenomenon, there might be contexts where its presence does not have overall negative effects. Our work is however agnostic regarding which interventions must take place on such networks.

Summary of Contributions. We showed the algorithmic hardness of checking (Theorems 8.1 and 8.2) and eliminating (Theorems 8.5 and 8.6) q-majority illusion. Furthermore, we provided a number of parametrised algorithms for the verification problem (see Table 8.1). Moreover, we demonstrated $\mathbf{W}[1]$-hardness for the elimination of majority illusion (Theorems 8.5 and 8.6). Informally, we have shown even if illusion identification is a hard problem in general, there are various natural constraints that make it feasible.

Future Research. Our research in this chapter leaves a vast number of avenues for further study. Here, we identify a few of specific potential directions.

- We note that, for elimination we considered, the problem remains hard for a natural parameter. It remains open, however, whether we can find another, good parametrisation.
- Further, in this chapter we have only shown the hardness of q-MAJORITY ILLUSION for $q \geqslant \frac{1}{2}$. Although we conjecture that the problem remains **NP**-hard for smaller values of q, we leave establishing the complexity of q-MAJORITY ILLUSION for such q as an avenue for further research.
- Another open challenge is to explore the setting in which the assumption of binary labelling is lifted. As we have shown in Section 8.4, surprisingly, there are social networks that do not admit a majority illusion but do admit a "plurality" illusion, i.e., agents have a wrong perception of the plurality winner, when more than two colours are allowed. This is particularly relevant for voting contexts such as elections with multiple candidates.
- Furthermore, it remains open to check the existence of a single-exponential time algorithm with respect to neighbourhood diversity.
- We note that, in addition to our parametrised complexity results, designing an efficient algorithm that approximates the maximal value of q in a given instance of q-MAJORITY ILLUSION would be a natural direction for further study.
- Finally, exploring the connections between majority illusion and opinion diffusion is a natural and important follow up. One can observe that in a labelling, which induces 1-majority illusion, all agents adopt the minority opinion after just one opinion diffusion step, given the protocol explored in Chapter 7. This observation motivates further connections between the majority illusion and the spread of opinions.

Chapter 9

Conclusion

Here, we provide a summary of the results obtained in this thesis. Subsequently, we describe avenues for further research, which arise from our work.

Summary. We provided an analysis of algorithmic problems concerning strategic candidate selection by coalitions, as well as of selected aspects of opinion diffusion in social networks, which are relevant to such games. First, we studied an extension of the Hotelling-Downs model, in which parties can only select a position which corresponds to the views of one of their potential nominees. We have shown that checking if a pure Nash equilibrium exists in this context is **NP**-complete in the general case. However, we demonstrated that this problem can be solved in linear time when only two parties compete. Subsequently, we analysed the problem of checking the pure Nash equilibrium existence in competitions, in the context of knockout tournaments. There, we found the problem to be solvable in quasi-polynomial, or polynomial time, for all of the cases which we considered. Then, we have demonstrated that in the case of tournaments which do not involve competing in rounds, the problem of checking the existence of a pure Nash equilibrium is **NP**-complete, when the Uncovered Set rule, or even the Condorcet Winner rule, is used as a selection mechanism. Moreover, for the Uncovered Set, we have found that it is also intractable to check if a coalition can win under some strategy profile. Furthermore, we studied the problem of the convergence of the opinion diffusion protocol, in which agents change their opinion if the strict majority of their influencers disagrees with them. We have found that checking if this protocol terminates for a given input is **PSPACE**-complete. Finally, we established the computational complexity of checking if a social network can be labelled so that at least a given fraction of agents is under majority illusion, i.e., has the strict majority of their neighbours labelled with the colour assigned to the strict minority of agents in the network. We further showed the complexity of checking if the connections between the agents in a labelled network can be altered in a limited way, so that the number of those under majority illusion is sufficiently small.

We note that the results of this thesis can have impact for the research both in social choice theory, and in social network analysis. Further, our research on nominee selection constitutes a starting point for the study of strategic aspects of primaries.

Avenues for Further Research. Our investigations leave vast room for further research.

- First, our results concerning nominee selection assume that parties choose their candidate in order to perform best in the elections, while the other actors in the process, e.g., voters or candidates themselves, are not strategic. We note that in real-world scenarios strategic behaviour of these actors is plausible. Thus, it would be a natural follow-up to study the problem of primaries taking into account the possibility of manipulation of the elections from their side. Furthermore, in all of the frameworks discussed in this thesis, competitions often admit a large number of pure Nash equilibria. We note that this fact constitutes a challenge from the perspective of predicting parties' choices. This observation motivates studying the classes of games in our models, in which there is a unique pure Nash equilibrium.

- Furthermore, our analysis of nominee selection was mainly aimed at establishing the algorithmic results concerning the existence of equilibria. We have not, however, investigated in depth what are the other properties of Nash equilibria in the contexts we considered. For instance, it would be natural to show the bounds on the price of anarchy in our variation of the Hotelling-Downs model, with social welfare of a profile being measured as a sum of distances of voters to their favourite candidate.

- A large amount of questions regarding the impact that opinion diffusion has on primaries remains to be answered. Some of the problems regarding this direction involve checking if there exists a pure Nash equilibrium in a game, after a certain number of rounds of communication. Another potential challenge is to determine what is the complexity of checking if a coalition has a dominant strategy, i.e., a candidate who does best in a competition in every strategy profile, after the determined number of opinion diffusion steps.

- Another important avenue for future investigations is to better understand the classes of networks, in which the threshold-based opinion diffusion protocol we considered terminates in one step. As we have seen, the labelled networks in which all agents are under majority illusion constitute one of such classes. It would be of interest to identify other types of networks for which it is the case, and to study their properties.

- Given that many of the computational problems we considered are not tractable, it would be of interest to explore strategic nominee selection from the empirical perspective. An experimental approach would allow us to better understand structures of equilibria in the games studied in this thesis, and how they can be influenced by agents' deliberation.

Bibliography

Hee-Kap Ahn, Siu-Wing Cheng, Otfried Cheong, Mordecai Golin, and Rene Van Oostrum. Competitive facility location: the Voronoi game. *Theoretical Computer Science*, 310:457–467, 2004.

Nazanin Alipourfard, Buddhika Nettasinghe, Andrés Abeliuk, Vikram Krishnamurthy, and Kristina Lerman. Friendship paradox biases perceptions in directed networks. *Nature Communications*, 11(1):1–9, 2020.

Alon Altman, Ariel D Procaccia, and Moshe Tennenholtz. Nonmanipulable selections from a tournament. In *IJCAI*, pages 686–690, 2009.

Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

Vincenzo Auletta, Ioannis Caragiannis, Diodato Ferraioli, Clemente Galdi, and Giuseppe Persiano. Minority becomes majority in social networks. In *WINE*, pages 74–88, 2015.

Vincenzo Auletta, Ioannis Caragiannis, Diodato Ferraioli, Clemente Galdi, and Giuseppe Persiano. Information retention in heterogeneous majority dynamics. In *WINE*, pages 30–43, 2017.

Vincenzo Auletta, Diodato Ferraioli, and Gianluigi Greco. Reasoning about consensus when opinions diffuse through majority dynamics. In *IJCAI*, pages 49–55, 2018.

Vincenzo Auletta, Diodato Ferraioli, and Gianluigi Greco. On the complexity of reasoning about opinion diffusion under majority dynamics. *Artificial Intelligence*, 284:103–288, 2020.

Vincenzo Auletta, Diodato Ferraioli, and Gianluigi Greco. Optimal majority dynamics for the diffusion of an opinion when multiple alternatives are available. *Theoretical Computer Science*, 869:156–180, 2021.

Robert Mashall Axelrod. The dissemination of culture: a model with local convergence and global polarization. *The Journal of Conflict Resolution*, pages 203–226, 1997.

Haris Aziz, Serge Gaspers, Simon Mackenzie, Nicholas Mattei, Paul Stursberg, and Toby Walsh. Fixing a balanced knockout tournament. In *AAAI*, pages 552–558, 2014.

Haris Aziz, Markus Brill, Felix Fischer, Paul Harrenstein, Jérôme Lang, and Hans Georg Seedig. Possible and necessary winners of partial tournaments. *Journal of Artificial Intelligence Research*, 54:493–534, 2015.

Haris Aziz, Serge Gaspers, Simon Mackenzie, Nicholas Mattei, Paul Stursberg, and Toby Walsh. Fixing balanced knockout and double elimination tournaments. *Artificial Intelligence*, 262:1–14, 2018.

Sayan Bandyapadhyay, Aritra Banik, Sandip Das, and Hirak Sarkar. Voronoi game on graphs. *Theoretical Computer Science*, 562:270–282, 2015.

Christopher L. Barrett, Harry B. Hunt, Madhav V. Marathe, S. S. Ravi, Daniel J. Rosenkrantz, and Richard Edwin Stearns. Reachability problems for sequential dynamical systems with threshold functions. *Theoretical Computer Science*, 295: 41–64, 2003.

Christopher L. Barrett, Harry B. Hunt, Madhav V. Marathe, S. S. Ravi, Daniel J. Rosenkrantz, Richard Edwin Stearns, and Mayur Thakur. Predecessor existence problems for finite discrete dynamical systems. *Theoretical Computer Science*, 386(1-2):3–37, 2007.

Dorothea Baumeister, Ann-Kathrin Selker, and A. Wilczynski. Manipulation of opinion polls to influence iterative elections. In *AAMAS*, pages 132–140, 2020.

Piotr Berman, Marek Karpinski, and Alexander Scott. Approximation hardness of short symmetric instances of MAX-3SAT. Technical report, Weizmann Institute of Science, 2004.

Vittorio Bilò, Michele Flammini, and Cosimo Vinci. The quality of content publishing in the digital era. In *ECAI*, pages 35–42, 2020.

Béla Bollobás. *Modern Graph Theory*. Graduate Texts in Mathematics. Springer, 1998.

Meena Boppana, Rani Hod, Michael Mitzenmacher, and Tom Morgan. Voronoi Choice Games. In *ICALP*, pages 23:1–23:13, 2016.

Allan Borodin, Omer Lev, Nisarg Shah, and Tyrone Strangway. Primarily about primaries. In *AAAI*, pages 1804–1811, 2019.

Sirin Botan, Umberto Grandi, and Laurent Perrussel. Multi-issue opinion diffusion under constraints. In *AAMAS*, pages 828–836, 2019.

F Brandt, M Brill, and Bernhard Harrenstein. *Tournament solutions*, pages 57–84. Cambridge University Press, 2016a.

Felix Brandt and Felix A. Fischer. Computing the minimal covering set. *Mathematical Social Sciences*, 56(2):254–268, 2008.

Felix Brandt, Felix A. Fischer, Paul Harrenstein, and Maximilian Mair. A computational analysis of the tournament equilibrium set. *Social Choice and Welfare*, 34(4):597–609, 2010.

Felix Brandt, Markus Brill, and Bernhard Harrenstein. Tournament Solutions. In F Brandt, V Conitzer, U Endriss, J Lang, and A. D. Procaccia, editors, *Handbook of Computational Social Choice*, pages 453–474. Cambridge University Press, 2016b.

Felix Brandt, Markus Brill, and Paul Harrenstein. Extending tournament solutions. *Social Choice and Welfare*, 51(2):193–222, 2018.

Robert Bredereck and Edith Elkind. Manipulating opinion diffusion in social networks. In *IJCAI*, pages 894–900, 2017.

Markus Brill. Interactive democracy. In *AAMAS*, pages 1183–1187, 2018.

Markus Brill and Vincent Conitzer. Strategic voting and strategic candidacy. In *AAAI*, pages 819–826, 2015.

Sandro Brusco, Marcin Dziubiński, and Jaideep Roy. The Hotelling–Downs model with runoff voting. *Games and Economic Behavior*, 74(2):447–469, 2012.

Matteo Castiglioni, Diodato Ferraioli, and Nicola Gatti. Election control in social networks via edge addition or removal. In *AAAI*, 2020.

Matteo Castiglioni, Diodato Ferraioli, Nicola Gatti, and Giulia Landriani. Election manipulation on social networks: Seeding, edge removal, edge addition. *Journal of Artificial Intelligence Research*, 71:1049–1090, 2021.

Eric Goles Chacc, Françoise Fogelman-Soulié, and Didier Pellegrin. Decreasing energy functions as a tool for studying threshold networks. *Discrete Applied Mathematics*, 12(3):261–277, 1985.

Krishnendu Chatterjee, Rasmus Ibsen-Jensen, and Josef Tkadlec. Robust draws in balanced knockout tournaments. In *IJCAI*, pages 172–179, 2016.

Dmitry Chistikov, Grzegorz Lisowski, Mike Paterson, and Paolo Turrini. Convergence of opinion diffusion is PSPACE-complete. In *AAAI*, pages 7103–7110, 2020.

Rajesh Chitnis and Nimrod Talmon. Can we create large k-cores by adding few edges? In *Computer Science – Theory and Applications*, pages 78–89. Springer International Publishing, 2018.

Zoé Christoff and Davide Grossi. Stability in binary opinion diffusion. In *LORI*, pages 166–180, 2017.

Federico Corò, Emilio Cruciani, Gianlorenzo D'Angelo, and Stefano Ponziani. Vote for me!: Election control via social influence in arbitrary scoring rule voting systems. In *AAMAS*, pages 1895–1897, 2019.

Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

Konrad K. Dabrowski, Matthew Johnson, and Daniël Paulusma. *Clique-width for hereditary graph classes*, page 1–56. London Mathematical Society Lecture Note Series. Cambridge University Press, 2019.

Morris H. de Groot. Reaching a consensus. *Journal of the American Statistical Association*, 69:118–121, 1974.

John A. Doucette, Alan Tsang, Hadi Hosseini, Kate Larson, and Robin Cohen. Inferring true voting outcomes in homophilic social networks. pages 298–329, 2019.

Anthony Downs. An economic theory of democracy. 1957.

John Dryzek and Christian List. Social choice theory and deliberative democracy: a reconciliation. *British Journal of Political Science*, 33(1):1–28, 2003.

Christoph Durr and Nguyen Kim Thang. Nash equilibria in Voronoi games on graphs. In *ESA*, 2007.

Bhaskar Dutta, Matthew O Jackson, and Michel Le Breton. Strategic candidacy and voting procedures. *Econometrica*, 69(4):1013–1037, 2001.

B Curtis Eaton and Richard G Lipsey. The principle of minimum differentiation reconsidered: Some new developments in the theory of spatial competition. *The Review of Economic Studies*, 42(1):27–49, 1975.

H. A. Eiselt, Gilbert Laporte, and Jacques-François Thisse. Competitive location models: A framework and bibliography. *Transportation Science*, 27(1):44–54, 1993.

Horst A. Eiselt. Equilibria in competitive location models. In *Foundations of Location Analysis*, pages 139–162. Springer, 2011.

Edith Elkind, Evangelos Markakis, Svetlana Obraztsova, and Piotr Skowron. Equilibria of plurality voting: Lazy and truth-biased voters. In *SAGT*, pages 110–122. Springer, 2015.

Ulle Endriss and Umberto Grandi. Binary aggregation by selection of the most representative voters. In *AAAI*, pages 668–674, 2014.

Hülya Eraslan and Andrew McLennan. Strategic candidacy for multivalued voting procedures. *Journal of Economic Theory*, 117(1):29–54, 2004.

Piotr Faliszewski and Jörg Rothe. Control and bribery in voting. In *Handbook of Computational Social Choice*, pages 146–168. Cambridge University Press, 2016.

Piotr Faliszewski, Laurent Gourvès, Jérôme Lang, Julien Lesca, and Jérôme Monnot. How Hard is it for a Party to Nominate an Election Winner? In *IJCAI*, pages 257–263, 2016.

Piotr Faliszewski, Rica Gonen, Martin Koutecký, and Nimrod Talmon. Opinion diffusion and campaigning on society graphs. In *IJCAI*, 2018.

Piotr Faliszewski, Rica Gonen, Martin Koutecý, and Nimrod Talmon. Opinion diffusion and campaigning on society graphs. *Journal of Logic and Computation*, 32(6):1162–1194, 2022.

Michal Feldman, Amos Fiat, and Iddan Golomb. On voting and facility location. In *EC*, pages 269–286, 2016a.

Michal Feldman, Amos Fiat, and Svetlana Obraztsova. Variations on the Hotelling-Downs model. In *AAAI*, pages 496–501, 2016b.

David Fisher and Jennifer Ryan. Tournament games and positive tournaments. *Journal of Graph Theory*, 19(2):217–236, 1995.

Gaëtan Fournier. General distribution of consumers in pure Hotelling games. *International Journal of Game Theory*, 48(1):33–59, 2019.

Jacob Fox, Tim Roughgarden, C. Seshadhri, Fan Wei, and Nicole Wein. Finding cliques in social networks: A new distribution-free model. In *ICALP*, volume 107, pages 55:1–55:15, 2018.

Jacob Fox, Tim Roughgarden, C. Seshadhri, Fan Wei, and Nicole Wein. Finding cliques in social networks: A new distribution-free model. *SIAM Journal on Computing*, 49(2):448–464, 2020.

András Frank and Éva Tardos. An application of simultaneous diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1):49–65, 1987.

Eric Goles, Pedro Montealegre, Ville Salo, and Ilkka Törmä. PSPACE-completeness of majority automata networks. *Theoretical Computer Science*, 609:118–128, 2016.

M. Grabisch and A. Rusinowska. A model of influence in a social network. *Theory and Decision*, 69(1):69–96, 2010.

Umberto Grandi. Social choice and social networks. *Trends in Computational Social Choice*, pages 169–184, 2017.

Umberto Grandi, Emiliano Lorini, and Laurent Perrussel. Propositional opinion diffusion. In *AAMAS*, pages 989–997, 2015.

Mark Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 83(6):1420–1443, 1978.

Frank Gurski and Egon Wanke. The tree-width of clique-width bounded graphs without $K_{n,\ n}$. In *WG*, pages 196–205, 2000.

Paul Harrenstein, Grzegorz Lisowski, Ramanujan Sridharan, and Paolo Turrini. A Hotelling-Downs framework for party nominees. In *AAMAS*, pages 593–601, 2021.

Noam Hazon, Paul Dunne, Sarit Kraus, and Michael Wooldridge. How to rig elections and competitions. In *COMSOC*, pages 301–312, 2008.

Lane Hemaspaandra. Computational social choice and computational complexity: Bffs? In *AAAI*, pages 7911–7977, 2018.

Nathan O Hodas, Farshad Kooti, and Kristina Lerman. Friendship paradox redux: Your friends are more interesting than you. In *ICWSM*, 2013.

Harold Hotelling. Stability in competition. *The Economic Journal*, 39(153):41–57, 1929.

Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

J. R. Isbell. A class of simple games. *Duke Mathematical Journal*, 25(3):423–439, 1958.

Matthew O. Jackson and Simon Wilkie. Endogenous games and mechanisms: Side payments among players. *Review of Economic Studies*, 72(2):543–566, 2005.

Neil F. Johnson, Nicolas Velásquez, Nicholas Johnson Restrepo, Rhys Leahy, Nicholas Gabriel, Sara El Oud, Minzhang Zheng, Pedro Manrique, Stefan Wuchty, and Yonatan Lupu. The online competition between pro- and anti-vaccination views. *Nature*, 582(7811):230–233, 2020.

Ravi Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of Operations Research*, 12(3):415–440, 1987.

Bruce Kapron, Gautam Srivastava, and S Venkatesh. Social network anonymization via edge addition. In *ASONAM*, pages 155–162, 2011.

Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

Akinori Kawachi, Mitsunori Ogihara, and Kei Uchizawa. Generalized predecessor existence problems for boolean finite dynamical systems on directed graphs. *Theoretical Computer Science*, 762:25–40, 2019.

David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *SIGKDD*, pages 137–146, 2003.

David Kempe, Jon M. Kleinberg, and Éva Tardos. Influential nodes in a diffusion model for social networks. In *ICALP*, pages 1127–1138, 2005.

Michael P Kim, Warut Suksompong, and Virginia Vassilevska Williams. Who can win a single-elimination tournament? *SIAM Journal on Discrete Mathematics*, 31(3):1751–1764, 2017.

Tomohiro Koana, Christian Komusiewicz, and Frank Sommer. Exploiting c-closure in kernelization algorithms for graph problems. In *ESA*, pages 65:1–65:17, 2020.

Aleksei Kondratev and Vladimir Mazalov. Tournament solutions based on cooperative game theory. *International Journal of Game Theory*, pages 1–27, 2019.

Christine Konicki and Virginia Vassilevska Williams. Bribery in balanced knockout tournaments. In *AAMAS*, pages 2066–2068, 2019.

Tuukka Korhonen. A single-exponential time 2-approximation algorithm for treewidth. In *FOCS*, pages 184–192, 2021.

Sven Kosub. Dichotomy results for fixed-point existence problems for boolean dynamical systems. *Mathematics in Computer Science*, 1(3):487–505, 2008.

G. Laffond, J. F. Laslier, and M. Le Breton. The bipartisan set of a tournament game. *Games and Economic Behavior*, 5:182–201, 1993.

Michael Lampis. Algorithmic meta-theorems for restrictions of treewidth. *Algorithmica*, 64(1):19–37, 2012.

Jean-Francois Laslier. *Tournament solutions and majority voting*. Studies in Economic Theory. Springer Verlag, 1997.

Silvio Lattanzi and D Sivakumar. Affiliation networks. In *STOC*, pages 427–434, 2009.

H. W. Jr. Lenstra. Integer programming with a fixed number of variables. *Mathemathics of Operations Research*, 8(4):538–548, 1983.

Kristina Lerman, Xiaoran Yan, and Xin-Zeng Wu. The "majority illusion" in social networks. *PloS one*, 11(2):1–13.

David Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11(2):329–343, 1982.

Grzegorz Lisowski, M.S Ramanujan, and Paolo Turrini. Equilibrium computation for knockout tournaments played by groups. In *AAMAS*, pages 807–815, 2022.

Michael Maschler, Shmuel Zamir, and Eilon Solan. *Game theory*. Cambridge University Press, 2020.

Marios Mavronicolas, Burkhard Monien, Vicky G. Papadopoulou, and Florian Schoppmann. Voronoi games on cycle graphs. In *Mathematical Foundations of Computer Science*, pages 503–514, 2008.

Reshef Meir. *Strategic Voting*. Morgan & Claypool, 2018.

Reshef Meir, Maria Polukarov, Jeffrey S. Rosenschein, and Nicholas R. Jennings. Iterative voting and acyclic games. *Artificial Intelligence*, 252:100–122, 2017.

Nicholas R. Miller. A new solution set for tournaments and majority voting: Further graph- theoretical approaches to the theory of voting. *American Journal of Political Science*, 24(1):68–96, 1980.

Neeldhara Misra. On the parameterized complexity of party nominations. In *ADT*, pages 112–125. Springer, 2019.

JW Moon. *Topics on tournaments*. Holt, Reinhart and Winston, 1968.

Hervé Moulin. On strategy-proofness and single peakedness. *Public Choice*, 35(4): 437–455, 1980.

Matías Núñez and Marco Scarsini. Competing over a finite number of locations. *Economic Theory Bulletin*, 4(2):125–136, 2016.

Matías Núñez and Marco Scarsini. Large spatial competition. In *Spatial Interaction Models*, pages 225–246. Springer, 2017.

Svetlana Obraztsova, Edith Elkind, Maria Polukarov, and Zinovi Rabinovich. Strategic candidacy games with lazy candidates. In *IJCAI*, pages 610–616, 2015.

Mitsunori Ogihara and Kei Uchizawa. Computational complexity studies of synchronous Boolean finite dynamical systems on directed graphs. *Information and Computation*, 256:226–236, 2017.

Pekka Orponen. On the computational power of discrete Hopfield nets. In *ICALP*, pages 215–226, 1993.

Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

Annemieke Reijngoud and Ulle Endriss. Voter response to iterated poll information. In *AAMAS*, 2012.

Daniel J. Rosenkrantz, Madhav V. Marathe, S. S. Ravi, and Richard Edwin Stearns. Testing phase space properties of synchronous dynamical systems with nested canalyzing local functions. In *AAMAS*, pages 1585–1594, 2018.

Itay Sabato, Svetlana Obraztsova, Zinovi Rabinovich, and Jeffrey S Rosenschein. Real candidacy games: A new model for strategic candidacy. In *AAMAS*, pages 867–875, 2017.

Fernando P Santos, Simon A Levin, and Vítor V Vasconcelos. Biased perceptions explain collective action deadlocks and suggest new mechanisms to prompt cooperation. *iScience*, 24(4):102375, 2021.

Reinhard Selten. Reexamination of the perfectness concept for equilibrium points in extensive games. *International Journal of Game Theory*, 4:25–55, 1975.

Abhijit Sengupta and Kunal Sengupta. A Hotelling–Downs model of electoral competition with the option to quit. *Games and Economic Behavior*, 62(2):661–674, 2008.

Alexander J. Stewart, Mohsen Mosleh, Marina Diakonova, Antonio A. Arechar, David G. Rand, and Joshua B. Plotkin. Information gerrymandering and undemocratic decisions. *Nature*, 573(7772):117–121, 2019.

Donald E. Stokes. Spatial models of party competition. *American Political Science Review*, 57(2):368–377, 1963.

Predrag T. Tosic. Phase transitions in possible dynamics of cellular and graph automata models of sparsely interconnected multi-agent systems. In *AAMAS*, pages 474–483, 2017.

Virginia Vassilevska Williams. Fixing a tournament. In *AAAI*, pages 895–900, 2010.

Emiel Christiaan Henrik Veendorp and Anjum Majeed. Differentiation in a two-dimensional market. *Regional Science and Urban Economics*, 25(1):75–83, 1995.

Matheus P Viana, Emanuele Strano, Patricia Bordin, and Marc Barthelemy. The simplicity of planar networks. *Scientific reports*, 3(1):1–6, 2013.

Thuc Vu, Alon Altman, and Yoav Shoham. On the complexity of schedule control problems for knockout tournaments. In *AAMAS*, pages 225–232, 2009.

A. Wilczynski. Poll-confident voters in iterative voting. In *AAAI*, pages 2205–2212.

Bryan Wilder and Yevgeniy Vorobeychik. Controlling elections through social influence. In *AAMAS*, pages 265–273.

Virginia Vassilevska Williams and Hervé Moulin. Knockout tournaments. In Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D.Editors Procaccia, editors, *Handbook of Computational Social Choice*, pages 453–474. Cambridge University Press, 2016.

Xiaotian Zhou and Zhongzhi Zhang. *Maximizing Influence of Leaders in Social Networks*, page 2400–2408. Association for Computing Machinery, New York, NY, USA, 2021.

Zhongxin Zhou, Fan Zhang, Xuemin Lin, Wenjie Zhang, and Chen Chen. K-core maximization: An edge addition approach. In *IJCAI*, pages 4867–4873, 2019.